Machine Learning Approaches that Extend Healthcare: Algorithms & Applications

Yuzhe Yang

Machine Learning Approaches that Extend Healthcare: Algorithms & Applications

by

Yuzhe Yang

B.S., Peking University (2018)
M.S., Massachusetts Institute of Technology (2020)

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

May 2024

© 2024 Yuzhe Yang. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by: Yuzhe Yang
Department of Electrical Engineering and Computer Science
May 17, 2024

Certified by: Dina Katabi
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Machine Learning Approaches that Extend Healthcare: Algorithms & Applications

by

Yuzhe Yang

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2024, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Modern clinical systems frequently exhibit sporadic patient visits, delayed diagnoses, and unequal care distribution among diverse populations. Often, diseases aren't identified until they reach advanced stages. The scarcity of specialists and disparities in healthcare access further complicate the long-term monitoring, timely intervention, and unbiased assessments. This thesis addresses the above challenges by developing artificial intelligence (AI) and machine learning (ML) algorithms and building practical systems that use these algorithms to solve key problems in healthcare and medicine.

Specifically, on the algorithms front, the thesis introduces principled ML approaches to achieve fair, unbiased, and generalizable AI models, addressing core challenges in real-world medical data which encompass four main axes:

- **Label Scarcity:** The thesis presents a novel self-supervised learning scheme that learns periodic and frequency information in data without labels, enabling representation learning for periodic tasks like vital signs estimation with minimal labeling efforts.
- **Data Imbalance:** The thesis develops new ML algorithms to address data imbalance in regression, filling the gap in techniques for practical imbalanced regression problems.
- **Domain Generalization:** The thesis presents theoretically grounded learning methods that ensure generalization across imbalanced domains and unseen environments.
- **Subpopulation Shifts:** The thesis studies learning in the presence of underrepresented subgroups, providing actionable insights for model deployment in real-world settings.

On the applications front, the thesis develops new AI-driven biomarkers and systems for human disease and medicine leveraging the proposed algorithms, enabling discovery and advancing delivery and equity in healthcare:

- **Early Diagnosis Biomarker for Parkinson's:** The thesis presents an AI-based biomarker for Parkinson's disease that enables early detection years before standard clinical diagnosis, as well as longitudinal progression tracking using nocturnal breathing signals.
- **In-Home Touchless Monitoring of Sleep Posture:** The thesis designs novel AI systems for continuous and contactless sleep posture monitoring overnight in the user's own home using wireless signals.

- **Equitable Medical AI Deployments In The Wild:** The thesis establishes best practices for medical imaging AI models that maintain their performance and fairness in deployments beyond their initial training contexts, across diverse populations and unseen sites.

Thesis Supervisor:	Dina Katabi
Title:	Professor of Electrical Engineering and Computer Science

Previously Published Material

Chapter 2 revises a previous publication [1]: Yuzhe Yang, Xin Liu, Jiang Wu, Silviu Borac, Dina Katabi, Ming-Zher Poh, Daniel McDuff. SimPer: Simple Self-Supervised Learning of Periodic Targets. ICLR 2023.

Chapter 3 revises a previous publication [2]: Yuzhe Yang, Kaiwen Zha, Ying-Cong Chen, Hao Wang, Dina Katabi. Delving into Deep Imbalanced Regression. ICML 2021.

Chapter 4 revises a previous publication [3]: Yuzhe Yang, Hao Wang, Dina Katabi. On Multi-Domain Long-Tailed Recognition, Generalization and Beyond. ECCV 2022.

Chapter 5 revises a previous publication [4]: Yuzhe Yang, Haoran Zhang, Dina Katabi, Marzyeh Ghassemi. Change is Hard: A Closer Look at Subpopulation Shift. ICML 2023.

Chapter 6 revises a previous publication [5]: Yuzhe Yang, Yuan Yuan, Guo Zhang, Hao Wang, Ying-Cong Chen, Yingcheng Liu, Christopher Tarolli, Daniel Crepeau, Jan Bukartyk, Mithri Junna, Aleksandar Videnovic, Terry Ellis, Melissa Lipford, Ray Dorsey, Dina Katabi. Artificial Intelligence-Enabled Detection and Assessment of Parkinson's Disease using Nocturnal Breathing Signals. Nature Medicine, 28(10), 2022.

Chapter 7 revises a previous publication [6]: Shichao Yue, Yuzhe Yang, Hao Wang, Hariharan Rahul, Dina Katabi. BodyCompass: Monitoring Sleep Posture with Wireless Signals. ACM UbiComp 2020.

Chapter 8 revises a previous publication [7]: Yuzhe Yang, Haoran Zhang, Judy Gichoya, Dina Katabi, Marzyeh Ghassemi. The Limits of Fair Medical Imaging AI In The Wild. To appear in Nature Medicine, 2024.

Acknowledgments

First and foremost, I would like to express my deepest thanks to my advisor Dina Katabi. Over the past years, she has worked closely with me, provided continuous support, and consistently inspired me. She gave me the freedom to explore a diverse array of projects beyond her own research focus. Dina devoted countless hours to discussing my research and career, assisting with my papers and talks, and shaping me as an independent researcher. She always motivated me to aim higher, to solve more impactful problems, and to become a better researcher. She has truly been a role model for who I want to be as a researcher, mentor, and advisor. I could not ask for a better advisor, and I aspire to extend the same guidance and support to my students that Dina has offered me.

I am sincerely grateful for my thesis committee members and letter writers: Marzyeh Ghassemi, Daniel McDuff, Shwetak Patel, Hao Wang, and Terry Ellis. Marzyeh has been a long-term collaborator and a beacon of brilliance, creativity, and kindness. She has always been available to mentor me on research, life, and beyond, which has profoundly influenced my Ph.D. journey and research trajectory. Daniel mentored me at Google Health Research during my summer internship. He has been a fantastic friend, an insightful mentor, and a great collaborator. I hope to have many conversations and collaborations with Daniel in the years ahead. Shwetak exemplifies not only conducting impactful academic research but also developing products that reach millions. It has been as enlightening working with him and his group as it has been learning from his insights. Hao has been a mentor, friend, and elder brother for years. He not only showed me the ropes of doing research but also the path to becoming a successful researcher, reinforcing my determination to pursue an academic career. Terry has collaborated with me on multiple projects on

monitoring patients with Parkinson's disease, which has made me realize that nothing is more fulfilling than seeing my research make a real-world impact on patients' lives.

Sincere thanks also go to all the senior mentors who have provided tremendous help and guidance through this journey. Devavrat Shah provided insights and guidance during our collaborations in the initial years of my time at MIT. Tommi Jaakkola offered invaluable feedback on my projects, regardless of his direct involvement, and also served on my research qualifying exam committee. Muriel Médard has been my academic advisor since I joined MIT. Ming-Zher Poh mentored me and provided support throughout my internship at Google. I am also profoundly thankful to the people who helped me reach MIT in the first place: I am grateful to Lingyang Song and Kaigui Bian, who introduced me to research and mentored my undergraduate research thesis. I was also fortunate to work with Zhu Han during my undergraduate studies. I thank them all for their insights, patience, kindness, and extensive support.

I would like to extend my heartfelt thanks to my exceptional clinical collaborators. As a researcher at the intersection of AI and health, my research would not have been possible without their expertise and generous help: Ray Dorsey, Judy Gichoya, Avanti Gulhane, Mithri Junna, Melissa Lipford, Faisal Mahmood, Domenico Mastrodicasa, Dushyant Sahani, Christopher Tarolli, Ipsit Vahia, Aleksandar Videnovic, and Wei Wu.

Much credit for the research in this thesis and throughout my Ph.D. is owed to my incredibly talented colleagues. Zhi Xu has been both a close collaborator and a friend since I joined MIT. Our discussions and brainstorming have always been productive, laying the groundwork for my Master's thesis. Xin Liu and I met during our overlapping internships at Google, and this turned out to be the beginning of our multi-year, fruitful collaborations. I am grateful to Xin for his warm introduction to the community at UW, and I look forward to our enduring collaborations and friendship. Haoran Zhang has been my partner-in-crime for over two years, as we tirelessly engaged in brainstorming, coding, debugging, and writing. I am confident in Haoran's bright future at MIT and beyond, and I am excited to continue our efforts together.

I have also been fortunate to collaborate with many other co-authors: Jamie Adams, Silviu Borac, Jan Bukartyk, Peng Cao, Richard J. Chen, Tiffany Y. Chen, Ying-Cong Chen, Hailey Cray, Daniel Crepeau, Emma C. Dyer, Lijie Fan, Rogerio Feris, Minghao Guo, Tom Hartvigsen, Hao He, Rumen Hristov, Piotr Indyk, Guillaume Jaume, Stella Jensen-Roberts,

Karl Kieburtz, Tianhong Li, Dahua Lin, Jana Lipkova, Yingcheng Liu, Yujia Liu, Ziwei Liu, Karlo Lizarraga, Ming Y. Lu, Chengqian Ma, Rose May, William McGrory, Intae Moon, Taylor Myers, Girish Narayanswamy, Timothy Nordahl, Meghan Pawlik, Hariharan Rahul, Ruth Schneider, Muhammad Shaban, Jeany Son, Andrew Song, Dogyoon Song, Julia Soto, Anurag Vaidya, Emma Waddell, Edward J. Wang, Drew F. K. Williamson, Renee Wilson, Jiang Wu, Rui Xu, Yuan Yuan, Shichao Yue, Kaiwen Zha, and Guo Zhang. I am immensely grateful for their contributions and for all that they have taught me.

I have also had the privilege of being part of the NETMIT research group together with: Deepak Vasisht, Zachary Kabelac, Chen-Yu Hsu, Mingmin Zhao, Michail Ouroutzoglou, Chao Li, Ali Mirzazadeh, and Mary McDavitt. Together, we experienced numerous rounds of internal feedback on papers and talks, and many enjoyable group outings, making the lab a fun and engaging place to spend time.

I have also received tremendous guidance from people at various institutions during my job search. In addition to those I mentioned above, I would like to thank Omid Abari, Ehsan Adeli, Venkat Arun, Vineet Bafna, Irene Chen, Sumit Chopra, Ying Ding, Noémie Elhadad, David Harwath, Ben Hu, Haojian Jin, Shalmali Joshi, Swarun Kumar, Yunzhu Li, Yifan Peng, Aditya Prakash, Jim Rehg, Chiara Sabatti, Jimeng Sun, Yizhou Sun, Berk Ustun, Erdem Varol, Byron Wallace, Atlas Wang, Kai Wang, Sheng Wang, Wei Wang, Yao Wang, Adam Yala, Serena Yeung, Huan Zhang, and Marinka Zitnik, for their support that helped me navigate the stressful yet rewarding process. I deeply appreciate them.

During my time at MIT, I have been funded by industrial fellowships and scholarships from MathWorks, Takeda, and Baidu, and research grants from NSF and the Michael J. Fox Foundation. I appreciate the support of all funding agencies.

◇

I have been extremely lucky to be surrounded by many wonderful friends who have brought immense joy and fun into my life. With Jie Xu and Xia Xiao, we have had countless unforgettable trips and fantastic game nights across time and location. Yiyue Luo and Zeyu Wu have been fantastic travel companions, card game enthusiasts, dining partners, and sports buddies. Spending time with Zeyuan Shang was always enjoyable, and watching the Celtics games with Wenbo Tao and others at TD Garden was a highlight. A special shout-out to Tianhong Li and Lijie Fan; starting in the same lab and year, we have shared numerous memorable moments both inside and outside the lab. Being a member of the

MIT CSSA basketball team (the Winning Team) has been a fantastic experience, and I will always cherish the times we spent practicing, playing pick-up games, and competing in tournaments within MIT and across the Northeastern Division. I would also like to thank Lei Cao, Zhi Huang, Bingzhao Li, Tian Li, Hongzi Mao, Liang Shi, Hanchen Wang, Xuhai Xu, Lu Yang, Ruohan Zhang, and many others, for all the chats about research and life. Many friends provided warmth and companionship even though they were located far away. I will forever cherish their friendship, warmth, support, and conversations.

Finally, yet most importantly, words cannot fully express my gratitude to my family. My parents have provided unending support and love throughout my life, as well as access to the best education possible. My grandparents, parents-in-law, sisters, brothers, uncles, and aunts have supported and cared for me tirelessly, despite the thousands of miles that separate us during my Ph.D. journey. I could not have achieved this without your unconditional love and support. To my wife, Luxin, and our cat, Luckie; with you, I feel at home no matter where we are. This thesis is dedicated to you.

Contents

List of Figures

List of Tables

CHAPTER 1
Introduction

Recent decades have witnessed the profound transformations in clinical and healthcare systems [8], significantly driven by artificial intelligence (AI) and machine learning (ML) that hold tremendous potential to reshape health and medicine [9]. The application of AI in health spans a wide array of crucial medical tasks including automated diagnosis, risk modeling, triage, remote health monitoring, treatment selection, optimizing clinical trials, disease progression modeling, and enhancing patient interaction with healthcare systems [10, 11, 12, 13, 14, 15, 16, 17]. Yet, despite these advancements, today's healthcare continues to face unique real-world challenges and persistent gaps that need to be addressed:

- **Time Gap:** The infrequent nature of clinical visits, which spans months or even years, poses a significant challenge for early disease detection and timely intervention [5]. For example, small delays in diagnosing conditions like cancer can drastically increase mortality risk [18]. This issue is further exacerbated by substantial gaps in healthcare observation – periods during which patients receive no clinical oversight [19]. Therefore, methods that can see across *time* are essential, facilitating the early detection and longitudinal tracking of diseases.

- **Location Gap:** The scarcity of healthcare professionals and facilities often restricts hospital care to a limited number of individuals. Not everyone can easily access clinical resources, especially for continuous care [20]. For instance, approximately 40% of individuals with Parkinson's disease never see a specialist, largely due to health inequities,

including restricted access to specialized medical centers [10]. Consequently, intelligent sensing and computing technologies that deliver healthcare directly to patients' homes are crucial, enabling equitable access to medical services regardless of *location*.

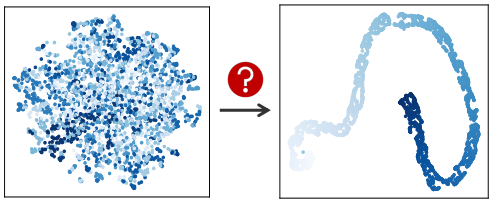
- **Individual Gap:** With the increasing use of AI for tasks such as triage or screening, it is crucial that these technologies function effectively for every patient they serve. However, AI models could fail in unexpected ways, especially when deployed in new environments or subpopulations [4]. For instance, Epic's AI model for detecting early signs of sepsis, while effective in initial tests, frequently misdiagnosed patients when deployed across hundreds of hospitals [21]. Thus, it is critical to develop AI systems that perform consistently and accurately across diverse settings, domains, and *individuals*, ensuring reliable healthcare outcomes for all patients.

The goal of the research presented in this dissertation is to address the above gaps by developing new AI and ML algorithms, and building practical systems that deploy these algorithms to extend healthcare beyond the clinic. However, developing such algorithms is non-trivial. Real-world medical data is inherently imperfect and biased, often marked by limited annotations, skewed data distributions, underrepresented subgroups, and pervasive biases affecting diverse populations. Additionally, translating these algorithms into practical systems, however, is not always straightforward.

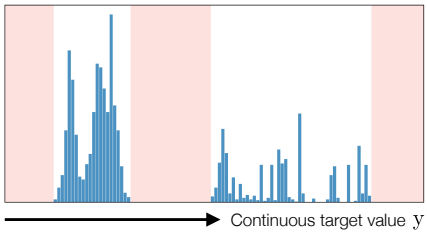
In this dissertation, we make contributions on both **algorithm** and **application** fronts:

- **Algorithms (Fig. 1-1):** This dissertation designs generic ML algorithms to tackle challenges presented by real-world healthcare data, including label scarcity [1], data imbalance and biases [2, 22, 23], subpopulation fairness [4], and domain generalization [3]. These principled learning algorithms serve as foundational elements to build trustworthy medical decision-making systems, ensuring fairness, robustness, and generalizability for high-stakes applications.

- **Applications (Fig. 1-2):** This dissertation introduces novel AI-driven biomarkers and systems that bridge the aforementioned gaps and improve discovery, delivery, and equity in healthcare and medicine. Specifically, they address persistent clinical constraints in (1) **time**, by facilitating pre-clinical diagnosis and post-clinical prognosis [5, 16], (2) **location**, by delivering medical assessments directly to patients' homes [24, 6], and (3) **individual**, by ensuring precise and equitable outcomes for all patients [25, 26, 7].



(a) Periodic learning **without labels** (Chapter 2).



(b) Learning **imbalanced regression** (Chapter 3).

(c) Learning from **distinct domains** (Chapter 4).

(d) Learning over **subpopulations** (Chapter 5).

Figure 1-1: **Machine learning algorithms developed to address real-world medical data problems. (a)** We study simple self-supervised learning of periodic targets with no labels (more information is available here). **(b)** We address learning imbalanced data with continuous targets (more information is available here). **(c)** We tackle imbalanced learning and generalization with data arising from multiple distinct domains (more information is available here). **(d)** We analyze subpopulation shift and investigate learning with underrepresented subgroups (more information is available here).

1.1 Learning Algorithms

Real-world health data are by their nature complex and imperfect, characterized by *scarce* clinical annotations, *skewed* data distributions, *underrepresentation* of certain demographics, and *biases* across diverse populations. This dissertation aims to tackle these multifaceted challenges, addressing label scarcity (Fig. 1-1(a)), data imbalance (Fig. 1-1(b)), domain generalization (Fig. 1-1(c)), and subpopulation shifts (Fig. 1-1(d)). While driven by healthcare, the algorithmic foundations are broadly applicable to other high-stakes applications.

1.1.1 Label Scarcity

From human physiology to environmental evolution, important processes in nature often exhibit meaningful and strong *periodic* or *quasi-periodic* changes. Due to their inherent label scarcity, learning useful representations for periodic tasks with limited or no supervision is of great benefit. Yet, existing self-supervised learning (SSL) methods overlook the intrinsic periodicity in data, and fail to learn representations that capture periodic or frequency attributes (see Fig. 1-1(a)).

This dissertation first presents SimPer, a simple contrastive SSL regime for learning periodic information in data. To exploit the periodic inductive bias, SimPer introduces customized augmentations, feature similarity measures, and a generalized contrastive loss for learning efficient and robust periodic representations.

The resulting algorithm SimPer learns robust periodic representations with high frequency resolution, as shown in Fig. 1-1(a). Extensive experiments on common real-world tasks in human behavior analysis, environmental sensing, and healthcare domains verify the superior performance of SimPer compared to state-of-the-art SSL methods, highlighting its intriguing properties including better data efficiency, robustness to spurious correlations, and generalization to distribution shifts.

1.1.2 Data Imbalance

Real-world data often exhibit imbalanced distributions, where certain target values have significantly fewer observations. Existing techniques for dealing with imbalanced data focus on targets with categorical indices, i.e., different classes. However, many tasks involve continuous targets, where hard boundaries between classes do not exist.

This dissertation formally defines and studies the Deep Imbalanced Regression (DIR) problem as learning from such imbalanced data with continuous targets, dealing with potential missing data for certain target values, and generalizing to the entire target range (see Fig. 1-1(b)). Motivated by the intrinsic difference between categorical and continuous label space, we propose distribution smoothing for both labels and features, which explicitly acknowledges the effects of nearby targets, and calibrates both label and learned feature distributions. We curate and benchmark large-scale DIR datasets from common real-world tasks in computer vision, natural language processing, and healthcare domains. Extensive experiments verify the superior performance of our strategies. This thesis fills the gap in benchmarks and techniques for practical imbalanced regression problems.

1.1.3 Domain Generalization

Existing studies on data imbalance focus on single-domain settings, i.e., samples are from the same data distribution. However, natural data can originate from distinct domains, where a minority class in one domain could have abundant instances from other domains. This dissertation further formalizes the task of Multi-Domain Long-Tailed Recognition

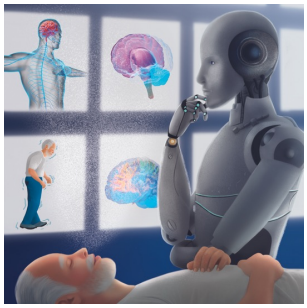
(MDLT), which learns from multi-domain imbalanced data, addresses *label imbalance*, *domain shift*, and *divergent label distributions across domains*, and generalizes to all domain-class pairs (see Fig. 1-1(c)).

We first develop the *domain-class transferability graph*, and show that such transferability governs the success of learning in MDLT. We then propose `BoDA`, a theoretically grounded learning strategy that tracks the upper bound of transferability statistics, and ensures *balanced* alignment and calibration across imbalanced domain-class distributions. We curate five MDLT benchmarks based on widely-used multi-domain datasets, and compare `BoDA` to twenty algorithms that span different learning strategies. Extensive and rigorous experiments verify the superior performance of `BoDA`. Further, as a byproduct, `BoDA` establishes new state-of-the-art on Domain Generalization benchmarks, highlighting the importance of addressing data imbalance across domains, which can be crucial for improving generalization to unseen domains.

■ 1.1.4 Subpopulation Shifts

Finally, ML models often perform poorly on *subgroups* that are underrepresented in the training data. Yet, little is understood on the variation in mechanisms that cause subpopulation shifts, and how algorithms generalize across such diverse shifts at scale. In this dissertation, we provide a fine-grained analysis to model and benchmark subpopulation shift (see Fig. 1-1(d)).

We first propose a unified framework that dissects and explains common shifts in subgroups. We then establish a comprehensive benchmark of 20 state-of-the-art algorithms evaluated on 12 real-world datasets in vision, language, and healthcare domains. With results obtained from training over 10,000 models, we reveal intriguing observations for future progress in this space. First, existing algorithms only improve subgroup robustness over certain types of shifts but not others. Moreover, while current algorithms rely on group-annotated validation data for model selection, we find that a simple selection criterion based on worst-class accuracy is surprisingly effective even without any group information. Finally, unlike existing works that solely aim to improve worst-group accuracy (WGA), we demonstrate the fundamental tradeoff between WGA and other important metrics, highlighting the need to carefully choose testing metrics.



(a) **Early diagnosis** of Parkinson's disease using nocturnal breathing (Chapter 6).

(b) **In-home touchless** monitoring of sleep posture using wireless signals (Chapter 7).

(c) **Equitable AI deployment** across diverse environments and subgroups (Chapter 8).

Figure 1-2: **New AI-driven biomarkers and systems for disease and medicine using the proposed algorithms. (a)** AI-based biomarker for early detection and longitudinal progression tracking of Parkinson's disease using nocturnal breathing signals [5]. **(b)** Contactless monitoring of sleep posture overnight in the home using AI and wireless signals [6]. **(c)** Fair and equitable medical AI model deployment in new environments and patient populations [7].

1.2 Applications in Healthcare and Medicine

Next, we translate the proposed ML algorithms to develop practical systems that extend healthcare capabilities. Specifically, these systems enhance healthcare across three key dimensions: **time**, **location**, and **individual**. For time, they enable early detection of chronic diseases before clinical diagnosis (Fig. 1-2(a)). For location, the systems bring comprehensive health assessments into people's homes, and passively profile diverse facets of human health (Fig. 1-2(b)). For individuals, they support equitable decision-making systems, providing actionable clinical insights for AI models deployed in real-world settings (Fig. 1-2(c)).

1.2.1 Early Diagnosis Biomarker for Parkinson's Disease

Parkinson's disease (PD) is the fastest-growing neurological disease in the world [27]. Over one million people are living with PD in the US as of 2020 [28], resulting in an economic burden of $52 billion per year [29]. Today, however, there are no effective biomarkers for diagnosing PD or tracking its progression. In this dissertation, we develop an AI model to detect PD and track its progression from nocturnal breathing signals (see Fig. 1-2(a)).

The AI-based system takes as input one night of breathing signals, which can be collected using a breathing belt worn on the person's chest or abdomen [30], or using low power radio signals and analyzing its reflections off the person's body [6]. The nocturnal

breathing is passed as input to our neural network, which analyses it to produce two outputs: (1) it predicts whether the person has PD, and (2) it estimates the severity of PD with respect to the clinical gold standard [5].

The model is evaluated on a large dataset comprising 7,671 individuals, created by pulling data from several US hospitals and multiple public datasets. The AI model can detect PD with an area-under-the-curve (AUC) of 0.90 and 0.85 on the held-out and external test sets, respectively. The AI model can also estimate PD severity and progression in accordance with the Movement Disorder Society-Unified Parkinson's Disease Rating Scale (R=0.94, p=3.6e-25). Moreover, the model can assess PD in the home setting in a touchless manner, by extracting breathing from radio waves that bounce off a person's body during sleep. Our study demonstrates the feasibility of objective, noninvasive, at-home assessment of PD, and also provides initial evidence that this AI model may be useful for risk assessment prior to clinical diagnosis.

1.2.2 In-Home Touchless Monitoring of Sleep Posture

Monitoring sleep posture is important for avoiding bedsores after surgery, reducing apnea events, tracking the progression of Parkinson's disease, and even alerting epilepsy patients to potentially fatal sleep postures. Today, there is no easy way to track sleep postures. Past work has proposed installing cameras in the bedroom, mounting accelerometers on the subject's chest, or embedding pressure sensors in their bedsheets. Unfortunately, such solutions jeopardize either the privacy of the user or their sleep comfort.

In this dissertation, we introduce BodyCompass, the first RF-based system that provides accurate sleep posture monitoring overnight in the user's own home (see Fig. 1-2(b)). BodyCompass works by studying the RF reflections in the environment. It disentangles RF signals that bounced off the subject's body from other multipath signals. It then analyzes those signals via a custom machine learning algorithm to infer the subject's sleep posture. BodyCompass is easily transferable and can apply to new homes and users with minimal effort. We empirically evaluate BodyCompass using over 200 nights of sleep data from 26 subjects in their own homes. Our results show that, given one week, one night, or 16 minutes of labeled data from the subject, BodyCompass's corresponding accuracy is 94%, 87%, and 84%, respectively.

1.2.3 Equitable Medical AI across Environments and Subgroups

Finally, as AI rapidly approaches human-level performance in medical imaging, it is crucial that it does not exacerbate or propagate healthcare disparities. Prior research has established AI's capacity to infer demographic data from chest X-rays, leading to a key concern: do models using demographic shortcuts have unfair predictions across subpopulations? In this dissertation, we conduct a thorough investigation into the extent to which medical AI utilizes demographic encodings, focusing on potential fairness discrepancies within both in-distribution training sets and external test sets (see Fig. 1-2(c)).

Our analysis covers three key medical imaging disciplines: radiology, dermatology, and ophthalmology, and incorporates data from six global chest X-ray datasets. We confirm that medical imaging AI leverages demographic shortcuts in disease classification. While correcting shortcuts algorithmically effectively addresses fairness gaps to create "locally optimal" models within the original data distribution, this optimality is not true in new test settings. Surprisingly, we find that models with less encoding of demographic attributes are often most "globally optimal", exhibiting better fairness during model evaluation in new test environments. Our analysis provides best practices for medical imaging models which maintain their performance and fairness in deployments beyond their initial training contexts, underscoring critical considerations for AI clinical deployments across populations and sites.

1.3 Thesis Roadmap

This thesis is divided into two parts.

Part I describes the algorithmic foundations of reliable machine learning in the wild. Chapter 2 presents a simple self-supervised learning algorithm for periodic targets. Chapter 3 introduces algorithms and benchmarks for imbalanced regression. Chapter 4 focuses on long-tailed learning in the presence of multiple imbalanced domains and how to generalize to novel domains. Chapter 5 analyzes subpopulation shifts and provides actionable insights for model deployment in real-world settings.

Part II describes the applications and systems designed using the proposed ML algorithms. Chapter 6 describes an AI-driven biomarker for Parkinson's disease that enables early detection years before standard clinical diagnosis, as well as longitudinal progres-

sion tracking using nocturnal breathing signals. Chapter 7 presents a novel AI system for continuous and contactless sleep posture monitoring overnight in the user's own home using wireless signals. Chapter 8 establishes best practices for medical imaging AI models that maintain their performance and fairness in deployments beyond their initial training contexts, across diverse populations and unseen sites.

Finally, in Chapter 9, we conclude and discuss the future work.

Part I

Algorithms: Machine Learning In The Wild

CHAPTER 2
Simple Self-Supervised Learning of Periodic Targets

Practical and important applications of machine learning in the real world, from monitoring the earth from space using satellite imagery [31] to detecting physiological vital signs in a human being [32], often involve recovering ***periodic*** changes. In the **health** domain, learning from video measurement has shown to extract (quasi-)periodic vital signs including atrial fibrillation [33], sleep apnea episodes [34] and blood pressure [32]. In the **environmental remote sensing** domain, periodic learning is often needed to enable nowcasting of environmental changes such as precipitation patterns or land surface temperature [35]. In the **human behavior analysis** domain, recovering the frequency of changes or the underlying temporal morphology in human motions (e.g., gait or hand motions) is crucial for those rehabilitating from surgery [17], or for detecting the onset or progression of neurological conditions such as Parkinson's disease [16, 5].

While learning periodic targets is important, labeling such data is typically challenging and resource intensive. For example, if designing a method to measure heart rate, collecting videos with highly synchronized gold-standard signals from a medical sensor is time consuming, labor intensive, and requires storing privacy sensitive bio-metric data. Fortunately, given the large amount of unlabeled data, *self-supervised learning* that captures the underlying periodicity in data would be promising.

Yet, despite the great success of self-supervised learning (SSL) schemes on solving dis-

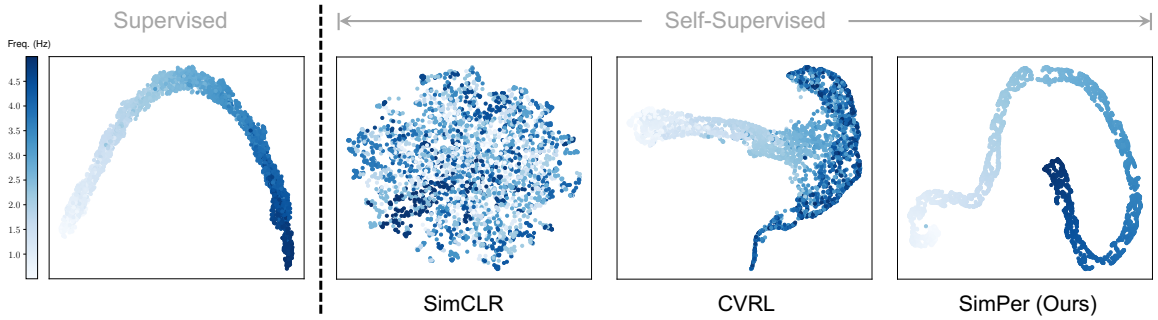


Figure 2-1: **Learned representations of different methods on a periodic learning dataset, RotatingDigits (details in Section 2.3).** Existing self-supervised learning schemes fail to capture the underlying periodic or frequency information in data. In contrast, `SimPer` learns robust periodic representations with high frequency resolution.

crete classification or segmentation tasks, such as image classification [36, 37], object detection [38], action recognition [39], or semantic labeling [40], less attention has been paid to designing algorithms that capture periodic or quasi-periodic temporal dynamics from data. Interestingly, we highlight that existing SSL methods inevitably overlook the intrinsic periodicity in data: Fig. 2-1 shows the UMAP [41] visualization of learned representations on RotatingDigits, a toy periodic learning dataset that aims to recover the underlying rotation frequency of different digits (details in Section 2.3). As the figure shows, state-of-the-art (SOTA) SSL schemes fail to capture the underlying periodic or frequency information in the data. Such observations persist across tasks and domains as we show later in Section 2.3.

To fill the gap, we present `SimPer`, a simple self-supervised regime for learning periodic information in data. Specifically, to leverage the temporal properties of periodic targets, `SimPer` first introduces a *temporal self-contrastive learning* framework, where positive and negative samples are obtained through *periodicity-invariant* and *periodicity-variant* augmentations from the **same** input instance. Further, we identify the problem of using conventional feature similarity measures (e.g., $\cos(\cdot)$) for periodic representation, and propose *periodic feature similarity* to explicitly define how to measure similarity in the context of periodic learning. Finally, to harness the intrinsic *continuity* of augmented samples in the frequency domain, we design a *generalized contrastive loss* that extends the classic InfoNCE loss to a soft regression variant that enables contrasting over continuous labels (frequency).

To support practical evaluation of SSL of periodic targets, we benchmark `SimPer` against

SOTA SSL schemes on six diverse periodic learning datasets for common real-world tasks in human behavior analysis, environmental remote sensing, and healthcare. Rigorous experiments verify the robustness and efficiency of `SimPer` on learning periodic information in data.

In this chapter, we make the following contributions: (i) We identify the limitation of current SSL methods on periodic learning tasks, and uncover intrinsic properties of learning periodic dynamics with self-supervision over other mainstream tasks. (ii) We design `SimPer`, a simple & effective SSL framework that learns periodic information in data. (iii) We conduct extensive experiments on six diverse periodic learning datasets in different domains: human behavior analysis, environmental sensing, and healthcare. Rigorous evaluations verify the superior performance of `SimPer` against SOTA SSL schemes. (iv) Further analyses reveal intriguing properties of `SimPer` on its data efficiency, robustness to spurious correlations & reduced training data, and generalization to unseen targets.

2.1 Related Work

Periodic Tasks in Machine Learning. Learning or recovering periodic signals from high dimensional data is prevailing in real-world applications. Examples of periodic learning include recovering and magnifying physiological signals (e.g., heart rate or breathing) [42], predicting weather and environmental changes (e.g., nowcasting of precipitation or land surface temperatures) [35, 31], counting motions that are repetitious (e.g., exercises or therapies) [43, 44], and analyzing human behavior (e.g., gait) [16]. To date, much prior work has focused on designing customized neural architectures [45, 43], loss functions [46], and leveraging relevant learning paradigms including transfer learning [47] and meta-learning [48] for periodic learning in a *supervised* manner, with high-quality labels available. In contrast to these past work, we aim to learn robust & efficient periodic representations in a *self-supervised* manner.

Self-Supervised Learning. Learning with self-supervision has recently attracted increasing interests, where early approaches mainly rely on pretext tasks, including exemplar classification [49], solving jigsaw puzzles [50], object counting [51], clustering [52], and predicting image rotations [53]. More recently, a line of work based on contrastive losses [54, 55, 36, 37] shows great success in self-supervised representations, where similar em-

beddings are learned for different views of the same training example (*positives*), and dissimilar embeddings for different training examples (*negatives*). Successful extensions have been made to temporal learning domains including video understanding [56] or action classification [39]. However, current SSL methods have limitations in learning periodic information, as the periodic inductive bias is often overlooked in method design. Our work extends existing SSL frameworks to periodic tasks, and introduces new techniques suitable for learning periodic targets.

2.2 The SimPer Framework

When learning from periodic data in a self-supervised manner, a fundamental question arises:

*How do we design a self-supervised task such that **periodic** inductive biases are exploited?*

We note that periodic learning exhibits characteristics that are distinct from prevailing learning tasks. *First*, while most efforts on exploring invariances engineer transformations in the spatial (e.g., image recognition) or temporal (e.g., video classification) domains, dynamics in the **frequency** domain are essential in periodic tasks, which has implications for how we design (in)variances. *Second*, unlike conventional SSL where a cosine distance is typically used for measuring feature similarity, representations learned for repetitious targets inherently possess periodicity that is insensitive to certain shifts (e.g., shifts in feature index), which warrants new machinery for measuring *periodic* similarity. *Third*, labels of periodic data have a natural ordinality and continuity in the frequency domain, which inspires the need for strategies beyond instance discrimination, that contrast over *continuous* targets.

We present `SimPer` (Simple SSL of Periodic Targets), a unified SSL framework that addresses each of the above limitations. Specifically, `SimPer` first introduces a ***temporal self-contrastive learning*** scheme, where we design *periodicity-invariant* and *periodicity-variant* augmentations for the *same* input instance to create its effective *positive* and *negative* views in the context of periodic learning (Section 2.2.1). Next, `SimPer` presents ***periodic feature similarity*** to explicitly define how one should measure the feature similarity when the learned representations inherently possess periodic information (Section 2.2.2). Finally, in order to exploit the continuous nature of augmented samples in the frequency domain,

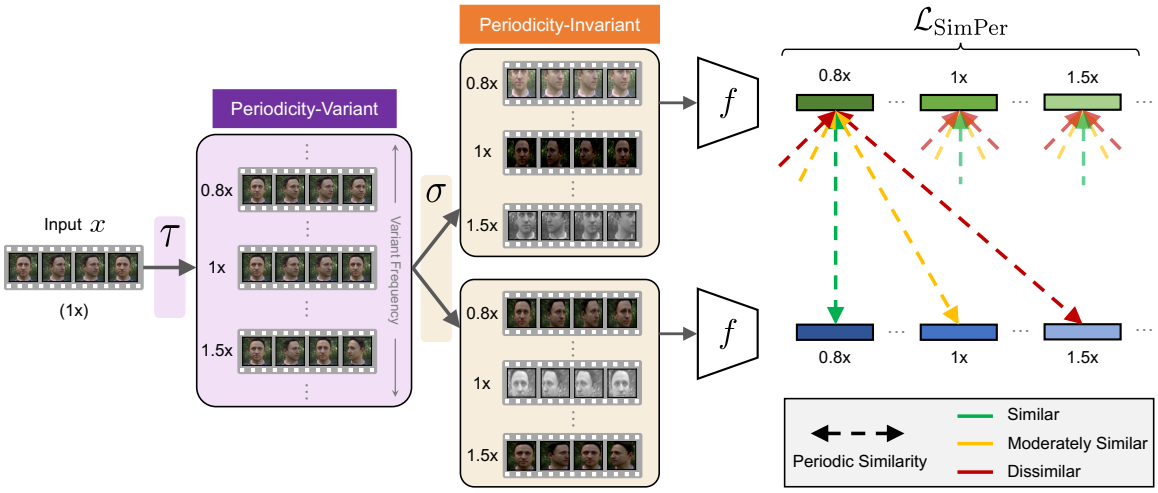


Figure 2-2: **An overview of the `SimPer` framework.** Input sequence is first passed through periodicity-variant transformations $\tau(\cdot)$ to create a series of speed (frequency) changed samples, where each augmented sample exhibits different underlying periodic signals due to the altered frequency, and can be treated as *negative* examples for each other. The augmented series are then passed through two sets of periodicity-invariant transformations $\sigma(\cdot)$ to create different invariant views (*positives*). All samples are then encoded in the feature space through a shared encoder $f(\cdot)$. The `SimPer` loss is calculated by contrasting over continuous speed (frequency) labels of different feature vectors, using customized periodic feature similarity measures.

we propose a ***generalized contrastive loss*** that extends the classic InfoNCE loss [54] from *discrete* instance discrimination to *continuous* contrast over frequencies, which takes into account the meaningful distance between continuous labels (Section 2.2.3).

■ 2.2.1 Temporal Self-Contrastive Learning Framework

Problem Setup. Let $\mathcal{D} = \{(\mathbf{x}_i)\}_{i=1}^{N}$ be the unlabeled training set, where $\mathbf{x}_i \in \mathbb{R}^D$ denotes the input sequence. We denote as $\mathbf{z} = f(\mathbf{x}; \theta)$ the representation of $\mathbf{x}$, where $f(\cdot; \theta)$ is parameterized by a deep neural network with parameter θ. To preserve the full temporal dynamics and information, $\mathbf{z}$ typically extracts *frame-wise* feature of $\mathbf{x}$, i.e., $\mathbf{z}$ has the same length as input $\mathbf{x}$.

As motivated, ***frequency*** information is most essential when learning from periodic data. Precisely, augmentations that change the underlying frequency effectively alter the identity of the data (periodicity), and vice versa. This simple insight has implications for how we design proper (in)variances.

Periodicity-Variant Augmentations. We construct negative views of input data through

Table 2-1: **Differences of view constructions.**

Algorithm	Positives	Negatives
Conventional SSL methods	**Instance:** Same **Aug.:** Invariant	**Instance:** Different **Aug.:** Invariant
`SimPer`	**Instance:** Same **Aug.:** Period.-Invariant	**Instance:** Same **Aug.:** Period.-Variant

transformations in the frequency domain. Specifically, given input sequence $\mathbf{x}$, we define periodicity-variant augmentations $\tau \in \mathcal{T}$, where $\mathcal{T}$ represents the set of transformations that change $\mathbf{x}$ with an *arbitrary* speed that is feasible under the Nyquist sampling theorem. As Fig. 2-2 shows, `SimPer` augments $\mathbf{x}$ by M times, obtaining a series of *speed* (*frequency*) changed samples $\{\tau_1(\mathbf{x}), \tau_2(\mathbf{x})\}, \ldots, \tau_M(\mathbf{x})\}$, whose relative speeds satisfy $s_1 < s_2 < ... < s_M, s_i \propto \text{freq}(\tau_i(\mathbf{x}))$. Such augmentation effectively changes the underlying periodic targets with shifted frequencies, thus creating different *negative* views. Therefore, although the original target frequency is unknown, we effectively devise *pseudo speed (frequency) labels* for unlabeled $\mathbf{x}$. In practice, we limit the speed change range to be within $[s_{\text{min}}, s_{\text{max}}]$, ensuring the augmented sequence is longer than a fixed length in the time dimension.

Periodicity-Invariant Augmentations. We further define periodicity-invariant augmentation $\sigma \in \mathcal{S}$, where $\mathcal{S}$ denotes the set of transformations that do not change the *identity* of the original input. When the set is finite, i.e., $\mathcal{S} = \{\sigma_1, \ldots, \sigma_k\}$, we have $\text{freq}(\sigma_i(\mathbf{x})) = \text{freq}(\sigma_j(\mathbf{x})), \forall i, j \in [k]$. Such augmentations can be used to learn invariances in the data from the perspective of periodicity, creating different *positive* views. Practically, we leverage spatial (e.g., crop & resize) and temporal (e.g., reverse, delay) augmentations to create different views of the same instance (see Fig. 2-2).

Temporal Self-Contrastive Learning. Unlike conventional contrastive SSL algorithms where augmentations are exploited to produce invariances, i.e., creating different positive views of the data, `SimPer` introduces periodicity-variant augmentations to explicitly model what *variances* should be in periodic learning. Concretely, negative views are no longer from other *different* instances, but directly from the *same* instance itself, realizing a ***self-contrastive*** scheme. Table 2-1 details the differences.

We highlight the benefits of using the self-contrastive framework. First, it provides *ar-*

bitrarily large negative sample sizes, as long as the Nyquist sampling theorem is satisfied. This makes `SimPer` not dependent on the actual training set size, and enables effective contrasting even under limited data scenarios. We show in Section 2.3.2 that when drastically reducing the dataset size to only 5% of the total samples, `SimPer` still works equally well, substantially outperforming supervised counterparts. Second, our method naturally leads to *hard* negative samples, as periodic information is directly being contrasted, while unrelated information (e.g., frame appearance) are maximally preserved across negative samples. This makes `SimPer` robust to spurious correlations in data (Section 2.3.5).

■ 2.2.2 Feature Similarity in the Context of Periodic Learning

We identify that *feature similarity measures* are also different in the context of periodic representations. Consider sampling two short clips $\mathbf{x}_1, \mathbf{x}_2$ from the same input sequence, but with a frame shift t. Assume the frequency does not change within the sequence, and its period $T > t$. Since the underlying information does not change, by definition their features should be close in the embedding space (i.e., high feature similarity). However, due to the shift in time, when extracting frame-level feature vectors, the indexes of the feature representations (which represent different time stamps) will no longer be aligned. In this case, if directly using a cosine similarity as defined in conventional SSL literature, the similarity score would be low, despite the fact that the actual similarity is high.

Periodic Feature Similarity. To overcome this limitation, we propose to use *periodic* feature similarity measures in `SimPer`. Fig. 2-3 highlights the properties and differences between conventional feature similarity measures and the desired similarity measure in periodic learning. Specifically, existing SSL methods adopt similarity measures that emphasize strict "closeness" between two feature vectors, and are sensitive to shifted or reversed feature indexes. In contrast, when aiming for learning periodic features, a proper periodic feature measure should retain high similarity for features with shifted (sometimes reversed) indexes, while also capturing a continuous similarity change when the feature frequency varies, due to the meaningful distance in the frequency domain.

Concrete Instantiations. We provide two practical instantiations to effectively capture the periodic feature similarity. Note that these instantiations can be easily extended to high-dimensional features (in addition to the time dimension) by averaging across other dimensions.

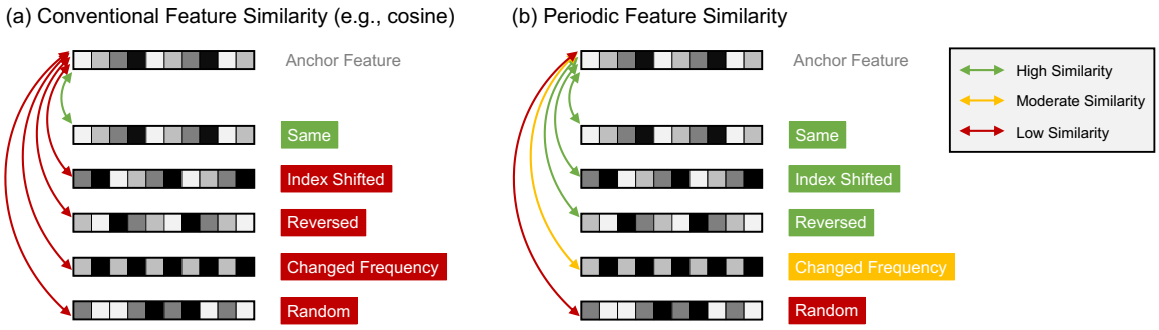


Figure 2-3: **Differences between (a) conventional feature similarity, and (b) *periodic* feature similarity.** A proper periodic feature similarity measure should induce high similarity for features with shifted (sometimes reversed) indexes, while capturing a continuous similarity change when the feature frequency varies.

- *Maximum cross-correlation (**MXCorr**)* measures the maximum similarity as a function of offsets between signals [57], which can be efficiently computed in the frequency domain.
- *Normalized power spectrum density (**nPSD**)* calculates the distance between the normalized PSD of two feature vectors. The distance can be a cosine or L_2 distance (details in Appendix A.3.4).

■ 2.2.3 Generalized Contrastive Loss with Continuous Targets

Motivated by the fact that the augmented views are ***continuous*** in frequency, where the *pseudo speed labels* $\{s_i\}_{i=1}^{M}$ are known through augmentation (i.e., a view at $1.1\times$ is more similar to the original than that at $2\times$), we relax and extend the original InfoNCE contrastive loss [54] to a soft variant, where it generalizes from discrete instance discrimination to continuous targets.

From Discrete Instance Discrimination to Continuous Contrast. The classic formulation of the InfoNCE contrastive loss for each input sample $\mathbf{x}$ is written as

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\text{sim}(\mathbf{z}, \widehat{\mathbf{z}})/\nu)}{\sum_{\mathbf{z}' \in \mathcal{Z} \setminus \{\mathbf{z}\}} \exp(\text{sim}(\mathbf{z}, \mathbf{z}')/\nu)}, \tag{2.1}$$

where $\widehat{\mathbf{z}} = f(\widehat{\mathbf{x}})$ ($\widehat{\mathbf{x}}$ is the positive pair of $\mathbf{x}$ obtained through augmentations), $\mathcal{Z}$ is the set of features in current batch, ν is the temperature constant, and $\text{sim}(\cdot)$ is usually instantiated by a dot product. Such format indicates a *hard* classification task, where target label is 1 for positive pair and 0 for all negative pairs. However, negative pairs in `SimPer` inherently

possess a meaningful distance, which is reflected by the similarity of their relative speed (frequency). To capture this intrinsic continuity, we consider the contributions from *all* pairs, with each scaled by the *similarity* in their labels.

Generalized InfoNCE Loss. For an input sample $\mathbf{x}$, `SimPer` creates M variant views with different speed labels $\{s_i\}_{i=1}^M$. Given the features of two sets of invariant views $\{\mathbf{z}_i\}_{i=1}^M$, $\{\mathbf{z}'_i\}_{i=1}^M$, we have

$$\mathcal{L}_{\text{SimPer}} = \sum_i -\sum_{j=1}^{M} \frac{\exp(w_{i,j})}{\sum_{k=1}^{M}\exp(w_{i,k})} \log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}'_j)/\nu)}{\sum_{k=1}^{M}\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}'_k)/\nu)}, \quad w_{i,j} := \text{sim}_{\text{label}}(s_i, s_j), \tag{2.2}$$

where $\text{sim}(\cdot)$ denotes the *periodic* feature similarity as described previously, and $\text{sim}_{\text{label}}(\cdot)$ denotes the *continuous* label similarity measure. In practice, $\text{sim}_{\text{label}}(\cdot)$ can be simply instantiated as inverse of the L_1 or L_2 label difference (e.g., $1/|s_i - s_j|$).

Interpretation. $\mathcal{L}_{\text{SimPer}}$ is a simple generalization of the InfoNCE loss from discrete instance discrimination (single target classification) to a weighted loss over all augmented pairs (soft regression variant), where the soft target $\exp(w_{i,j})/\sum_k \exp(w_{i,k})$ is driven by the *label* (speed) similarity $w_{i,j}$ of each pair. Note that when the label becomes discrete (i.e., $w_{i,j} \in \{0, 1\}$), $\mathcal{L}_{\text{SimPer}}$ degenerates to the original InfoNCE loss. We demonstrate in Appendix A.3.4 that such continuity modeling via a generalized loss helps achieve better downstream performance than simply applying InfoNCE.

■ 2.3 Experiments

Datasets. We perform extensive experiments on six datasets that span different domains and tasks. Complete descriptions of each dataset are in Appendix A.1, Fig. A-1, and Table A-1.

- ***RotatingDigits*** *(Synthetic Dataset)* is a toy periodic learning dataset consists of rotating MNIST digits [58]. The task is to predict the underlying digit rotation frequency.
- ***SCAMPS*** *(Human Physiology)* [59] consists of 2,800 synthetic videos of avatars with realistic peripheral blood flow. The task is to predict averaged heart rate from input videos.
- ***UBFC*** *(Human Physiology)* [60] contains 42 videos with synchronized gold-standard contact PPG recordings. The task is to predict averaged heart rate from input video clips.

Table 2-2: **Feature evaluation results on RotatingDigits.**

	FFT		1-NN	
Metrics	MAE↓	MAPE↓	MAE↓	MAPE↓
SimCLR [36]	2.96	109.27	0.98	48.30
MoCo v2 [37]	2.83	90.78	0.62	32.74
BYOL [63]	2.20	78.43	0.46	22.08
CVRL [39]	1.69	49.09	0.38	14.41
SimPer	**0.22**	**16.49**	**0.09**	**4.51**
Gains	**+1.47**	**+32.60**	**+0.29**	**+9.90**

Table 2-3: **Feature evaluation results on SCAMPS.**

	FFT		1-NN	
Metrics	MAE↓	MAPE↓	MAE↓	MAPE↓
SimCLR [36]	27.48	38.39	34.09	40.79
MoCo v2 [37]	28.16	40.23	35.61	42.47
BYOL [63]	26.15	37.34	32.77	38.26
CVRL [39]	27.67	38.80	33.32	39.54
SimPer	**14.45**	**22.09**	**13.75**	**18.64**
Gains	**+11.70**	**+15.25**	**+19.02**	**+19.62**

Table 2-4: **Feature evaluation results on UBFC.**

	FFT		1-NN	
Metrics	MAE↓	MAPE↓	MAE↓	MAPE↓
SimCLR [36]	16.92	14.73	16.23	18.62
MoCo v2 [37]	14.64	13.17	15.12	16.56
BYOL [63]	17.86	16.90	18.13	19.34
CVRL [39]	11.75	10.67	12.36	13.38
SimPer	**8.78**	**7.46**	**8.92**	**10.21**
Gains	**+2.97**	**+3.21**	**+3.44**	**+3.17**

Table 2-5: **Feature evaluation results on PURE.**

	FFT		1-NN	
Metrics	MAE↓	MAPE↓	MAE↓	MAPE↓
SimCLR [36]	23.70	22.07	29.48	31.44
MoCo v2 [37]	24.23	24.08	30.82	33.95
BYOL [63]	23.24	21.78	29.27	31.03
CVRL [39]	19.27	18.94	22.08	23.75
SimPer	**13.97**	**12.88**	**14.03**	**15.35**
Gains	**+5.30**	**+6.06**	**+8.05**	**+8.40**

- ***PURE*** *(Human Physiology)* [61] contains 60 videos with synchronized gold-standard contact PPG recordings. The task is to predict averaged heart rate from input video clips.
- ***Countix*** *(Action Counting)*. The Countix dataset [43] is a subset of the Kinetics [62] dataset annotated with segments of repeated actions and corresponding counts. The task is to predict the count number given an input video.
- ***Land Surface Temperature (LST)*** *(Satellite Sensing)*. LST contains hourly land surface temperature maps over the continental United States for 100 days (April 7th to July 16th, 2022). The task is to predict future temperatures based on past satellite measurements.

Network Architectures. We choose a set of logical architectures from prior work for our experiments. On RotatingDigits and SCAMPS, we employ a simple 3D variant of the CNN architecture as in [3]. Following [45], we adopt a variant of TS-CAN model for experiments on UBFC and PURE. Finally, on Countix and LST, we employ ResNet-3D-18 [64, 65] as our backbone network. Implementations details are in Appendix A.2.

Table 2-6: **Feature evaluation results on Countix.**

	FFT		1-NN	
Metrics	MAE↓	GM↓	MAE↓	GM↓
SimCLR [36]	3.90	2.26	4.43	3.19
MoCo v2 [37]	3.75	2.18	3.96	3.04
BYOL [63]	3.26	1.87	3.72	2.66
CVRL [39]	2.81	1.38	3.15	2.12
SimPer	**2.06**	**0.98**	**2.76**	**1.84**
Gains	**+0.75**	**+0.40**	**+0.99**	**+0.28**

Table 2-7: **Feature evaluation results on LST.**

	Linear Probing		
Metrics	MAE↓	MAPE↓	ρ↑
SimCLR [36]	5.12	0.20	0.89
MoCo v2 [37]	5.16	0.20	0.89
BYOL [63]	5.71	0.24	0.86
CVRL [39]	4.88	**0.18**	**0.91**
SimPer	**4.84**	**0.18**	0.90
Gains	**+0.04**	**+0.00**	**-0.01**

Baselines. We compare `SimPer` to SOTA SSL methods, including SimCLR [36], MoCo v2 [37], BYOL [63], and CVRL [39], as well as a supervised learning counterpart. We provide detailed descriptions in Appendix A.2.1.

Evaluation Metrics. To assess the prediction of continuous targets (e.g., frequency, counts), we use common metrics for regression, such as the mean-average-error (MAE), mean-average-percentage-error (MAPE), Pearson correlation (ρ), and error Geometric Mean (GM) [2].

2.3.1 Main Results

We report the main results in this section for all datasets. Complete training details, hyper-parameter settings, and additional results are provided in Appendix A.2 and A.3.

Feature Evaluation. Following the literature [37, 36], we first evaluate the representations learned by different methods. For dense prediction task (e.g., LST), we use the *linear probing* protocol by training a linear regressor on top of the fixed features. For tasks whose targets are frequency information, we directly evaluate the learned features using a Fourier transform (**FFT**) and a nearest neighbor classifier (**1-NN**). Table 2-2, 2-3, 2-4, 2-5, 2-6, 2-7 show the feature evaluation results of `SimPer` compared to SOTA SSL methods. As the tables confirm, across different datasets with various common tasks, `SimPer` is able to learn better representations that achieve the best performance. Furthermore, in certain datasets, the relative improvements are even larger than 50%.

Fine-tuning. Practically, to harness the power of pre-trained representations, fine-tuning the whole network with the encoder initialized using pre-trained weights is a widely

Table 2-8: **Fine-tune evaluation results on all datasets.** We first pre-train the feature encoder using different SSL methods, then fine-tune the whole network initialized with the pre-trained weights.

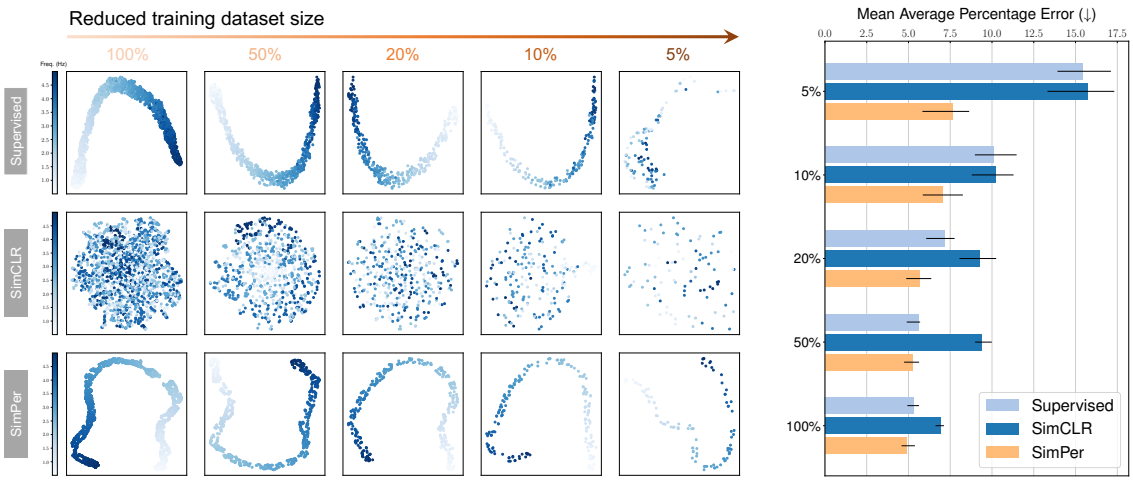
	RotatingDigits		SCAMPS		UBFC		PURE		Countix		LST	
Metrics	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	GM$^{\downarrow}$	MAE$^{\downarrow}$	$\rho^{\uparrow}$
SUPERVISED	0.72	28.96	3.61	5.33	5.13	4.72	4.25	4.93	1.50	0.73	1.54	**0.96**
SIMCLR [36]	0.69	26.54	4.96	6.92	5.32	4.96	4.86	5.32	1.58	0.80	1.54	0.95
MOCO V2 [37]	0.64	24.73	5.33	7.24	5.05	4.64	4.97	5.60	1.54	0.79	1.53	0.95
BYOL [63]	0.39	20.91	3.49	5.27	5.51	5.07	4.28	4.97	1.47	0.71	1.62	0.92
CVRL [39]	0.34	18.82	5.52	7.34	5.07	4.70	4.19	4.71	1.48	0.71	1.49	**0.96**
SIMPER	**0.20**	**14.33**	**3.27**	**4.89**	**4.24**	**3.97**	**3.89**	**4.01**	**1.33**	**0.59**	**1.47**	**0.96**
GAINS VS. SUPERVISED	**+0.52**	**+14.63**	**+0.34**	**+0.44**	**+0.89**	**+0.75**	**+0.36**	**+0.92**	**+0.17**	**+0.14**	**+0.07**	**+0.00**

adopted approach [37]. To evaluate whether `SimPer` pre-training is helpful for each downstream task, we fine-tune the whole network and compare the final performance. The details of the setup for each dataset and algorithm can be found in Appendix A.2. As Table 2-8 confirms, across different datasets, `SimPer` consistently outperforms all other SOTA SSL methods, and obtains better results compared to the supervised baseline. This demonstrates that `SimPer` is able to capture meaningful periodic information that is beneficial to the downstream tasks.

■ 2.3.2 Data Efficiency

In real-world periodic learning applications, data is often prohibitively expensive to obtain. To study the data efficiency of `SimPer`, we manually reduce the overall size of RotatingDigits, and plot the representations learned as well as the final fine-tuning accuracy of different methods in Fig. 2-4.

As the figure confirms, when the dataset size is large (e.g., using 100% of the data), both supervised learning baseline and `SimPer` can learn good representations (Fig. 2-4(a)) and achieve low test errors (Fig. 2-4(b)). However, when the training dataset size becomes smaller, the learned representations using supervised learning get worse, and eventually lose the frequency information and resolution when only 5% of the data is available. Correspondingly, the final error in this extreme case also becomes much higher. In contrast, even with small number of training data, `SimPer` can consistently learn the periodic information and maintain high frequency resolution, with significant performance gains especially when the available data amount is small.



(a) Learned representations under different training dataset sizes (b) Quantitative error changes

Figure 2-4: **Data efficiency analysis of `SimPer`. (a)** Learned representations of different algorithms on RotatingDigits when training dataset size reduces from 100% to 5%. **(b)** The quantitative MAPE errors on SCAMPS with varying training dataset sizes. Complete quantitative results are provided in Appendix A.3.1.

Table 2-9: **Transfer learning results.**

	UBFC → PURE		PURE → UBFC	
Metrics	MAE↓	MAPE↓	MAE↓	MAPE↓
SUPERVISED	7.83	8.85	3.15	3.11
SIMCLR	7.86	8.79	3.46	3.80
SIMPER	**6.46**	**6.98**	**2.76**	**2.38**
GAINS	**+1.37**	**+1.87**	**+0.39**	**+0.73**

■ 2.3.3 Transfer Learning

We evaluate whether the self-supervised representations are transferable across datasets. We use UBFC and PURE, which share the same prediction task. Following [36], we fine-tune the pre-trained model on the new dataset, and compare the performance across both SSL and supervised methods. Table 2-9 reports the results, where in both cases, `SimPer` is able to achieve better final performance compared to supervised and SSL baselines, showing its ability to learn transferable periodic representations across different datasets.

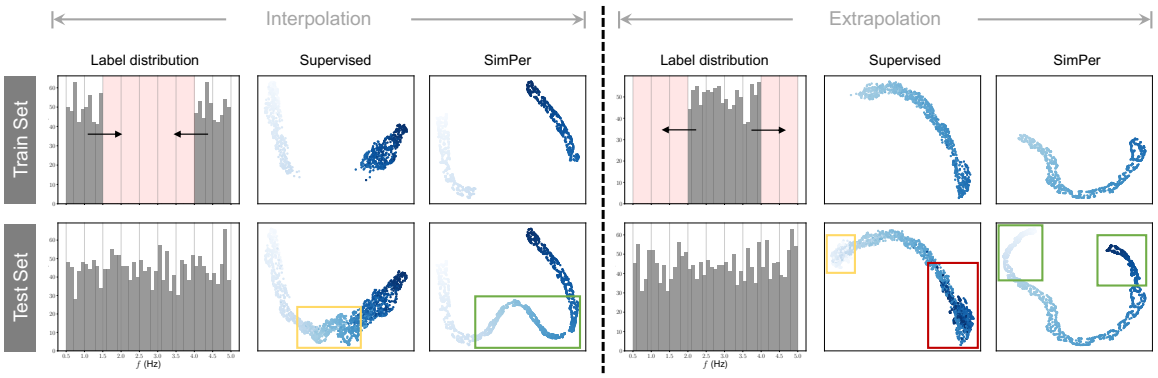


Figure 2-5: **Zero-shot generalization analysis.** We create training sets with missing target frequencies and keep test sets evenly distributed across the target range. Green regions indicate successful generalization with high frequency resolution. Yellow regions indicate successful generalization but with low frequency resolution. Red regions represent failed generalization. `SimPer` learns robust representations that generalize to unseen targets.

Table 2-10: **Mean absolute error (MAE) results for zero-shot generalization analysis.**

	Interpolation		**Extrapolation**	
	Seen	Unseen	Seen	Unseen
SUPERVISED	0.09	0.85	0.03	1.74
SIMPER	**0.05**	**0.07**	**0.02**	**0.02**
GAINS	**+0.04**	**+0.78**	**+0.01**	**+1.72**

■ 2.3.4 Zero-shot Generalization to Unseen Targets

Given the continuous nature of the frequency domain, periodic learning tasks can (and almost certainly will) have unseen frequency targets during training, which motivates the need for target (frequency) extrapolation and interpolation. To investigate **zero-shot** generalization to unseen targets, we manually create training sets that have certain missing targets (Fig. 2-5), while making the test sets evenly distributed across the target range. As Fig. 2-5 confirms, in the interpolation case, both supervised learning and `SimPer` can successfully interpolate the missing targets. However, the quality of interpolation varies: For supervised learning, the frequency resolution is low within the interpolation range, resulting in mixed representations for a wide missing range. In contrast, `SimPer` learns better representations with higher frequency resolution, which has desirable discriminative properties.

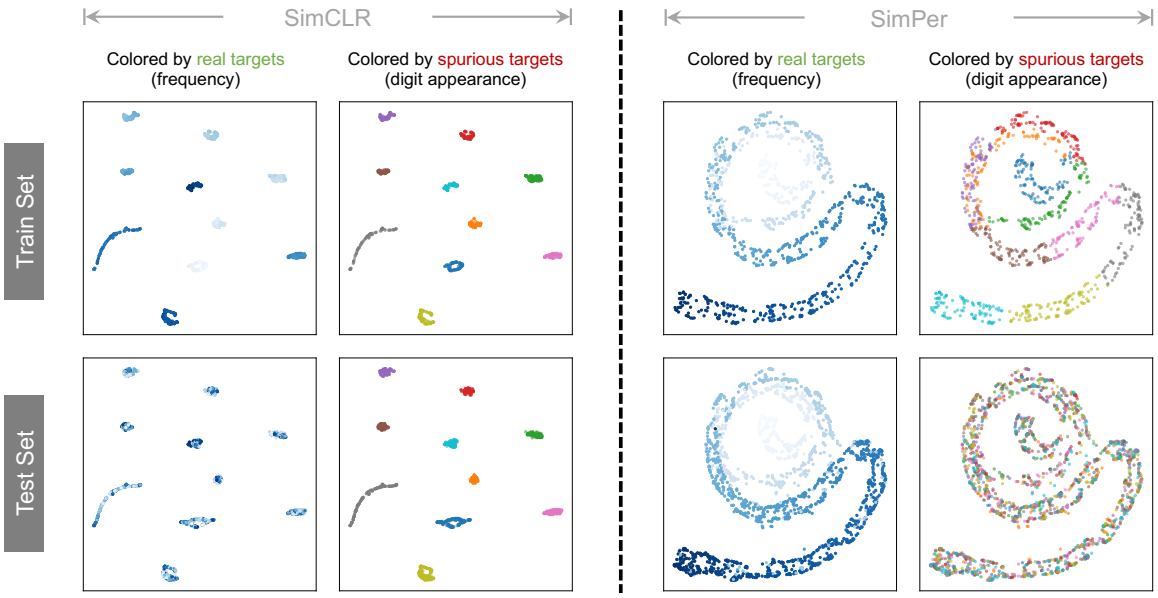


Figure 2-6: **Robustness to spurious correlations.** We make the target frequency spuriously correlated with digit appearances in training set, while removing this correspondence in test set. `SimPer` is able to capture underlying periodic information & learn robust representations that generalize. Quantitative results are in Appendix A.3.3.

Furthermore, in the extrapolation case, in the lower frequency range, both methods extrapolate reasonably well, with `SimPer` capturing a higher frequency resolution. However, when extrapolating to a higher frequency range, the supervised baseline completely fails to generalize, with learned features largely overlapping with the existing frequency targets in the training set. In contrast, `SimPer` is able to generalize robustly even for the higher unseen frequency range, demonstrating its effectiveness of generalization to distribution shifts and unseen targets. Quantitative results in Table 2-10 confirm the observations.

■ 2.3.5 Robustness to Spurious Correlations

We show that `SimPer` is able to deal with spurious correlations that arise in data, while existing SSL methods often fail to learn generalizable features. Specifically, RotatingDigits dataset naturally has a spurious target: the digit appearance (number). We further enforce this information by coloring different digits with different colors as in [66]. We then construct a spuriously correlated training set by assigning a unique rotating frequency range to a specific digit, i.e., $[0.5\text{Hz}, 1\text{Hz}]$ for digit 0, $[1\text{Hz}, 1.5\text{Hz}]$ for digit 1, etc, while removing the spurious correlations in test set.

As Fig. 2-6 verifies, SimCLR is easy to learn information that is spuriously correlated in

the training data, but not the actual target of interest (frequency). As a result, the learned representations do not generalize. In contrast, SimPer learns the underlying frequency information even in the presence of strong spurious correlations, demonstrating its ability to learn robust representations that generalize.

2.3.6 Further Analysis and Ablation Studies

Amount of labeled data for fine-tuning (Appendix A.3.2). We show that when the amount of labeled data is limited for fine-tuning, SimPer still substantially outperforms baselines by a large margin, achieving a 67% relative improvement in MAE even when the labeled data fraction is only 5%.

Ablation: Frequency augmentation range (Appendix A.3.4). We study the effects of different speed (frequency) augmentation ranges when creating periodicity-variant views (Table A-5). While a proper range can lead to certain gains, SimPer is reasonably robust to different choices.

Ablation: Number of augmented views (Appendix A.3.4). We investigate the influence of different number of augmented views (i.e., M) in SimPer. Interestingly, we find SimPer is surprisingly robust to different M in a given range (Table A-6), where larger M often delivers better results.

Ablation: Choices of different similarity metrics (Appendix A.3.4). We explore the effects of different periodic similarity measures in SimPer, where we show that SimPer is robust to all aforementioned periodic similarity measures, achieving similar performances (Table A-7).

Ablation: Effectiveness of generalized contrastive loss (Appendix A.3.4). We confirm the effectiveness of the generalized contrastive loss by showing its consistent performance gains across all six datasets, as compared to the vanilla InfoNCE loss (Table A-8).

2.4 Limitations and Broader Impacts

Limitations. There are some limitations to our approach in its current form. The SimPer features learnt in some cases were not highly effective without certain fine-tuning on a downstream task. This may be explained by the fact that some videos may contain mul-

tiple periodic processes (e.g., pulse/PPG, breathing, blinking, etc.). A pure SSL approach will learn features related to all these periodic signals, but not information that is specific to any one. One practical solution for this limitation could be incorporating the *frequency priors* of the targets of interest. Precisely, one can filter out unrelated frequencies during `SimPer` pre-training to force the network to learn features that are constrained within a certain frequency range. We leave this part as future work.

Broader Impacts. While our methods are generic to tasks that involve learning periodic signals, we have selected some specific tasks on which to demonstrate their efficacy more concretely. The measurement of health information from videos has tremendous potential for positive impact, helping to lower the barrier to access to frequent measurement and reduce the discomfort or inconvenience caused by wearable devices. However, there is the potential for negative applications of such technology. Whether by negligence, or bad intention, unobtrusive measurement could be used to measure information covertly and without the consent of a user. Such an application would be unethical and would also violate laws in many parts of the world[1]. It is important that the same stringent measures applied to traditional medical sensing are also applied to video-based methods. We will be releasing code for our approach under a Responsible AI License (RAIL) [67] to help practically mitigate unintended negative behavioral uses of the technology while still making the code available.

■ 2.5 Summary

In this chapter, we present `SimPer`, a simple and effective SSL framework for learning periodic information from data. `SimPer` develops customized periodicity-variant and invariant augmentations, periodic feature similarity, and a generalized contrastive loss to exploit periodic inductive biases. Extensive experiments on different datasets over various real-world applications verify the superior performance of `SimPer`, highlighting its intriguing properties such as better efficiency, robustness & generalization.

[1] https://www.ilga.gov/legislation/ilcs/ilcs3.asp?ActID=3004&ChapterID=57

CHAPTER 3
Delving into Deep Imbalanced Regression

Data imbalance is ubiquitous and inherent in the real world. Rather than preserving an ideal uniform distribution over each category, the data often exhibit skewed distributions with a long tail [68, 69], where certain target values have significantly fewer observations. This phenomenon poses great challenges for deep recognition models, and has motivated many prior techniques for addressing data imbalance [70, 71, 69, 72, 73].

Existing solutions for learning from imbalanced data, however, focus on targets with categorical indices, i.e., the targets are different classes. However, many real-world tasks involve continuous and even infinite target values. For example, in vision applications, one needs to infer the age of different people based on their visual appearances, where age is a continuous target and can be highly imbalanced. Treating different ages as distinct classes is unlikely to yield the best results because it does not take advantage of the similarity between people with nearby ages. Similar issues happen in medical applications since many health metrics including heart rate, blood pressure, and oxygen saturation, are continuous and often have skewed distributions across patient populations.

In this work, we systematically investigate *Deep Imbalanced Regression* (DIR) arising in real-world settings (see Fig. 3-1). We define DIR as learning continuous targets from natural imbalanced data, dealing with potentially missing data for certain target values, and generalizing to a test set that is balanced over the entire range of continuous target

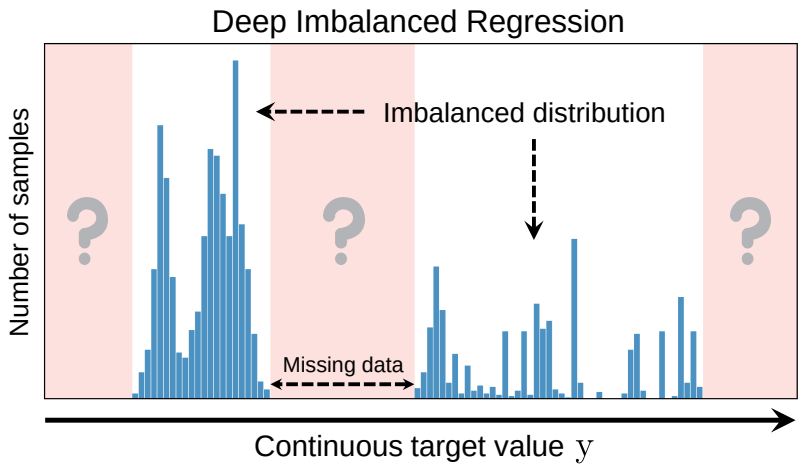


Figure 3-1: **Overview of Deep Imbalanced Regression (DIR).** DIR aims to learn from imbalanced data with continuous targets, tackle potential missing data for certain regions, and generalize to the entire target range.

values. This definition is analogous to the class imbalance problem [69], but focuses on the continuous setting.

DIR brings new challenges distinct from its classification counterpart. First, given continuous (potentially infinite) target values, the hard boundaries between classes no longer exist, causing ambiguity when directly applying traditional imbalanced classification methods such as re-sampling and re-weighting. Moreover, continuous labels inherently possess a meaningful distance between targets, which has implication for how we should interpret data imbalance. For example, say two target labels t_1 and t_2 have a small number of observations in training data. However, t_1 is in a highly represented neighborhood (i.e., there are many samples in the range $[t_1 - \Delta, t_1 + \Delta]$), while t_2 is in a weakly represented neighborhood. In this case, t_1 does not suffer from the same level of imbalance as t_2. Finally, unlike classification, certain target values may have no data at all, which motivates the need for target extrapolation & interpolation.

In this chapter, we propose two simple yet effective methods for addressing DIR: label distribution smoothing (LDS) and feature distribution smoothing (FDS). A key idea underlying both approaches is to leverage the similarity between nearby targets by employing a kernel distribution to perform explicit distribution smoothing in the label and feature spaces. Both techniques can be easily embedded into existing deep networks and allow optimization in an end-to-end fashion. We verify that our techniques not only successfully calibrate for the intrinsic underlying imbalance, but also provide large and consistent gains

when combined with other methods.

To support practical evaluation of imbalanced regression, we curate and benchmark large-scale DIR datasets for common real-world tasks in computer vision, natural language processing, and healthcare. They range from single-value prediction such as age, text similarity score, health condition score, to dense-value prediction such as depth. We further set up benchmarks for proper DIR performance evaluation.

In this chapter, our contributions are as follows: (i) We formally define the DIR task as learning from imbalanced data with continuous targets, and generalizing to the entire target range. DIR provides thorough and unbiased evaluation of learning algorithms in practical settings. (ii) We develop two simple, effective, and interpretable algorithms for DIR, LDS and FDS, which exploit the similarity between nearby targets in both label and feature space. (iii) We curate benchmark DIR datasets in different domains: computer vision, natural language processing, and healthcare. We set up strong baselines as well as benchmarks for proper DIR performance evaluation. (iv) Extensive experiments on large-scale DIR datasets verify the consistent and superior performance of our strategies.

3.1 Related Work

Imbalanced Classification. Much prior work has focused on the imbalanced classification problem (also referred to as long-tailed recognition [69]). Past solutions can be divided into data-based and model-based solutions: Data-based solutions either over-sample the minority class or under-sample the majority [74, 75, 76]. For example, SMOTE generates synthetic samples for minority classes by linearly interpolating samples in the same class [74]. Model-based solutions include re-weighting or adjusting the loss function to compensate for class imbalance [70, 77, 72, 71, 78], and leveraging relevant learning paradigms, including transfer learning [79], metric learning [80], meta-learning [81], and two-stage training [82]. Recent studies have also discovered that semi-supervised learning and self-supervised learning lead to better imbalanced classification results [22]. In contrast to these past work, we identify the limitations of applying class imbalance methods to regression problems, and introduce new techniques particularly suitable for learning continuous target values.

Imbalanced Regression. Regression over imbalanced data is not as well explored. Most of

the work on this topic is a direct adaptation of the SMOTE algorithm to regression scenarios [83, 84, 85]. Synthetic samples are created for pre-defined rare target regions by either directly interpolating both inputs and targets [83], or using Gaussian noise augmentation [84]. A bagging-based ensemble method that incorporates multiple data pre-processing steps has also been introduced [85]. However, there exist several intrinsic drawbacks for these methods. First, they fail to take the distance between targets into account, and rather heuristically divide the dataset into rare and frequent sets, then plug in classification-based methods. Moreover, modern data is of extremely high dimension (e.g., images and physiological signals); linear interpolation of two samples of such data does not lead to meaningful new synthetic samples. Our methods are intrinsically different from past work in their approach. They can be combined with existing methods to improve their performance, as we show in Sec. 3.3. Further, our approaches are tested on large-scale real-world datasets in computer vision, NLP, and healthcare.

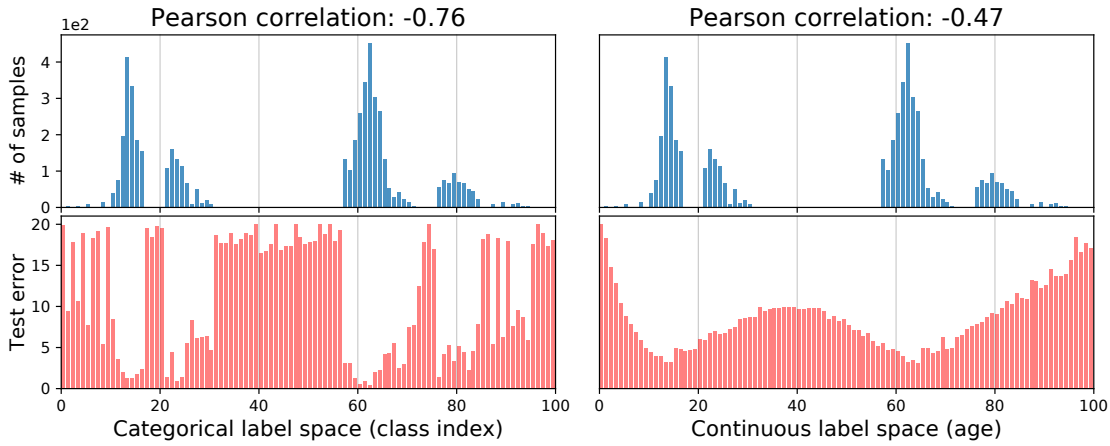
3.2 Methods

Problem Setting. Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be a training set, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the input and $y_i \in \mathbb{R}$ is the label, which is a continuous target. We introduce an additional structure for the label space $\mathcal{Y}$, where we divide $\mathcal{Y}$ into B groups (bins) with equal intervals, i.e., $[y_0, y_1), [y_1, y_2), \ldots, [y_{B-1}, y_B)$. Throughout the chapter, we use $b \in \mathcal{B}$ to denote the group index of the target value, where $\mathcal{B} = \{1, \ldots, B\} \subset \mathbb{Z}^+$ is the index space. In practice, the defined bins reflect a minimum resolution we care for grouping data in a regression task. For instance, in age estimation, we could define $\delta y \triangleq y_{b+1} - y_b = 1$, showing a minimum age difference of 1 is of interest. Finally, we denote $\mathbf{z} = f(\mathbf{x}; \theta)$ as the feature for $\mathbf{x}$, where $f(\mathbf{x}; \theta)$ is parameterized by a deep neural network model with parameter θ. The final prediction $\widehat{y}$ is given by a regression function $g(\cdot)$ that operates over $\mathbf{z}$.

3.2.1 Label Distribution Smoothing

We start by showing an example to demonstrate the difference between classification and regression when imbalance comes into the picture.

Motivating Example. We employ two datasets: (1) CIFAR-100 [86], which is a 100-class classification dataset, and (2) the IMDB-WIKI dataset [87], which is a large-scale image



(a) CIFAR-100 (subsampled) (b) IMDB-WIKI (subsampled)

Figure 3-2: **Comparison on the test error distribution (bottom) using same training label distribution (top) on two different datasets.** (a) CIFAR-100, a classification task with categorical label space. (b) IMDB-WIKI, a regression task with continuous label space.

dataset for age estimation from visual appearance. The two datasets have intrinsically different label space: CIFAR-100 exhibits *categorical label space* where the target is class index, while IMDB-WIKI has a *continuous label space* where the target is age. We limit the age range to $0 \sim 99$ so that the two datasets have the same label range, and subsample them to simulate data imbalance, while ensuring they have exactly the same label density distribution (Fig. 3-2). We make both test sets balanced. We then train a plain ResNet-50 model on the two datasets, and plot their test error distributions.

We observe from Fig. 3-2(a) that the error distribution *correlates* with label density distribution. Specifically, the test error as a function of class index has a high negative Pearson correlation with the label density distribution (i.e., -0.76) in the categorical label space. The phenomenon is expected, as majority classes with more samples are better learned than minority classes. Interestingly however, as Fig. 3-2(b) shows, the error distribution is very different for IMDB-WIKI with continuous label space, even when the label density distribution is the same as CIFAR-100. In particular, the error distribution is much smoother and no longer correlates well with the label density distribution (-0.47).

The reason why this example is interesting is that all imbalanced learning methods, directly or indirectly, operate by compensating for the imbalance in the *empirical* label density distribution. This works well for class imbalance, but for continuous labels the empirical density does not accurately reflect the imbalance as seen by the neural network. Hence,

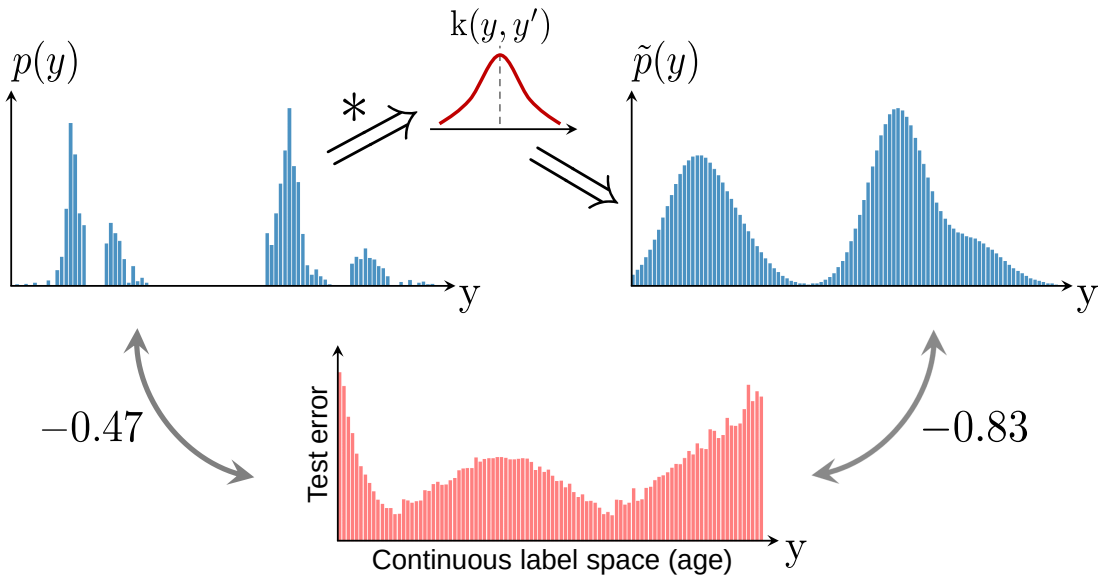


Figure 3-3: **Label distribution smoothing (LDS).** LDS convolves a symmetric kernel with the empirical label density to estimate the effective label density distribution that accounts for the continuity of labels.

compensating for data imbalance based on empirical label density is inaccurate for the continuous label space.

LDS for Imbalanced Data Density Estimation. The above example shows that, in the continuous case, the empirical label distribution does not reflect the real label density distribution. This is because of the dependence between data samples at nearby labels (e.g., images of close ages). In fact, there is a significant literature in statistics on how to estimate the expected density in such cases [88]. Thus, Label Distribution Smoothing (LDS) advocates the use of kernel density estimation to learn the effective imbalance in datasets that corresponds to continuous targets.

LDS convolves a symmetric kernel with the empirical density distribution to extract a kernel-smoothed version that accounts for the overlap in information of data samples of nearby labels. A symmetric kernel is any kernel that satisfies: $\mathrm{k}(y, y') = \mathrm{k}(y', y)$ and $\nabla_y \mathrm{k}(y, y') + \nabla_{y'} \mathrm{k}(y', y) = 0$, $\forall y, y' \in \mathcal{Y}$. Note that a Gaussian or a Laplacian kernel is a symmetric kernel, while $\mathrm{k}(y, y') = yy'$ is not. The symmetric kernel characterizes the similarity between target values y' and any y w.r.t. their distance in the target space. Thus, LDS computes the *effective label density distribution* as:

$$\widetilde{p}(y') \triangleq \int_{\mathcal{Y}} \mathrm{k}(y, y') p(y) dy, \tag{3.1}$$

where $p(y)$ is the number of appearances of label of y in the training data, and $\widetilde{p}(y')$ is the effective density of label y'.

Fig. 3-3 illustrates LDS and how it smooths the label density distribution. Further, it shows that the resulting label density computed by LDS correlates well with the error distribution (-0.83). This demonstrates that LDS captures the real imba-lance that affects regression problems.

Now that the effective label density is available, techniques for addressing class imbalance problems can be directly adapted to the DIR context. For example, a straightforward adaptation can be the cost-sensitive re-weighting method, where we re-weight the loss function by multiplying it by the inverse of the LDS estimated label density for each target. We show in Sec. 3.3 that LDS can be seamlessly incorporated with a wide range of techniques to boost DIR performance.

■ 3.2.2 Feature Distribution Smoothing

We are motivated by the intuition that continuity in the target space should create a corresponding continuity in the feature space. That is, if the model works properly and the data is balanced, one expects the feature statistics corresponding to nearby targets to be close to each other.

Motivating Example. We use an illustrative example to highlight the impact of data imbalance on feature statistics in DIR. Again, we use a plain model trained on the images in the IMDB-WIKI dataset to infer a person's age from visual appearance. We focus on the learned feature space, i.e., $\mathbf{z}$. We use a minimum bin size of 1, i.e., $y_{b+1} - y_b = 1$, and group features with the same target value in the same bin. We then compute the feature statistics (i.e., mean and variance) with respect to the data in each bin, which we denote as $\{\boldsymbol{\mu}_b, \boldsymbol{\sigma}_b\}_{b=1}^{B}$. To visualize the similarity between feature statistics, we select an anchor bin b_0, and calculate the cosine similarity of the feature statistics between b_0 and all other bins. The results are summarized in Fig. 3-4 for $b_0 = 30$. The figure also shows the regions with different data densities using the colors purple, yellow, and pink.

Fig. 3-4 shows that the feature statistics around $b_0 = 30$ are highly similar to their values at $b_0 = 30$. Specifically, the cosine similarity of the feature mean and feature variance for all bins between age 25 and 35 are within a few percent from their values at age 30 (the anchor age). Further, the similarity gets higher for tighter ranges around the anchor. Note

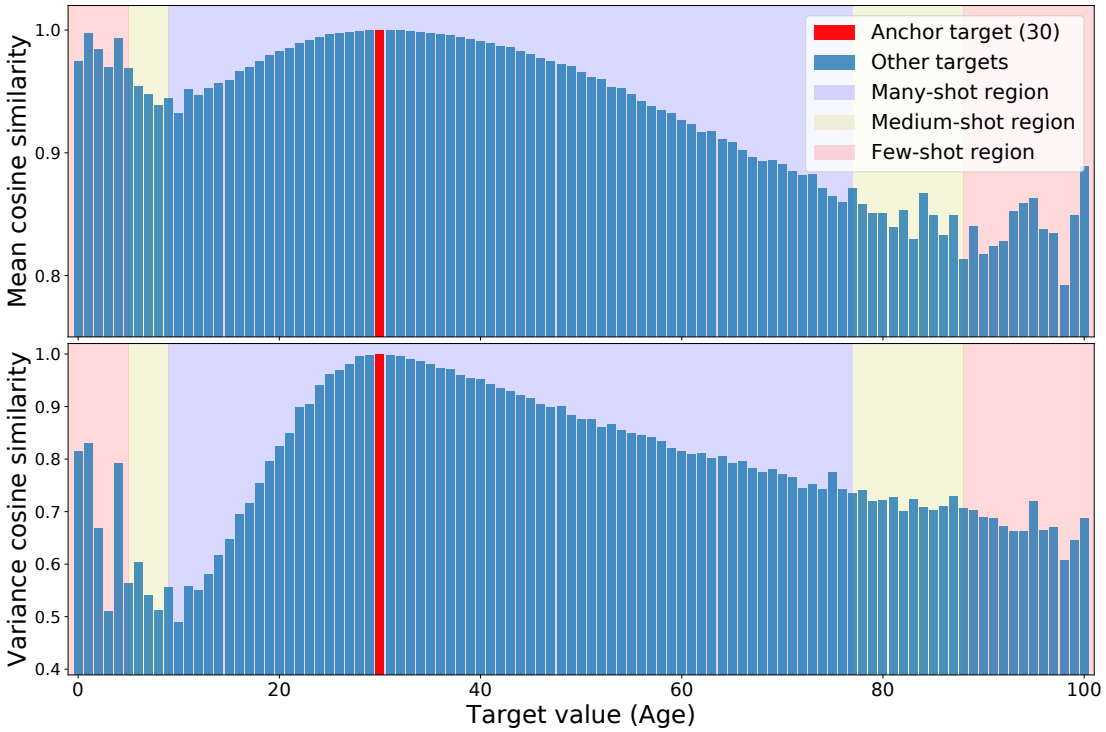


Figure 3-4: **Feature statistics similarity. Top:** Cosine similarity of the feature mean at a particular age w.r.t. its value at the anchor age. **Bottom:** Cosine similarity of the feature variance at a particular age w.r.t. its value at the anchor age. The color of the background refers to the data density in a particular target range. The figure shows that nearby ages have close similarities; However, it also shows that there is unjustified similarity between images at ages 0 to 6 and age 30, due to data imbalance.

that bin 30 falls in the high shot region. In fact, it is among the few bins that have the most samples. So, the figure confirms the intuition that when there is enough data, and for continuous targets, the feature statistics are similar to nearby bins. Interestingly, the figure also shows the problem with regions that have very few data samples, like the age range 0 to 6 years (shown in pink). Note that the mean and variance in this range show unexpectedly high similarity to age 30. In fact, it is shocking that the feature statistics at age 30 are more similar to age 1 than age 17. This unjustified similarity is due to data imbalance. Specifically, since there are not enough images for ages 0 to 6, this range thus inherits its priors from the range with the maximum amount of data, which is the range around age 30.

FDS Algorithm. Inspired by these observations, we propose feature distribution smoothing (FDS), which performs distribution smoothing on the feature space, i.e., transfers the feature statistics between nearby target bins. This procedure aims to calibrate the poten-

tially biased estimates of feature distribution, especially for underrepresented target values (e.g., medium- and few-shot groups) in training data.

FDS is performed by first estimating the statistics of each bin. Without loss of generality, we substitute variance with covariance to reflect also the relationship between the various feature elements within $\mathbf{z}$:

$$\boldsymbol{\mu}_b = \frac{1}{N_b} \sum_{i=1}^{N_b} \mathbf{z}_i, \tag{3.2}$$

$$\boldsymbol{\Sigma}_b = \frac{1}{N_b - 1} \sum_{i=1}^{N_b} (\mathbf{z}_i - \boldsymbol{\mu}_b)(\mathbf{z}_i - \boldsymbol{\mu}_b)^\top, \tag{3.3}$$

where N_b is the total number of samples in b-th bin. Given the feature statistics, we employ again a symmetric kernel $\mathrm{k}(y_b, y_{b'})$ to smooth the distribution of the feature mean and covariance over the target bins $\mathcal{B}$. This results in a smoothed version of the statistics:

$$\widetilde{\boldsymbol{\mu}}_b = \sum_{b' \in \mathcal{B}} \mathrm{k}(y_b, y_{b'}) \boldsymbol{\mu}_{b'}, \tag{3.4}$$

$$\widetilde{\boldsymbol{\Sigma}}_b = \sum_{b' \in \mathcal{B}} \mathrm{k}(y_b, y_{b'}) \boldsymbol{\Sigma}_{b'}. \tag{3.5}$$

With both $\{\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$ and $\{\widetilde{\boldsymbol{\mu}}_b, \widetilde{\boldsymbol{\Sigma}}_b\}$, we then follow the standard whitening and re-coloring procedure [89] to calibrate the feature representation for each input sample:

$$\widetilde{\mathbf{z}} = \widetilde{\boldsymbol{\Sigma}}_b^{\frac{1}{2}} \boldsymbol{\Sigma}_b^{-\frac{1}{2}} (\mathbf{z} - \boldsymbol{\mu}_b) + \widetilde{\boldsymbol{\mu}}_b. \tag{3.6}$$

We integrate FDS into deep networks by inserting a feature calibration layer after the final feature map. To train the model, we employ a *momentum update* of the running statistics $\{\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$ across each epoch. Correspondingly, the smoothed statistics $\{\widetilde{\boldsymbol{\mu}}_b, \widetilde{\boldsymbol{\Sigma}}_b\}$ are updated across different epochs but fixed within each training epoch. The momentum update, which performs an exponential moving average (EMA) of running statistics, results in more stable and accurate estimations of the feature statistics during training. The calibrated features $\widetilde{\mathbf{z}}$ are then passed to the final regression function and used to compute the loss.

We note that FDS can be integrated with any neural network model, as well as any past work on improving label imbalance. In Sec. 3.3, we integrate FDS with a variety of prior

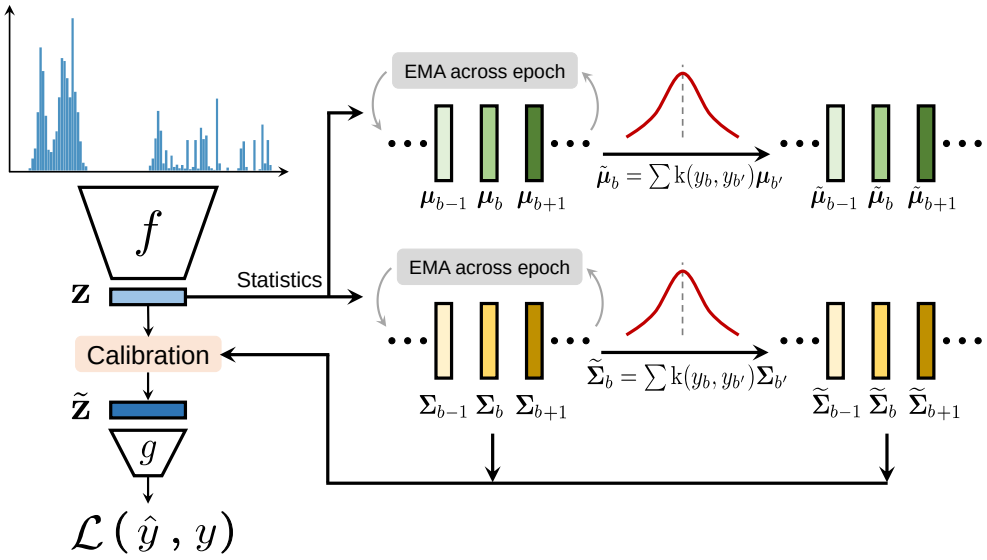


Figure 3-5: **Feature distribution smoothing (FDS).** FDS introduces a feature calibration layer that uses kernel smoothing to smooth the distributions of feature mean and covariance over the target space.

techniques for addressing data imbalance, and demonstrate that it consistently improves performance.

■ 3.3 Benchmarking DIR

Datasets. We curate five DIR benchmarks that span computer vision, natural language processing, and healthcare. Fig. 3-6 shows the label density distribution of these datasets, and their level of imbalance.

- *IMDB-WIKI-DIR (age)*: We construct IMDB-WIKI-DIR using the IMDB-WIKI dataset [87], which contains 523.0K face images and the corresponding ages. We filter out unqualified images, and manually construct balanced validation and test set over the supported ages. The length of each bin is 1 year, with a minimum age of 0 and a maximum age of 186. The number of images per bin varies between 1 and 7149, exhibiting significant data imbalance. Overall, the curated dataset has 191.5K images for training, 11.0K images for validation and testing.

- *AgeDB-DIR (age)*: AgeDB-DIR is constructed in a similar manner from the AgeDB dataset [90]. It contains 12.2K images for training, with a minimum age of 0 and a maximum age of 101, and maximum bin density of 353 images and minimum bin density of 1. The

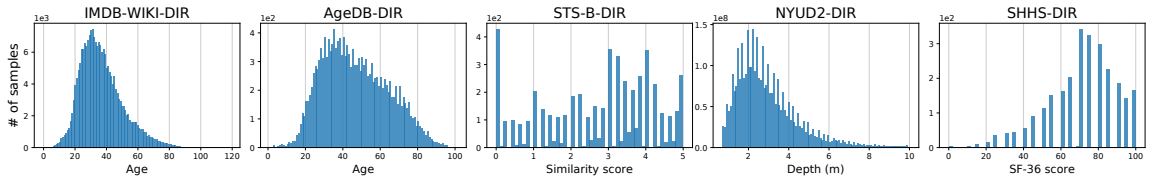


Figure 3-6: **Overview of training set label distribution for five DIR datasets.** They range from single-value prediction such as age, textual similarity score, and health condition score, to dense-value prediction such as depth estimation. More details are provided in Appendix B.1.

validation and test set are balanced with 2.1K images.

- *STS-B-DIR (text similarity score)*: We construct STS-B-DIR from the Semantic Textual Similarity Benchmark [91, 92], which is a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Each pair is annotated by multiple annotators with an averaged continuous similarity score from 0 to 5. From the original training set of 7.2K pairs, we create a training set with 5.2K pairs, and balanced validation set and test set of 1K pairs each. The length of each bin is 0.1.

- *NYUD2-DIR (depth)*: We create NYUD2-DIR based on the NYU Depth Dataset V2 [93], which provides images and depth maps for different indoor scenes. The depth maps have an upper bound of 10 meters and we set the bin length as 0.1 meter. Following standard practices [94, 95], we use 50K images for training and 654 images for testing. We randomly select 9357 test pixels for each bin to make the test set balanced.

- *SHHS-DIR (health condition score)*: We create SHHS-DIR based on the SHHS dataset [96], which contains full-night Polysomnography (PSG) from 2651 subjects. Available PSG signals include Electroencephalography (EEG), Electrocardiography (ECG), and breathing signals (airflow, abdomen, and thorax), which are used as inputs. The dataset also includes the 36-Item Short Form Health Survey (SF-36) [97] for each subject, where a General Health score is extracted. The score is used as the target value with a minimum score of 0 and maximum of 100.

Network Architectures. We employ ResNet-50 [64] as our backbone network for IMDB-WIKI-DIR and AgeDB-DIR. Following [92], we adopt the same BiLSTM + GloVe word embeddings baseline for STS-B-DIR. For NYUD2-DIR, we use ResNet-50-based encoder-decoder architecture introduced in [94]. Finally, for SHHS-DIR, we use the same CNN-

RNN architecture with ResNet block for PSG signals as in [98].

Baselines. Since the literature has only a few proposals for DIR, in addition to past work on imbalanced regression [83, 84], we adapt a few imbalanced classification methods for regression, and propose a strong set of baselines. Below, we describe the baselines, and how we can combine LDS with each method. For FDS, it can be directly integrated with any baseline as a calibration layer, as described in Sec. 3.2.2.

- *Vanilla model*: We use term **VANILLA** to denote a model that does not include any technique for dealing with imbalanced data. To combine the vanilla model with LDS, we re-weight the loss function by multiplying it by the inverse of the LDS estimated density for each target bin.

- *Synthetic samples*: We choose existing methods for imbalanced regression, including **SMOTER** [83] and **SMOGN** [84]. SMOTER first defines frequent and rare regions using the original label density, and creates synthetic samples for pre-defined rare regions by linearly interpolating both inputs and targets. SMOGN further adds Gaussian noise to SMOTER. We note that LDS can be directly used for a better estimation of label density when dividing the target space.

- *Error-aware loss*: Inspired by the Focal loss [99] for classification, we propose a regression version called **Focal-R**, where the scaling factor is replaced by a continuous function that maps the absolute error into $[0, 1]$. Precisely, Focal-R loss based on L_1 distance can be written as $\frac{1}{n}\sum_{i=1}^{n}\sigma(|\beta e_i|)^{\gamma}e_i$, where e_i is the L_1 error for i-th sample, $\sigma(\cdot)$ is the `Sigmoid` function, and β, γ are hyper-parameters. To combine Focal-R with LDS, we multiply the loss with the inverse frequency of the estimated label density.

- *Two-stage training*: Following [82] where feature and classifier are decoupled and trained in two stages, we propose a regression version called regressor re-training (**RRT**), where in the first stage we train the encoder normally, and in the second stage freeze the encoder and re-train the regressor $g(\cdot)$ with inverse re-weighting. When adding LDS, the re-weighting in the second stage is based on the label density estimated through LDS.

- *Cost-sensitive re-weighting*: Since we divide the target space into finite bins, classic re-weighting methods can be directly plugged in. We adopt two re-weighting schemes based on the label distribution: inverse-frequency weighting (**INV**) and its square-root

Table 3-1: **Benchmarking results on IMDB-WIKI-DIR.**

Metrics	MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
SMOTER [83]	8.14	7.42	14.15	25.28	4.64	**4.30**	9.05	19.46
SMOGN [84]	8.03	**7.30**	14.02	25.93	4.63	**4.30**	8.74	20.12
SMOGN + **LDS**	8.02	7.39	13.71	23.22	4.63	4.39	8.71	15.80
SMOGN + **FDS**	8.03	7.35	14.06	23.44	4.65	4.33	8.87	16.00
SMOGN + **LDS** + **FDS**	**7.97**	7.38	**13.22**	**22.95**	**4.59**	4.39	**7.84**	**14.94**
FOCAL-R	7.97	7.12	15.14	26.96	4.49	4.10	10.37	21.20
FOCAL-R + **LDS**	7.90	**7.10**	14.72	25.84	**4.47**	**4.09**	10.11	19.14
FOCAL-R + **FDS**	7.96	7.14	14.71	26.06	4.51	4.12	10.16	19.56
FOCAL-R + **LDS** + **FDS**	**7.88**	**7.10**	**14.08**	**25.75**	**4.47**	4.11	**9.32**	**18.67**
RRT	7.81	7.07	14.06	25.13	4.35	4.03	8.91	16.96
RRT + **LDS**	7.79	7.08	13.76	24.64	4.34	**4.02**	8.72	16.92
RRT + **FDS**	**7.65**	**7.02**	12.68	23.85	**4.31**	4.03	7.58	16.28
RRT + **LDS** + **FDS**	**7.65**	7.06	**12.41**	**23.51**	**4.31**	4.07	**7.17**	**15.44**
SQINV	7.87	7.24	12.44	22.76	4.47	4.22	7.25	15.10
SQINV + **LDS**	7.83	7.31	**12.43**	22.51	4.42	4.19	7.00	13.94
SQINV + **FDS**	7.83	7.23	12.60	22.37	4.42	4.20	**6.93**	13.48
SQINV + **LDS** + **FDS**	**7.78**	**7.20**	12.61	**22.19**	**4.37**	**4.12**	7.39	**12.61**
OURS (BEST) VS. VANILLA	**+0.41**	**+0.21**	**+2.71**	**+4.14**	**+0.26**	**+0.15**	**+3.66**	**+7.85**

weighting variant (**SQINV**). When combining with LDS, instead of using the original label density, we use the LDS estimated target density.

Evaluation Process and Metrics. Following [69], we divide the target space into three disjoint subsets: *many-shot region* (bins with over 100 training samples), *medium-shot region* (bins with 20∼100 training samples), and *few-shot region* (bins with under 20 training samples), and report results on these subsets, as well as overall performance. We also refer to regions with no training samples as *zero-shot*, and investigate the ability of our techniques to generalize to zero-shot regions in Sec. 3.3.2. For metrics, we use common metrics for regression, such as the mean-average-error (MAE), mean-squared-error (MSE), and Pearson correlation. We further propose another metric, called error Geometric Mean (**GM**), and is defined as $(\prod_{i=1}^{n} e_i)^{\frac{1}{n}}$ for better prediction fairness.

■ 3.3.1 Main Results

We report the main results in this section for all DIR datasets. All training details, hyperparameter settings, and additional results are provided in Appendix B.2 and B.3.

Table 3-2: **Benchmarking results on AgeDB-DIR.**

Metrics	MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	7.77	6.62	9.55	13.67	5.05	4.23	7.01	10.75
SMOTER [83]	8.16	7.39	8.65	12.28	5.21	4.65	5.69	8.49
SMOGN [84]	8.26	7.64	9.01	12.09	5.36	4.90	6.19	8.44
SMOGN + **LDS**	7.96	7.44	8.64	11.77	5.03	4.68	5.69	7.98
SMOGN + **FDS**	8.06	7.52	8.75	11.89	5.02	4.66	5.63	8.02
SMOGN + **LDS** + **FDS**	**7.90**	**7.32**	**8.51**	**11.19**	**4.98**	**4.64**	**5.41**	**7.35**
FOCAL-R	7.64	6.68	9.22	13.00	4.90	4.26	6.39	9.52
FOCAL-R + **LDS**	7.56	**6.67**	8.82	12.40	4.82	4.27	5.87	8.83
FOCAL-R + **FDS**	7.65	6.89	8.70	**11.92**	4.83	4.32	5.89	**8.04**
FOCAL-R + **LDS** + **FDS**	**7.47**	6.69	**8.30**	12.55	**4.71**	**4.25**	**5.36**	8.59
RRT	7.74	6.98	8.79	11.99	5.00	4.50	5.88	8.63
RRT + **LDS**	7.72	7.00	8.75	11.62	4.98	4.54	5.71	8.27
RRT + **FDS**	7.70	**6.95**	8.76	11.86	4.82	**4.32**	5.83	8.08
RRT + **LDS** + **FDS**	**7.66**	6.99	**8.60**	**11.32**	**4.80**	4.42	**5.53**	**6.99**
SQINV	7.81	7.16	8.80	11.20	4.99	4.57	5.73	7.77
SQINV + **LDS**	7.67	**6.98**	8.86	10.89	4.85	4.39	5.80	7.45
SQINV + **FDS**	7.69	7.10	8.86	**9.98**	4.83	4.41	5.97	**6.29**
SQINV + **LDS** + **FDS**	**7.55**	7.01	**8.24**	10.79	**4.72**	**4.36**	**5.45**	6.79
OURS (BEST) VS. VANILLA	**+0.30**	**-0.05**	**+1.31**	**+3.69**	**+0.34**	**-0.02**	**+1.65**	**+4.46**

Inferring Age from Images: IMDB-WIKI-DIR & AgeDB-DIR. We report the performance of different methods in Table 3-1 and 3-2, respectively. For each dataset, we group the baselines into four sections to reflect their different strategies. First, as both tables indicate, when applied to modern high-dimensional data like images, SMOTER and SMOGN can actually degrade the performance in comparison to the vanilla model. Moreover, within each group, adding either LDS, FDS, or both leads to performance gains, while LDS + FDS often achieves the best results. Finally, when compared to the vanilla model, using our LDS and FDS maintains or slightly improves the performance overall and on the many-shot regions, while substantially boosting the performance for the medium-shot and few-shot regions.

Inferring Text Similarity Score: STS-B-DIR. Table 3-3 shows the results, where similar observations can be made on STS-B-DIR. Again, both SMOTER and SMOGN perform worse than the vanilla model. In contrast, both LDS and FDS consistently and substantially improve the results for various methods, especially in medium- and few-shot regions. The advantage is even more profound under *Pearson correlation*, which is commonly used for

Table 3-3: **Benchmarking results on STS-B-DIR.**

Metrics	MSE ↓				Pearson correlation (%) ↑			
Shot	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	0.974	0.851	1.520	0.984	74.2	72.0	62.7	75.2
SMOTER [83]	1.046	0.924	1.542	1.154	72.6	69.3	65.3	70.6
SMOGN [84]	0.990	0.896	1.327	1.175	73.2	70.4	65.5	69.2
SMOGN + **LDS**	0.962	0.880	1.242	1.155	74.0	71.5	65.2	69.8
SMOGN + **FDS**	0.987	0.945	**1.101**	1.153	73.0	69.6	**68.5**	69.9
SMOGN + **LDS** + **FDS**	**0.950**	**0.851**	1.327	**1.095**	**74.6**	**72.1**	65.9	**71.7**
FOCAL-R	0.951	0.843	1.425	0.957	74.6	72.3	61.8	76.4
FOCAL-R + **LDS**	0.930	**0.807**	1.449	0.993	**75.7**	**73.9**	62.4	75.4
FOCAL-R + **FDS**	**0.920**	0.855	**1.169**	1.008	75.1	72.6	**66.4**	74.7
FOCAL-R + **LDS** + **FDS**	0.940	0.849	1.358	**0.916**	74.9	72.2	66.3	**77.3**
RRT	0.964	0.842	1.503	0.978	74.5	72.4	62.3	75.4
RRT + **LDS**	0.916	0.817	1.344	0.945	75.7	73.5	64.1	76.6
RRT + **FDS**	0.929	0.857	**1.209**	1.025	74.9	72.1	**67.2**	74.0
RRT + **LDS** + **FDS**	**0.903**	**0.806**	1.323	**0.936**	**76.0**	**73.8**	65.2	**76.7**
INV	1.005	0.894	1.482	1.046	72.8	70.3	62.5	73.2
INV + **LDS**	0.914	0.819	1.319	0.955	75.6	73.4	63.8	76.2
INV + **FDS**	0.927	0.851	**1.225**	1.012	75.0	72.4	**66.6**	74.2
INV + **LDS** + **FDS**	**0.907**	**0.802**	1.363	**0.942**	**76.0**	**74.0**	65.2	**76.6**
OURS (BEST) VS. VANILLA	**+.071**	**+.049**	**+.419**	**+.068**	**+1.8**	**+2.0**	**+5.8**	**+2.1**

Table 3-4: **Benchmarking results on NYUD2-DIR.**

Metrics	RMSE ↓				δ_1 ↑			
Shot	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	1.477	0.591	0.952	2.123	0.677	0.777	0.693	0.570
VANILLA + **LDS**	1.387	0.671	0.913	1.954	0.672	0.701	0.706	0.630
VANILLA + **FDS**	1.442	**0.615**	0.940	2.059	0.681	**0.760**	0.695	0.596
VANILLA + **LDS** + **FDS**	**1.338**	0.670	**0.851**	**1.880**	**0.705**	0.730	**0.764**	**0.655**
OURS (BEST) VS. VANILLA	**+.139**	**-.024**	**+.101**	**+.243**	**+.028**	**-.017**	**+.071**	**+.085**

this NLP task.

Inferring Depth: NYUD2-DIR. For NYUD2-DIR, which is a dense regression task, we verify from Table 3-4 that adding LDS and FDS significantly improves the results. We note that the vanilla model can inevitably overfit to the many-shot regions during training. FDS and LDS help alleviate this effect, and generalize better to all regions, with minor degradation in the many-shot region but significant boosts for other regions.

Inferring Health Score: SHHS-DIR. Table 3-5 reports the results on SHHS-DIR. Since SMOTER and SMOGN are not directly applicable to this medical data, we skip them for

Table 3-5: **Benchmarking results on SHHS-DIR.**

Metrics	MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	15.36	12.47	13.98	16.94	10.63	8.04	9.59	12.20
FOCAL-R	14.67	11.70	13.69	17.06	9.98	7.93	8.85	11.95
FOCAL-R + **LDS**	14.49	12.01	12.43	16.57	9.98	7.89	8.59	11.40
FOCAL-R + **FDS**	14.18	**11.06**	13.56	15.99	9.45	**6.95**	8.81	11.13
FOCAL-R + **LDS** + **FDS**	**14.02**	11.08	**12.24**	**15.49**	**9.32**	7.18	**8.10**	**10.39**
RRT	14.78	12.43	14.01	16.48	10.12	8.05	9.71	11.96
RRT + **LDS**	14.56	12.08	13.44	16.45	9.89	7.85	9.18	11.82
RRT + **FDS**	14.36	11.97	13.33	16.08	9.74	7.54	9.20	11.31
RRT + **LDS** + **FDS**	**14.33**	**11.96**	**12.47**	**15.92**	**9.63**	**7.35**	**8.74**	**11.17**
INV	14.39	11.84	13.12	16.02	9.34	7.73	8.49	11.20
INV + **LDS**	14.14	11.66	12.77	16.05	9.26	7.64	8.18	11.32
INV + **FDS**	13.91	**11.12**	12.29	15.53	8.94	**6.91**	7.79	10.65
INV + **LDS** + **FDS**	**13.76**	**11.12**	**12.18**	**15.07**	**8.70**	6.94	**7.60**	**10.18**
OURS (BEST) VS. VANILLA	**+1.60**	**+1.41**	**+1.80**	**+1.87**	**+1.93**	**+1.13**	**+1.99**	**+2.02**

Table 3-6: **Interpolation & extrapolation results on the curated subset of IMDB-WIKI-DIR.** Using LDS and FDS, the generalization results on zero-shot regions can be consistently improved.

Metrics	MAE ↓				GM ↓			
Shot	All	w/ data	Interp.	Extrap.	All	w/ data	Interp.	Extrap.
VANILLA	11.72	9.32	16.13	18.19	7.44	5.33	14.41	16.74
VANILLA + **LDS**	10.54	8.31	14.14	17.38	6.50	4.67	12.13	15.36
VANILLA + **FDS**	11.40	8.97	15.83	18.01	7.18	5.12	14.02	16.48
VANILLA + **LDS** + **FDS**	**10.27**	**8.11**	**13.71**	**17.02**	**6.33**	**4.55**	**11.71**	**15.13**
OURS (BEST) VS. VANILLA	**+1.45**	**+1.21**	**+2.42**	**+1.17**	**+1.11**	**+0.78**	**+2.70**	**+1.61**

this dataset. The results again confirm the effectiveness of both FDS and LDS when applied for real-world imbalanced regression tasks, where by combining FDS and LDS we often get the highest gains over all tested regions.

■ 3.3.2 Further Analysis

Extrapolation & Interpolation. In real-world DIR tasks, certain target values can have no data at all (e.g., see SHHS-DIR and STS-B-DIR in Fig. 3-6). This motivates the need for target extrapolation and interpolation. We curate a subset from the training set of IMDB-WIKI-DIR, which has no training data in certain regions (Fig. 3-7), but evaluate on the original testset for zero-shot generalization analysis.

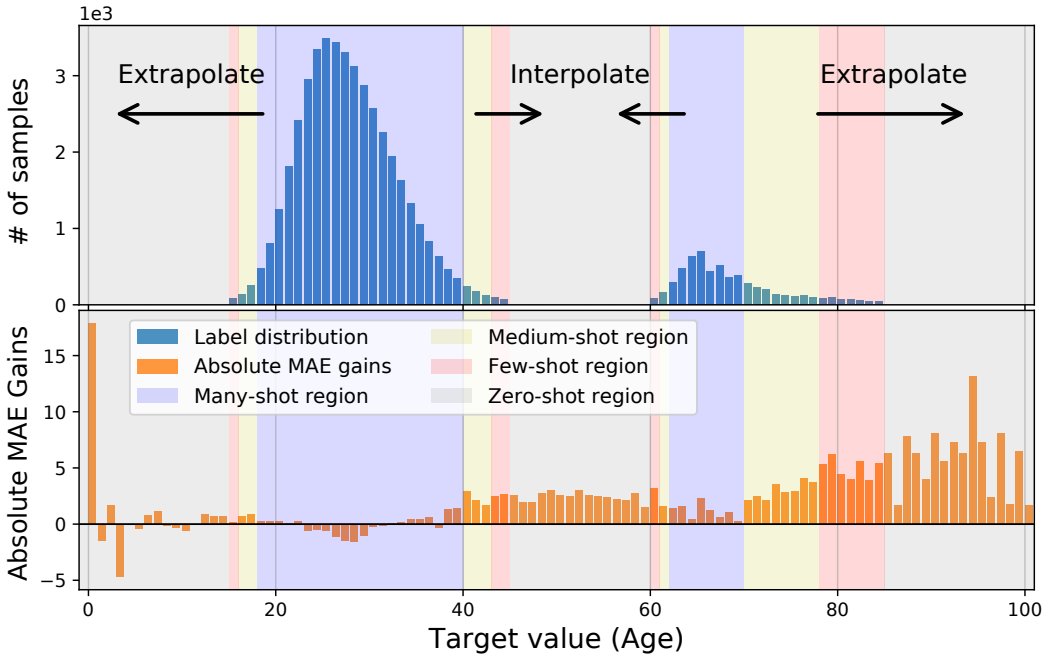
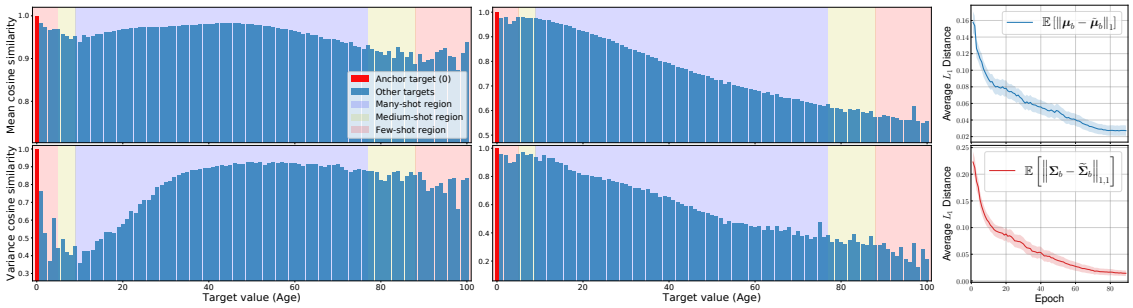


Figure 3-7: **The absolute MAE gains of LDS + FDS over the vanilla model.** We test on a curated subset of IMDB-WIKI-DIR with certain target values having no training data. We establish notable performance gains w.r.t. all regions, especially for extrapolation & interpolation.

As Table 3-6 shows, compared to the vanilla model, LDS and FDS can both improve the results not only on regions that have data, but also achieve larger gains on those without data. Specifically, substantial improvements are established for both target interpolation and extrapolation, where interpolation enjoys larger boosts.

We further visualize the absolute MAE gains of our method over vanilla model in Fig. 3-7. Our method provides a comprehensive treatment to the many, medium, few, as well as zero-shot regions, achieving remarkable performance gains.

Understanding FDS. We investigate how FDS influences the feature statistics. In Fig. 3-8(a) and 3-8(b) we plot the similarity of the feature statistics for anchor age 0, using model trained without and with FDS. As the figure indicates, since age 0 lies in the few-shot region, the feature statistics can have a large bias, i.e., age 0 shares large similarity with region $40 \sim 80$ as in Fig. 3-8(a). In contrast, when FDS is added, the statistics are better calibrated, resulting in a high similarity only in its neighborhood, and a gradually decreasing similarity score as target value becomes larger. We further visualize the L_1 distance between the running statistics $\{\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$ and the smoothed statistics $\{\widetilde{\boldsymbol{\mu}}_b, \widetilde{\boldsymbol{\Sigma}}_b\}$ during training in Fig. 3-8(c). Interestingly, the average L_1 distance becomes smaller and gradually diminishes as



(a) Feature similarity for age 0, w/o FDS (b) Feature similarity for age 0, with FDS (c) Stats change

Figure 3-8: **Analysis on how FDS works. (a) & (b)** Feature statistics similarity for anchor age 0, using model trained without and with FDS. **(c)** L_1 distance between the running statistics $\{\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\}$ and the smoothed statistics $\{\widetilde{\boldsymbol{\mu}}_b, \widetilde{\boldsymbol{\Sigma}}_b\}$ during training.

the training evolves, indicating that the model learns to generate features that are more accurate even without smoothing, and finally the smoothing module can be removed during inference. We provide more results for different anchor ages in Appendix B.4.7, where similar effects can be observed.

Ablation: Kernel type for LDS & FDS (Appendix B.4.1). We study the effects of different kernel types for LDS and FDS when applying distribution smoothing. We select three different kernel types, i.e., Gaussian, Laplacian, and Triangular kernel, and evaluate their influences on both LDS and FDS. In general, all kernel types lead to notable gains (e.g., $3.7\% \sim 6.2\%$ relative MSE gains on STS-B-DIR), with the Gaussian kernel often delivering the best results.

Ablation: Different regression loss functions (Appendix B.4.2). We investigate the influence of different training loss functions on LDS and FDS. We select three common losses used for regression tasks, i.e., L_1 loss, MSE loss, and the Huber loss (also referred to as smoothed L_1 loss). We find that similar results are obtained for all losses, indicating that both LDS and FDS are robust to different loss functions.

Ablation: Hyper-parameter for LDS & FDS (Appendix B.4.3). We investigate the effects of hyper-parameters on both LDS and FDS. As we mainly employ the Gaussian kernel for distribution smoothing, we extensively study different choices of the kernel size l and standard deviation σ. Interestingly, we find LDS and FDS are surprisingly robust to different hyper-parameters in a given range, and obtain similar gains. For example, on STS-B-DIR with $l \in \{5, 9, 15\}$ and $\sigma \in \{1, 2, 3\}$, overall MSE gains range from 3.3% to 6.2%, with $l = 5$

and $\sigma = 2$ exhibiting the best results.

Ablation: Robustness to diverse skewed label densities (Appendix B.4.4). We curate different imbalanced distributions for IMDB-WIKI-DIR by combining different number of disjoint skewed Gaussian distributions over the target space, with potential missing data in certain target regions, and evaluate the robustness of FDS and LDS to the distribution change. We verify that even under different imbalanced label distributions, LDS and FDS consistently boost the performance across all regions compared to the vanilla model, with relative MAE gains ranging from 8.8% to 12.4%.

Comparisons to imbalanced classification methods (Appendix B.4.6). Finally, to gain more insights on the intrinsic difference between imbalanced classification & imbalanced regression problems, we directly apply existing imbalanced classification schemes on several appropriate DIR datasets, and show empirical comparisons with imbalanced regression approaches. We demonstrate in Appendix B.4.6 that LDS and FDS outperform imbalanced classification schemes by a large margin, where the errors for few-shot regions can be reduced by up to 50% to 60%. Interestingly, the results also show that imbalanced classification schemes often perform *worse* than even the vanilla regression model, which confirms that regression requires different approaches for data imbalance than simply applying classification methods. We note that imbalanced classification methods could fail on regression problems for several reasons. First, they ignore the similarity between data samples that are close w.r.t. the continuous target. Moreover, classification cannot extrapolate or interpolate in the continuous label space, therefore unable to deal with missing data in certain target regions.

3.4 Summary

In this chapter, we introduce and study the DIR task that learns from natural imbalanced data with continuous targets, and generalizes to the entire target range. We propose two simple and effective algorithms for DIR that exploit the similarity between nearby targets in both label and feature spaces. Extensive results on five curated large-scale real-world DIR benchmarks confirm the superior performance of our methods. Our work fills the gap in benchmarks and techniques for practical DIR tasks.

CHAPTER 4

On Multi-Domain Long-Tailed Recognition, Generalization and Beyond

Real-world data often exhibit label imbalance – i.e., instead of a uniform label distribution over classes, in reality, data are by their nature imbalanced: a few classes contain a large number of instances, whereas many others have only a few instances [68, 71, 22]. This phenomenon poses a challenge for deep recognition models, and has motivated several prior solutions [71, 100, 69, 72, 22, 2]. Such prior solutions focus on *single domain* scenarios, i.e., samples are from the same data distribution; they propose techniques for learning from imbalanced training data and generalizing to a balanced test set.

In contrast, this chapter formulates the problem of *Multi-Domain Long-Tailed Recognition* (MDLT) as learning from multi-domain imbalanced data, with each domain having its own imbalanced label distribution, and generalizing to a test set that is balanced over all domain-class pairs. MDLT is a natural extension of the single domain case. It arises in real-world scenarios, where data targeted for one task can originate from different domains. For example, in visual recognition problems, minority classes from "photo" images could be complemented with potentially abundant samples from "sketch" images. Similarly, in autonomous driving, the minority accident class in "real" life could be enriched with accidents generated in "simulation". Also, in medical diagnosis, data from distinct

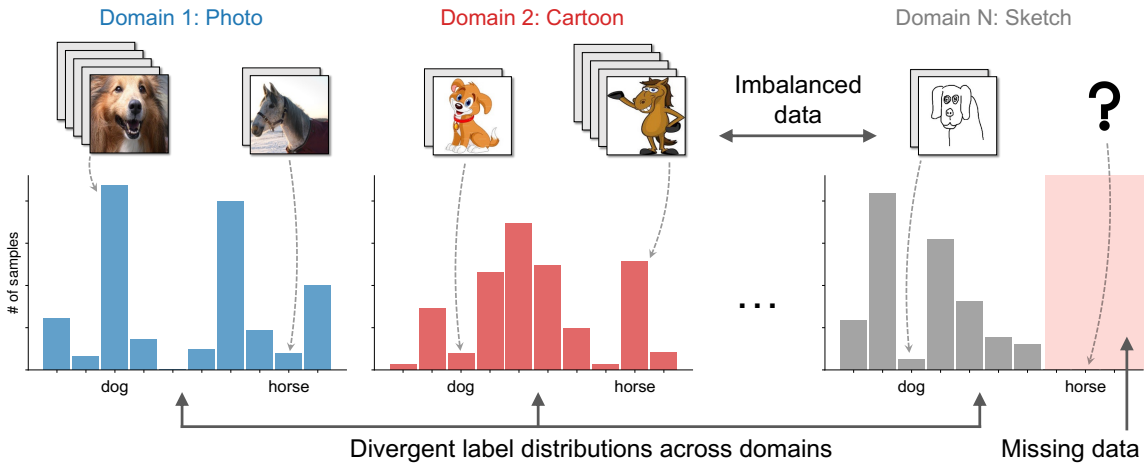


Figure 4-1: **Multi-Domain Long-Tailed Recognition (MDLT).** MDLT aims to learn from imbalanced data from multiple distinct domains, tackle label imbalance, domain shift, and divergent label distributions across domains, and generalize to the entire set of classes over all domains.

populations could enhance each other, where minority samples from one institution could be enriched with instances from others. In the above examples, different data types act as distinct *domains*, and such multi-domain data could be leveraged to tackle the inherent data imbalance within each domain.

We note that MDLT has key differences from its single-domain counterpart:

- First, the label distribution for each domain is likely different from other domains. For example, in Fig. 4-1, both "Photo" and "Cartoon" domains exhibit imbalanced label distributions; Yet, the "horse" class in "Cartoon" has many more samples than in "Photo". This creates challenges with *divergent label distributions across domains*, in addition to in-domain data imbalance.
- Second, multi-domain data inherently involves *domain shift*. Simply treating different domains as a whole and applying traditional data-imbalance methods is unlikely to yield the best results, as the domain gap can be arbitrarily large.
- Third, MDLT naturally motivates *zero-shot generalization within and across domains* – i.e., to generalize to both in-domain missing classes (Fig. 4-1 right part), as well as new domains with no training data, where the latter case is typically denoted as Domain Generalization (DG).

To deal with the above issues, we first develop the *domain-class transferability graph*, which quantifies the transferability between different domain-class pairs under data im-

balance. In this graph, each node refers to a domain-class pair, and each edge refers to the distance between two domain-class pairs in the embedding space. We show that the transferability graph dictates the performance of imbalanced learning across domains. Inspired by this, we design BoDA (Balanced Domain-Class Distribution Alignment), a new loss function that encourages similarity between features of the same class in different domains, and penalizes similarity between features of different classes within and across domains. BoDA does so while accounting for that different classes have very different number of samples, and hence the statistics of their features are intrinsically imbalanced. Analytically, we prove that minimizing the BoDA loss optimizes an upper bound of the *balanced* transferability statistics, corroborating the effectiveness of BoDA for learning multi-domain imbalanced data.

For MDLT evaluation, we curate five MDLT benchmarks based on datasets widely used for domain generalization (DG). These datasets naturally exhibit heavy class imbalance within each domain and data shift across domains, highlighting that the MDLT problem is widely present in current benchmarks. We compare BoDA against twenty algorithms that span different learning strategies. Extensive experiments across benchmarks and algorithms verify that BoDA consistently outperforms all these baselines on all datasets.

Additionally, we examine how BoDA performs in the DG setting. We show that combining BoDA with the DG state-of-the-art (SOTA) consistently brings further gains, yielding a new SOTA for DG. These results shed light on how label imbalance can affect out-of-distribution generalization and highlight the importance of integrating label imbalance into practical DG algorithm design.

In this chapter, we summarize our contributions as follows: (i) We formulate the MDLT problem as learning from multi-domain imbalanced data and generalizing across all domain-class pairs. (ii) We introduce the domain-class transferability graph, a unified model for investigating MDLT. We further show that the transferability statistics induced from such graph are crucial and govern the success of MDLT algorithms. (iii) We design BoDA, a simple, effective, and interpretable loss function for MDLT. We prove theoretically that minimizing the BoDA loss is equivalent to optimizing an upper bound of balanced transferability statistics. (iv) Extensive experiments on benchmark datasets verify the superior and consistent performance of BoDA. Further, combined with DG algorithms, BoDA establishes a new SOTA on DG benchmarks, highlighting the importance of tackling cross-domain data

imbalance for domain generalization.

4.1 Related Work

Long-Tailed Recognition. The literature is rich with research on long-tailed recognition [69, 101]. Proposed solutions include re-balancing the data by either over-sampling the minority classes or under-sampling the majority classes [74, 75], re-weighting or adjusting the loss functions [70, 72, 71, 78], as well as leveraging relevant learning paradigms such as transfer learning [69], metric learning [80], meta-learning [81], two-stage training [82], ensemble learning [102, 103], and self-supervised learning [22, 104]. Recent studies have also explored imbalanced regression [2]. In contrast to these past works, we extend long-tailed recognition to the multi-domain setting, and introduce new techniques suitable for learning from multi-domain imbalanced data.

Multi-Domain Learning. Multi-domain learning (MDL) aims to learn a model of minimal risk from datasets drawn from different underlying distributions [105], and is a specific case of transfer learning [106]. In contrast to domain adaptation (DA) [107, 106], which aims to minimize the risk over a single "target" domain, MDL minimizes the risk over all "source" domains, and considers both average and worst risks over all distributions [108]. Past solutions for MDL include designing shared and domain-specific models [105, 109], leveraging multi-task learning [110], and learning domain-invariant features [108, 111, 112, 113]. Our work falls under the MDL framework, but considers the practical and realistic setting where the label distribution is imbalanced within each domain and across domains.

Domain Generalization. Unlike MDL which focuses on in-domain generalization, domain generalization (DG) aims to learn from multiple training domains and generalize to unseen domains [114]. Previous approaches include learning domain-invariant features [115, 113, 112], learning transferable model parameters using meta-learning [116, 117], data augmentation [118, 119], and capturing causal relationships [66, 120]. Past work on DG has not investigated label imbalance within a domain and across domains. This chapter shows that label imbalance plays a crucial role in DG, and that by combating data imbalance, we substantially boost DG performance on standard benchmarks.

4.2 Domain-Class Transferability Graph

When learning from MDLT, a natural question arises:

*How do we model MDLT in the presence of both **domain shift** and **class imbalance** within and across domains?*

We argue that in contrast to single-domain imbalanced learning where the basic unit one cares about is a *class* (i.e., minority *vs.* majority classes), in MDLT, the basic unit naturally translates to a **domain-class pair**.

Problem Setup. Given a multi-domain classification task with a discrete label space $\mathcal{C} = \{1, \ldots, C\}$ and a domain space $\mathcal{D} = \{1, \ldots, D\}$, let $\mathcal{S} = \{(\mathbf{x}_i, c_i, d_i)\}_{i=1}^{N}$ be the training set, where $\mathbf{x}_i \in \mathbb{R}^l$ denotes the input, $c_i \in \mathcal{C}$ is the class label, and $d_i \in \mathcal{D}$ is the domain label. We denote as $\mathbf{z} = f(\mathbf{x}; \theta)$ the representation of $\mathbf{x}$, where $f : \mathcal{X} \to \mathcal{Z}$ maps the input into a representation space $\mathcal{Z} \subseteq \mathbb{R}^h$. The final prediction $\widehat{c} = g(\mathbf{z})$ is given by a classification function $g : \mathcal{Z} \to \mathcal{C}$. We denote the set of samples belonging to domain d and class c (i.e., the domain-class pair (d, c)) as $\mathcal{S}_{d,c} \subseteq \mathcal{S}$, with $N_{d,c} \triangleq |\mathcal{S}_{d,c}|$ as the number of samples. Similarly, $\mathcal{Z}_{d,c} \subseteq \mathcal{Z}$ denotes the representation set for (d, c). We use $\mathcal{M} = \mathcal{D} \times \mathcal{C} := \{(d, c) : d \in \mathcal{D}, c \in \mathcal{C}\}$ to denote the set of all domain-class pairs.

Definition 1 (Transferability). *Given a learned model and a distance function* $\mathsf{d} : \mathbb{R}^h \times \mathbb{R}^h \to \mathbb{R}$ *in the feature space, the transferability from domain-class pair* (d, c) *to* (d', c') *is:*

$$\text{trans}\big((d, c), (d', c')\big) \triangleq \mathbb{E}_{\mathbf{z} \in \mathcal{Z}_{d,c}}\big[\mathsf{d}\left(\mathbf{z}, \boldsymbol{\mu}_{d',c'}\right)\big],$$

where $\boldsymbol{\mu}_{d',c'} \triangleq \mathbb{E}_{\mathbf{z}' \in \mathcal{Z}_{d',c'}}[\mathbf{z}']$ *is the first order statistics (i.e., mean) of* (d', c').

Intuitively, the transferability between two domain-class pairs is the average distance between their learned representations, characterizing how close they are in the feature space. By default, d is chosen as the Euclidean distance, but it can also represent the higher order statistics of (d, c). For example, the Mahalanobis distance [121] uses the covariance $\boldsymbol{\Sigma}_{d,c} \triangleq \mathbb{E}_{\mathbf{z} \in \mathcal{Z}_{d,c}}\left[(\mathbf{z} - \boldsymbol{\mu}_{d,c})(\mathbf{z} - \boldsymbol{\mu}_{d,c})^\top\right]$. In the remainder of the chapter, with a slight abuse of the notation, we allow $\boldsymbol{\mu}_{d,c}$ to represent both the first and higher order statistics for (d, c).

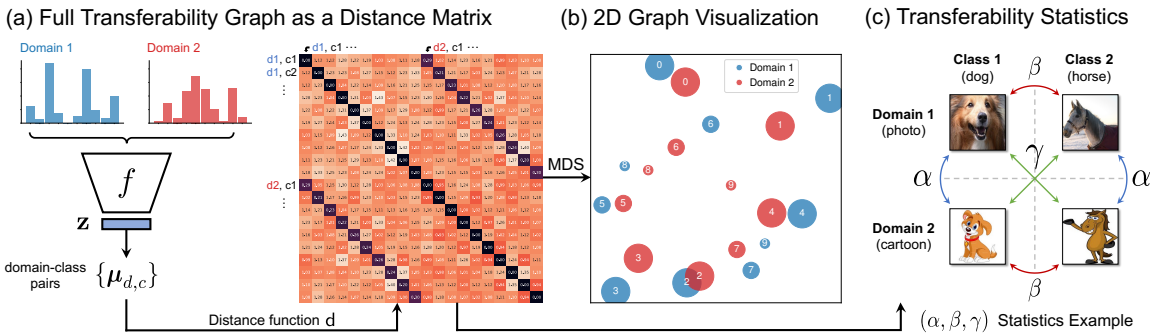


Figure 4-2: **Overall framework of transferability graph. (a)** Distribution statistics $\{\boldsymbol{\mu}_{d,c}\}$ is computed for all domain-class pairs, by which we generate a full transferability matrix. **(b)** MDS is used to project the graph into a 2D space for visualization. **(c)** We define (α, β, γ) transferability statistics to further describe the whole transferability graph.

Definition 2 (Transferability Graph). *The transferability graph for a learned model is defined as* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, where the vertices,* $\mathcal{V} \subseteq \{\boldsymbol{\mu}_{d,c}\}$*, represents the domain-class pairs, and the edges,* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$*, are assigned weights equal to* $\text{trans}\left((d, c), (d', c')\right)$.

Transferability Graph Visualization. It is convenient to directly visualize the transferability graph of a learned model in a 2D Cartesian space. To do so, we use the average of $\text{trans}\left((d, c), (d', c')\right)$ and $\text{trans}\left((d', c'), (d, c)\right)$ as a similarity measure between them. We can then visualize this similarity and the underlying transferability graph using multidimensional scaling (MDS) [122]. Figs. 4-2a and 4-2b show this process, where for each (d, c) pair, we estimate its distribution statistics $\{\boldsymbol{\mu}_{d,c}\}$ from the learned model and compute the transferability graph as a distance matrix. We then use MDS to project it into a 2D space, where each dot refers to one (d, c), and the distance represents transferability.

Definition 3 ((α, β, γ) Transferability Statistics). *The transferability graph can be summarized by the following transferability statistics:*

$$\textit{Different domains, same class:} \quad \alpha = \mathbb{E}_c \mathbb{E}_d \mathbb{E}_{d' \neq d} \left[\text{trans}\big((d, c), (d', c)\big)\right].$$

$$\textit{Same domain, different classes:} \quad \beta = \mathbb{E}_d \mathbb{E}_c \mathbb{E}_{c' \neq c} \left[\text{trans}\big((d, c), (d, c')\big)\right].$$

$$\textit{Different domains, different classes:} \quad \gamma = \mathbb{E}_d \mathbb{E}_{d' \neq d} \mathbb{E}_c \mathbb{E}_{c' \neq c} \left[\text{trans}\big((d, c), (d', c')\big)\right].$$

As illustrated in Fig. 4-2c, (α, β, γ) captures the similarity between features of the same class across domains and different classes within and across domains.

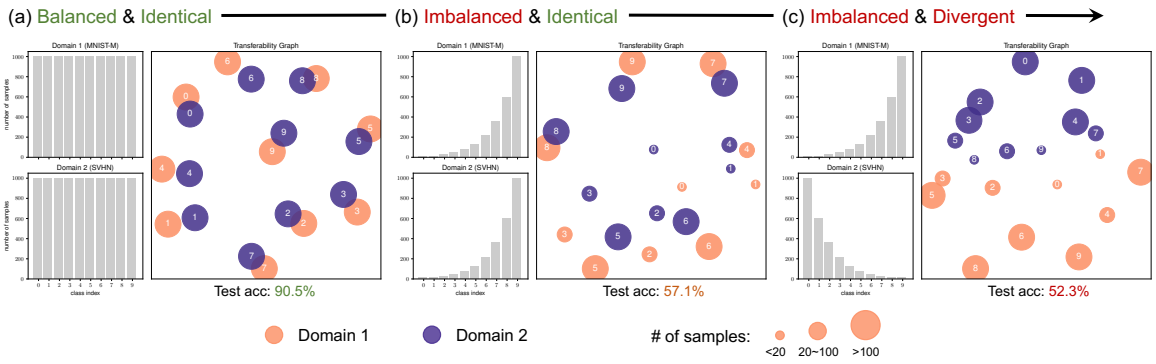


Figure 4-3: **The evolving pattern of transferability graph when varying label proportions of `Digits-MLT`. (a)** Label distributions for two domains are balanced and identical. **(b)** Label distributions for two domains are imbalanced but identical. **(c)** Label distributions for two domains are imbalanced and *divergent*.

4.3 What Makes for Good Representations in MDLT

4.3.1 Divergent Label Distributions Hamper Transferable Features

MDLT has to deal with differences between the label distributions across domains. To understand the implications of this issue we start with an example.

Motivating Example. We construct `Digits-MLT`, a two-domain toy MDLT dataset that combines two digit datasets: MNIST-M [113] and SVHN [123]. The task is 10-class digit classification. Details of the datasets are in Appendix C.3. We manually vary the number of samples for each domain-class pair to simulate different label distributions, and train a plain ResNet-18 [64] using empirical risk minimization (ERM) for each case. We keep all test sets balanced and identical.

The results in Fig. 4-3 reveal interesting observations. When the per-domain label distributions are balanced and *identical* across domains, although a domain gap exists, it does not prohibit the model from learning discriminative features of high accuracy (90.5%), as shown in Fig. 4-3a. If the label distributions are imbalanced but *identical*, as in Fig. 4-3b, ERM is still able to align similar classes in the two domains, where majority classes (e.g., class 9) are closer in terms of transferability than minority classes (e.g., class 0). In contrast, when the labels are both imbalanced and *mismatched* across domains, as in Fig. 4-3c, the learned features are no longer transferable, resulting in a clear gap across domains and the worst accuracy. This is because *divergent label distributions* across domains produce an undesirable shortcut; the model can minimize the classification loss simply by separating

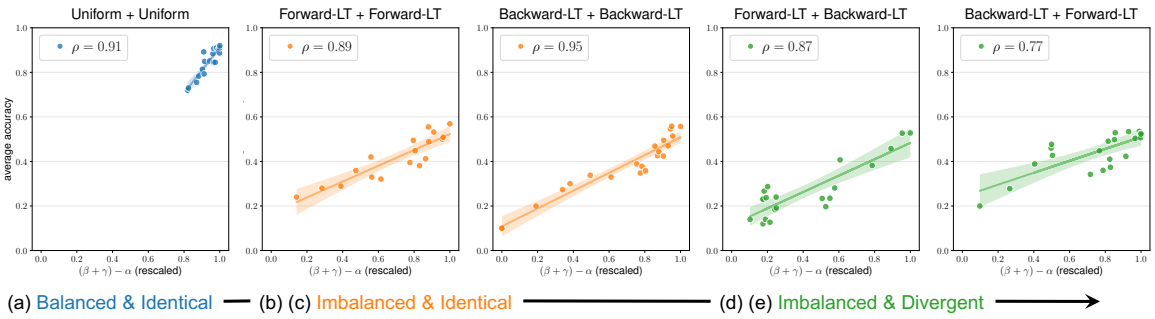


Figure 4-4: **Correspondence between $(\beta + \gamma) - \alpha$ quantity and test accuracy across different label configurations of `Digits-MLT`.** Each plot refers to specific label distributions for two domains (e.g., (a) employs "Uniform" for domain 1 and "Uniform" for domain 2). Each point corresponds to a model trained with ERM using different hyperparameters.

the two domains.

Transferable Features are Desirable. As the results indicate, *transferable* features across (d, c) pairs are needed, especially when imbalance occurs. In particular, the transferability link between the same class across domains should be greater than that between different classes within or across domains. This can be captured via the (α, β, γ) transferability statistics, as we show next.

■ 4.3.2 Transferability Statistics Characterize Generalization

Motivating Example. Again, we use `Digits-MLT` with varying label distributions. We consider three imbalance types to compose different label configurations: (1) **Uniform** (i.e., balanced labels), (2) **Forward-LT**, where the labels exhibit a long tail over class ids, and (3) **Backward-LT**, where labels are inversely long-tailed with respect to the class ids. For each configuration, we train 20 ERM models with varying hyperparameters. We then calculate the (α, β, γ) statistics for each model, and plot its classification accuracy against $(\beta + \gamma) - \alpha$.

Fig. 4-4 reveals the following findings: (1) *The (α, β, γ) statistics characterize a model's performance in MDLT.* In particular, the $(\beta + \gamma) - \alpha$ quantity displays a very strong correlation with test performance across the entire range and every label configuration. (2) *Data imbalance increases the risk of learning less transferable features.* When the label distributions are similar across domains (Fig. 4-4a), the models are robust to varying parameters, clustering in the upper-right region. However, as the labels become imbalanced (Figs. 4-4b, 4-4c)

and further divergent (Figs. 4-4d, 4-4e), chances that the model learns non-transferable features (i.e., lower $(\beta+\gamma)-\alpha$) increase, leading to a large drop in performance. We provide further evidence in Appendix C.7.4 showing that these observations hold regardless of datasets and training regimes.

■ 4.3.3 A Loss that Bounds the Transferability Statistics

We use the above findings to design a new loss function particularly suitable for MDLT. We will first introduce the loss function then prove that it minimizes an upper bound of the (α,β,γ) statistics. We start from a simple loss inspired by the metric learning objective [124, 125]. We call this loss $\mathcal{L}_{\texttt{DA}}$ since it aims for Domain-Class Distribution Alignment, i.e., aligning the features of the same class across domains. Let $(\mathbf{x}_i, c_i, d_i)$ denote a sample with feature $\mathbf{z}_i$. Given a set of training samples with feature set $\mathcal{Z}$, we have

$$\mathcal{L}_{\texttt{DA}}(\mathcal{Z},\{\boldsymbol{\mu}\}) = \sum_{\mathbf{z}_i\in\mathcal{Z}} \frac{-1}{|\mathcal{D}|-1} \sum_{d\in\mathcal{D}\setminus\{d_i\}} \log \frac{\exp\left(-\mathsf{d}(\mathbf{z}_i,\boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}}\exp\left(-\mathsf{d}(\mathbf{z}_i,\boldsymbol{\mu}_{d',c'})\right)}. \tag{4.1}$$

Intuitively, $\mathcal{L}_{\texttt{DA}}$ tackles label *divergence*, as (d,c) pairs that share same class would be pulled closer, and vice versa. It is also related to (α,β,γ) statistics, as the numerator represents *positive* cross-domain pairs (α), and the denominator represents *negative* cross-class pairs (β,γ). A detailed probabilistic interpretation of $\mathcal{L}_{\texttt{DA}}$ is provided in Appendix C.2.2.

But, $\mathcal{L}_{\texttt{DA}}$ does not address label *imbalance*. Note that (α,β,γ) is defined in a *balanced* way, independent of the number of samples of each (d,c). However, given an imbalanced dataset, most samples will come from majority domain-class pairs, which would dominate $\mathcal{L}_{\texttt{DA}}$ and cause minority pairs to be overlooked.

Balanced Domain-Class Distribution Alignment (BoDA). To tackle data imbalance across (d,c) pairs, we modify the loss in Eqn. (4.1) to the BoDA loss:

$$\mathcal{L}_{\texttt{BoDA}}(\mathcal{Z},\{\boldsymbol{\mu}\}) = \sum_{\mathbf{z}_i\in\mathcal{Z}} \frac{-1}{|\mathcal{D}|-1} \sum_{d\in\mathcal{D}\setminus\{d_i\}} \log \frac{\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i,\boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}}\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i,\boldsymbol{\mu}_{d',c'})\right)},\quad \widetilde{\mathsf{d}}(\mathbf{z}_i,\boldsymbol{\mu}_{d,c}) = \frac{\mathsf{d}(\mathbf{z}_i,\boldsymbol{\mu}_{d,c})}{N_{d_i,c_i}}. \tag{4.2}$$

BoDA scales the original d by a factor of $1/N_{d_i,c_i}$, i.e., it counters the effect of imbalanced domain-class pairs by introducing a *balanced* distance measure $\widetilde{\mathsf{d}}$.

Theorem 4 ($\mathcal{L}_{\texttt{BoDA}}$ as an Upper Bound). *Given a multi-domain long-tailed dataset $\mathcal{S}$ with domain label space $\mathcal{D}$ and class label space $\mathcal{C}$ satisfying $|\mathcal{D}| > 1$ and $|\mathcal{C}| > 1$, let $\mathcal{Z}$ be the representation set of all training samples, and (α, β, γ) be the transferability statistics for $\mathcal{S}$ defined in Definition 3. It holds that*

$$\mathcal{L}_{\texttt{BoDA}}(\mathcal{Z}, \{\boldsymbol{\mu}\}) \geq N \log \left(|\mathcal{D}| - 1 + |\mathcal{D}|(|\mathcal{C}| - 1) \exp \left(\frac{|\mathcal{C}||\mathcal{D}|}{N} \cdot \alpha - \frac{|\mathcal{C}|}{N} \cdot \beta - \frac{|\mathcal{C}|(|\mathcal{D}| - 1)}{N} \cdot \gamma \right) \right). \tag{4.3}$$

The proof of Theorem 4 is in Appendix C.1.2. Theorem 4 has the following interesting implications: (1) $\mathcal{L}_{\texttt{BoDA}}$ *upper-bounds (α, β, γ) statistics in a desired form that naturally translates to better performance.* By minimizing $\mathcal{L}_{\texttt{BoDA}}$, we ensure a low α (attract same classes) and high β, γ (separate different classes), which are essential conditions for generalization in MDLT. (2) *The constant factors correspond to how much each component contributes to the transferability graph.* Zooming on the arguments of $\exp(\cdot)$, we observe that the objective is proportional to $\alpha - (\frac{1}{|\mathcal{D}|}\beta + \frac{|\mathcal{D}|-1}{|\mathcal{D}|}\gamma)$. According to Definition 3, we note that α summarizes data similarity for the same class, while $(\frac{1}{|\mathcal{D}|}\beta + \frac{|\mathcal{D}|-1}{|\mathcal{D}|}\gamma)$ summarizes data similarity across different classes, using the weighted average of β and γ, where their weights are proportional to the number of associated domains (i.e., 1 for β, $(|\mathcal{D}| - 1)$ for γ).

■ 4.3.4 Calibration for Data Imbalance Leads to Better Transfer

`BoDA` works by encouraging feature transfer for similar classes across domains, i.e., if (d, c) and (d', c) refer to the same class in different domains, then we want to transfer their features to each other. But, minority domain-class pairs naturally have worse $\boldsymbol{\mu}_{d,c}$ estimates due to data scarcity, and forcing other pairs to transfer to them hurts learning. Thus, when bringing two domain-class pairs closer in the embedding space, we want the minority (d, c) to transfer to majority ones, not the inverse. The following example further clarifies this point.

Motivating Example. We use `Digits-MLT` with divergent labels (Fig. 4-5). We focus on *feature discrepancy*, i.e., the distance between training and test features for the same class. For each class in domain 1, we compute the distance in the feature space between the means of the training set and test set (solid line). We also compute the distance between the training data of domain 2 and test data of domain 1 (dashed line), for the same class.

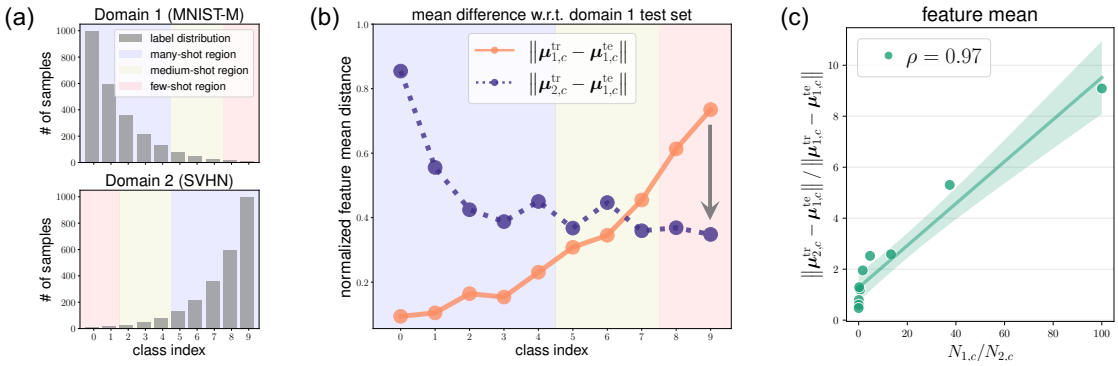


Figure 4-5: **The need for distance calibration.** **(a)** Per-domain label distribution of `Digits-MLT`. **(b)** Distance between training and test data. Solid line plots the distance between training and test data from the same domain-class pairs. Dashed line plots the distance between test data from a particular domain-class pair and the training data with which it shares the same class but differs in the domain. The blue and red background colors refer to majority and minority domain-class pairs, respectively. **(c)** Correspondence between the ratio of the sample size and their feature distances between testing and training across different domain-class pairs.

As shown by the solid orange line in Fig. 4-5b, for minority domain-class pairs such as class "8" and "9" in domain 1, the distance in the feature space between training and testing is large. In fact, the test set of these minority domain-class pairs is closer to the training data for "8" and "9" in domain 2 than in their own domain, as shown by the dashed purple line. This example indicates that a better training would try to transfer the features of minority domain-class pairs to majority pairs with which they share the same class, as shown by the grey arrow in Fig. 4-5b. Such transfer will improve generalization to the test set.

`BoDA` **with Calibrated Distance.** The above discussion motivates a modification to `BoDA` to favor transfer to majority domain-class pairs:

$$\widetilde{\mathcal{L}}_{\texttt{BoDA}}(\mathcal{Z}, \{\boldsymbol{\mu}\}) = \sum_{\mathbf{z}_i \in \mathcal{Z}} \frac{-1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D} \setminus \{d_i\}} \log \frac{\exp\left(-\lambda_{d_i,c_i}^{d,c_i} \widetilde{\mathtt{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c') \in \mathcal{M} \setminus \{(d_i,c_i)\}} \exp\left(-\lambda_{d_i,c_i}^{d',c'} \widetilde{\mathtt{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)}, \quad \lambda_{d,c}^{d',c'} = \left(\frac{N_{d',c'}}{N_{d,c}}\right)^{\nu}, \tag{4.4}$$

where ν is a constant that allows for a sublinear relation (default $\nu = 1$). $\lambda_{d,c}^{d',c'}$ indicates how much we would like to transfer (d, c) to (d', c'), based on their relative sample size. Fig. 4-5c verifies that the ratio of the sample size is highly correlated with the ratio of the distance between testing and training. Further, Theorem 6 in Appendix C.1 shows that $\widetilde{\mathcal{L}}_{\texttt{BoDA}}$ is an upper bound of the calibrated transferability statistics.

Variants of `BoDA`**: Matching Higher Order Statistics.** The distance $\mathtt{d}$ can be set to the Eu-

Table 4-1: **The benefits of decoupling the classifier.**

Algorithm	w/o decouple	w/ decouple
ERM [127]	77.6 ± 0.2	**79.2** ± 0.3
DANN [113]	77.7 ± 0.6	**79.0** ± 0.1
CORAL [111]	78.0 ± 0.1	**79.6** ± 0.2

clidean distance $\mathtt{d}(\mathbf{z}, \boldsymbol{\mu}_{d,c}) = \sqrt{(\mathbf{z}-\boldsymbol{\mu}_{d,c})^\top(\mathbf{z}-\boldsymbol{\mu}_{d,c})}$, which captures the first order statistics. To match higher order statistics in the features such as covariance, $\mathtt{d}(\mathbf{z}, \{\boldsymbol{\mu}_{d,c}, \boldsymbol{\Sigma}_{d,c}\}) = \sqrt{(\mathbf{z}-\boldsymbol{\mu}_{d,c})^\top \boldsymbol{\Sigma}_{d,c}^{-1}(\mathbf{z}-\boldsymbol{\mu}_{d,c})}$ is used, resembling the Mahalanobis distance [121]. We refer to these variants as $\widetilde{\mathcal{L}}_{\texttt{BoDA}}$ and $\widetilde{\mathcal{L}}_{\texttt{BoDA-M}}$.

Joint Loss. BoDA serves as a representation learning scheme for MDLT, which operates over $\mathcal{Z}$. For classification, we train deep networks by combining $\widetilde{\mathcal{L}}_{\texttt{BoDA}}$ and the standard cross-entropy (CE) loss in an end-to-end fashion, where CE is applied to the output layer, and BoDA is applied to the latent features. We combine the losses as $\mathcal{L}_{\texttt{CE}} + \omega\widetilde{\mathcal{L}}_{\texttt{BoDA}}$, with ω as a trade-off hyperparameter.

■ 4.4 What Makes for Good Classifiers in MDLT

In the long-tailed recognition literature, an important finding is that decoupling *representation learning* and *classifier learning* leads to better results [82, 126]. In particular, instance-balanced sampling is used during the first stage of learning, while class-balanced sampling is used for re-training the classifier (with the representation fixed) in the second stage [82]. Motivated by this, we explore whether a similar decoupling benefits MDLT. We use three learning algorithms, ERM [127], DANN [112], and CORAL [111]. We train each algorithm with and without the second stage classifier learning, and report the average accuracy over all MDLT datasets (presented later).

As Table 4-1 shows, similar to what has been observed in the single domain case [82, 126], regardless of algorithm, decoupling the classifier learning consistently improves performance. Since BoDA can support both coupled and decoupled classifier learning, we use $\texttt{BoDA}_r$ to refer to models that couple representation and classifier learning, and $\texttt{BoDA}_{r,c}$ for models that decouple representation from classifier learning. In the classifier learning stage, we simply use class-balanced sampling.

4.5 Benchmarking MDLT

Datasets. We curate five multi-domain datasets typically used in DG and adapt them for MDLT evaluation. To do so, for each dataset, we create two balanced datasets one for validation and the other for testing, and leave the rest for training. The size of the validation and test data sets is roughly 5% and 10% of original data, respectively. Table C-1 in Appendix C.3 provides the statistics of each MDLT dataset. Fig. 4-6 shows the label distributions across domains in the five datasets.

- `VLCS-MLT`. We construct `VLCS-MLT` using the `VLCS` dataset [128], which is an object recognition dataset with 10,729 images from 4 domains and 5 classes.
- `PACS-MLT`. `PACS-MLT` is constructed from the `PACS` dataset [129], an object recognition dataset with 9,991 images from 4 domains and 7 classes.
- `OfficeHome-MLT`. We set up `OfficeHome-MLT` using the `OfficeHome` dataset [130] which contains 15,588 images from 4 domains and 65 classes.
- `TerraInc-MLT`. `TerraInc-MLT` is created from `TerraIncognita` [131], a species classification dataset including 24,788 images from 4 domains and 10 classes.
- `DomainNet-MLT`. We construct `DomainNet-MLT` using `DomainNet` [132], a large-scale multi-domain dataset for object recognition. It contains 586,575 images from 345 classes and 6 domains.

Network Architectures. For experiments on the synthetic `Digits-MLT` dataset, we use a simple CNN architecture as in [133]. For the MDLT datasets, we follow [133], and use ResNet-50 [64] for all algorithms.

Competing Algorithms. We compare `BoDA` to a large number of algorithms that span different learning strategies and categories, including (1) *vanilla:* **ERM** [127], (2) *distributionally robust optimization:* **GroupDRO** [134], (3) *data augmentation:* **Mixup** [135], **SagNet** [136], (4) *meta-learning:* **MLDG** [116], (5) *domain-invariant feature learning:* **IRM** [66], **DANN** [113], **CDANN** [112], **CORAL** [111], **MMD** [137], (6) *transfer learning:* **MTL** [138], (7) *multi-task learning:* **Fish** [139], and (8) *imbalanced learning:* **Focal** [99], **CBLoss** [72], **LDAM** [71], **BSoftmax** [100], **SSP** [22], **CRT** [82]. We provide detailed descriptions in Appendix C.4.2.

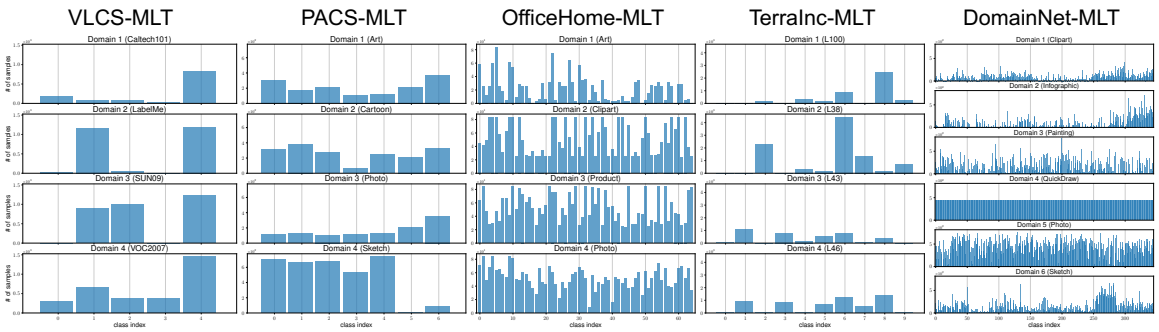


Figure 4-6: **Overview of training set label distribution for five MDLT datasets.** We set up MDLT benchmarks from datasets traditionally used for DG, and make validation/test sets balanced across all domain-class pairs. More details are provided in Appendix C.3.

Implementation and Evaluation Metrics. For a fair evaluation, following [133], for each algorithm we conduct a random search of 20 trials over a joint distribution of all hyperparameters (see Appendix C.4.3 for details). We then use the validation set to select the best hyperparameters for each algorithm, fix them and rerun the experiments under three different random seeds to report the final average accuracy with standard deviation. Such process ensures the comparison is best-versus-best, and the hyperparameters are optimized for all algorithms. In addition to the average accuracy across domains, we also report the worst accuracy over domains, and further divide all domain-class pairs into *many-shot* (pairs with over 100 training samples), *medium-shot* (pairs with 20∼100 training samples), *few-shot* (pairs with under 20 training samples), and *zero-shot* (pairs with no training data), and report the results for these subsets.

■ 4.5.1 Main Results

We report the main results in this section for all MDLT datasets. The complete results and all additional experiments are provided in Appendix C.5 and C.7.

Benchmark Results on MDLT Datasets. The performance of all methods on `VLCS-MLT`, `PACS-MLT`, `OfficeHome-MLT`, `TerraInc-MLT` and `DomainNet-MLT` are in Table 4-2, 4-3, 4-4, 4-5 and 4-6, respectively. We highlight rows in gray for `BoDA` and its variants, and bolden the best result in each column. First, as all tables indicate, `BoDA` consistently achieves the best average accuracy across all datasets. It also achieves the best worst-case accuracy most of the time. Moreover, on certain datasets (e.g., `OfficeHome-MLT`), MDL methods perform better (e.g., CORAL), while on others (e.g., `TerraInc-MLT`), imbalanced methods

Table 4-2: **Results on** `VLCS-MLT`.

Algorithm	Accuracy (by domain)		Accuracy (by shot)			
	Average	Worst	Many	Medium	Few	Zero
ERM [127]	76.3 ±0.4	53.6 ±1.1	84.6 ±0.5	76.6 ±0.4	–	32.9 ±0.4
IRM [66]	76.5 ±0.2	52.3 ±0.7	85.3 ±0.6	75.5 ±1.0	–	33.5 ±1.0
GroupDRO [134]	76.7 ±0.4	54.1 ±1.3	85.3 ±0.9	76.2 ±1.0	–	34.5 ±2.0
Mixup [135]	75.9 ±0.1	52.7 ±1.3	84.4 ±0.2	77.1 ±0.6	–	29.2 ±1.4
MLDG [116]	76.9 ±0.2	53.6 ±0.5	84.9 ±0.3	77.5 ±1.0	–	34.4 ±0.9
CORAL [111]	75.9 ±0.5	51.6 ±0.7	84.3 ±0.6	75.5 ±0.5	–	34.5 ±0.8
MMD [137]	76.3 ±0.6	53.4 ±0.3	84.5 ±0.8	77.1 ±0.5	–	32.7 ±0.3
DANN [113]	77.5 ±0.1	54.1 ±0.3	85.9 ±0.5	76.0 ±0.4	–	38.0 ±2.3
CDANN [112]	76.6 ±0.4	53.6 ±0.4	84.4 ±0.7	77.3 ±0.8	–	35.0 ±0.8
MTL [138]	76.3 ±0.3	52.9 ±0.5	84.8 ±0.9	76.2 ±0.6	–	33.3 ±1.4
SagNet [136]	76.3 ±0.2	52.3 ±0.2	85.3 ±0.3	75.1 ±0.2	–	32.9 ±0.3
Fish [139]	77.5 ±0.3	54.3 ±0.4	86.2 ±0.5	76.0 ±0.4	–	35.6 ±2.2
Focal [99]	75.6 ±0.4	52.3 ±0.2	84.0 ±0.2	75.5 ±0.6	–	32.7 ±0.9
CBLoss [72]	76.8 ±0.3	52.5 ±0.5	84.8 ±0.7	77.5 ±1.4	–	33.2 ±1.6
LDAM [71]	77.5 ±0.1	52.9 ±0.2	**86.5** ±0.4	75.5 ±0.5	–	35.2 ±0.6
BSoftmax [100]	76.7 ±0.5	52.9 ±0.9	84.4 ±0.9	78.2 ±0.6	–	34.3 ±0.9
SSP [22]	76.1 ±0.3	52.3 ±1.0	83.8 ±0.3	76.0 ±1.2	–	37.1 ±0.7
CRT [82]	76.3 ±0.2	51.4 ±0.3	84.5 ±0.1	77.3 ±0.0	–	31.7 ±1.0
BoDA$_r$	76.9 ±0.5	51.4 ±0.3	85.3 ±0.3	77.3 ±0.2	–	33.3 ±0.5
BoDA-M$_r$	77.5 ±0.3	53.4 ±0.3	85.8 ±0.2	77.3 ±0.2	–	35.7 ±0.7
BoDA$_{r,c}$	77.3 ±0.2	53.4 ±0.3	85.3 ±0.3	78.0 ±0.2	–	38.6 ±0.7
BoDA-M$_{r,c}$	**78.2** ±0.4	**55.4** ±0.5	85.3 ±0.3	**79.3** ±0.6	–	**43.3** ±1.1
BoDA *vs.* ERM	+1.9	+1.8	+0.7	+2.7	–	+10.4

Table 4-3: **Results on** `PACS-MLT`.

Algorithm	Accuracy (by domain)		Accuracy (by shot)			
	Average	Worst	Many	Medium	Few	Zero
ERM [127]	97.1 ±0.1	95.8 ±0.2	97.1 ±0.0	97.0 ±0.0	98.0 ±0.9	–
IRM [66]	96.7 ±0.2	95.2 ±0.4	96.8 ±0.2	96.7 ±0.7	94.7 ±1.4	–
GroupDRO [134]	97.0 ±0.1	95.3 ±0.4	97.3 ±0.1	95.3 ±1.2	94.7 ±3.6	–
Mixup [135]	96.7 ±0.2	95.1 ±0.2	97.0 ±0.1	96.7 ±0.3	91.3 ±2.7	–
MLDG [116]	96.6 ±0.1	94.1 ±0.3	96.8 ±0.1	96.3 ±0.7	92.7 ±0.5	–
CORAL [111]	96.6 ±0.5	94.3 ±0.7	96.6 ±0.5	97.0 ±0.8	94.7 ±0.5	–
MMD [137]	96.9 ±0.1	96.2 ±0.2	96.9 ±0.2	97.0 ±0.0	96.7 ±0.5	–
DANN [113]	96.5 ±0.0	94.3 ±0.1	96.5 ±0.1	98.0 ±0.0	94.7 ±2.4	–
CDANN [112]	96.1 ±0.1	94.5 ±0.2	96.1 ±0.1	96.3 ±0.5	94.0 ±0.9	–
MTL [138]	96.7 ±0.2	94.5 ±0.6	96.8 ±0.1	95.3 ±1.7	97.3 ±1.1	–
SagNet [136]	**97.2** ±0.1	95.2 ±0.3	**97.4** ±0.1	96.7 ±0.5	95.3 ±0.5	–
Fish [139]	96.9 ±0.2	95.2 ±0.2	97.0 ±0.1	97.0 ±0.5	94.7 ±1.1	–
Focal [99]	96.5 ±0.2	94.6 ±0.7	96.6 ±0.1	95.0 ±1.7	96.7 ±0.5	–
CBLoss [72]	96.9 ±0.1	95.1 ±0.4	96.8 ±0.2	97.0 ±1.2	**100.0** ±0.0	–
LDAM [71]	96.5 ±0.2	94.7 ±0.2	96.6 ±0.1	95.7 ±1.4	96.0 ±0.0	–
BSoftmax [100]	96.9 ±0.3	95.6 ±0.3	96.6 ±0.4	**98.7** ±0.7	99.3 ±0.5	–
SSP [22]	96.9 ±0.2	95.4 ±0.4	96.7 ±0.2	98.3 ±0.5	98.0 ±0.9	–
CRT [82]	96.3 ±0.1	94.9 ±0.1	96.3 ±0.1	97.3 ±0.3	94.0 ±0.9	–
BoDA$_r$	97.0 ±0.1	95.1 ±0.4	97.0 ±0.1	96.3 ±0.5	98.0 ±0.9	–
BoDA-M$_r$	97.1 ±0.1	94.9 ±0.1	97.3 ±0.1	96.3 ±0.5	96.0 ±0.0	–
BoDA$_{r,c}$	**97.2** ±0.1	95.7 ±0.3	**97.4** ±0.1	97.0 ±0.0	94.7 ±1.1	–
BoDA-M$_{r,c}$	97.1 ±0.2	**96.3** ±0.1	97.1 ±0.0	97.0 ±0.8	96.0 ±0.0	–
BoDA *vs.* ERM	+0.1	+0.5	+0.3	+0.0	-2.0	–

Table 4-4: **Results on** `OH-MLT`.

Algorithm	Accuracy (by domain)		Accuracy (by shot)			
	Average	Worst	Many	Medium	Few	Zero
ERM [127]	80.7 ±0.0	71.3 ±0.1	87.8 ±0.2	81.0 ±0.2	63.1 ±0.1	63.3 ±7.2
IRM [66]	80.6 ±0.4	70.7 ±0.2	87.6 ±0.4	81.5 ±0.4	61.1 ±0.9	56.7 ±1.4
GroupDRO [134]	80.1 ±0.3	68.7 ±0.9	88.1 ±0.2	80.8 ±0.4	59.8 ±1.2	51.7 ±3.6
Mixup [135]	81.2 ±0.2	72.3 ±0.6	87.9 ±0.4	81.8 ±0.1	64.1 ±0.4	60.0 ±4.1
MLDG [116]	80.4 ±0.2	70.2 ±0.6	87.1 ±0.1	81.3 ±0.3	61.3 ±1.0	61.7 ±1.4
CORAL [111]	81.9 ±0.1	**72.7** ±0.6	87.9 ±0.1	83.0 ±0.1	63.5 ±0.7	65.0 ±2.4
MMD [137]	78.4 ±0.4	67.7 ±0.8	85.2 ±0.2	79.4 ±0.7	58.8 ±0.4	56.7 ±3.6
DANN [113]	79.2 ±0.2	70.2 ±0.9	86.2 ±0.1	80.0 ±0.1	60.3 ±1.1	61.7 ±5.9
CDANN [112]	79.0 ±0.2	69.4 ±0.3	86.4 ±0.6	79.8 ±0.1	58.9 ±0.8	50.0 ±4.7
MTL [138]	79.5 ±0.2	69.8 ±0.6	87.3 ±0.3	79.8 ±0.2	61.1 ±0.2	51.7 ±2.7
SagNet [136]	80.9 ±0.1	70.5 ±0.5	87.8 ±0.4	81.9 ±0.1	61.2 ±0.9	56.7 ±3.6
Fish [139]	81.3 ±0.3	71.3 ±0.7	**88.2** ±0.2	81.9 ±0.3	63.2 ±0.8	61.7 ±1.4
Focal [99]	77.9 ±0.0	67.6 ±0.4	86.5 ±0.3	78.3 ±0.1	57.4 ±0.3	46.7 ±3.6
CBLoss [72]	79.8 ±0.2	69.5 ±0.7	86.6 ±0.4	80.6 ±0.2	61.1 ±1.4	65.0 ±2.4
LDAM [71]	80.3 ±0.2	69.9 ±0.5	87.1 ±0.2	81.3 ±0.3	61.1 ±0.2	51.7 ±2.7
BSoftmax [100]	80.4 ±0.2	70.9 ±0.5	86.7 ±0.5	81.3 ±0.3	62.4 ±1.0	60.0 ±4.1
SSP [22]	81.1 ±0.3	71.1 ±0.3	87.3 ±0.6	82.3 ±0.3	61.6 ±0.7	63.3 ±1.4
CRT [82]	81.2 ±0.0	72.5 ±0.2	87.7 ±0.1	81.8 ±0.1	64.0 ±0.1	65.0 ±2.4
BoDA$_r$	81.5 ±0.1	71.8 ±0.1	87.7 ±0.2	82.3 ±0.1	**64.2** ±0.3	63.3 ±1.4
BoDA-M$_r$	81.9 ±0.2	71.6 ±0.2	87.3 ±0.3	83.4 ±0.2	62.3 ±0.3	65.0 ±2.4
BoDA$_{r,c}$	82.3 ±0.1	72.3 ±0.3	87.1 ±0.2	**83.9** ±0.3	63.2 ±0.2	65.0 ±2.4
BoDA-M$_{r,c}$	**82.4** ±0.2	72.3 ±0.3	87.7 ±0.1	**83.9** ±0.6	**64.2** ±0.3	**66.7** ±2.7
BoDA *vs.* ERM	+1.7	+1.0	-0.1	+2.9	+1.1	+3.4

Table 4-5: **Results on** `TerraInc-MLT`.

Algorithm	Accuracy (by domain)		Accuracy (by shot)			
	Average	Worst	Many	Medium	Few	Zero
ERM [127]	75.3 ±0.3	67.4 ±0.3	85.6 ±0.8	69.6 ±3.2	66.1 ±2.4	14.4 ±2.8
IRM [66]	73.3 ±0.7	64.3 ±1.3	83.5 ±0.6	70.0 ±1.8	58.3 ±3.4	20.1 ±1.4
GroupDRO [134]	72.0 ±0.4	66.6 ±0.2	84.7 ±1.1	64.6 ±4.7	38.9 ±1.2	13.5 ±1.1
Mixup [135]	71.1 ±0.7	60.4 ±1.1	83.2 ±0.7	60.0 ±0.6	56.1 ±3.0	12.2 ±2.1
MLDG [116]	76.6 ±0.2	66.9 ±0.5	86.1 ±0.6	73.8 ±3.9	70.6 ±3.7	18.8 ±2.4
CORAL [111]	76.4 ±0.5	67.8 ±0.9	86.3 ±0.3	77.5 ±3.1	66.1 ±2.0	11.0 ±1.4
MMD [137]	73.3 ±0.4	63.7 ±1.1	84.0 ±0.4	67.9 ±2.7	60.6 ±1.6	13.6 ±2.6
DANN [113]	68.7 ±0.9	61.1 ±1.0	79.6 ±1.2	62.5 ±8.1	48.9 ±2.8	13.3 ±1.1
CDANN [112]	70.3 ±0.5	63.9 ±1.0	83.5 ±0.8	50.0 ±4.2	43.9 ±4.7	20.4 ±3.1
MTL [138]	75.0 ±0.7	67.7 ±1.4	85.2 ±0.7	73.8 ±1.6	61.1 ±2.8	12.4 ±4.0
SagNet [136]	75.1 ±1.6	66.5 ±2.1	85.5 ±0.9	77.1 ±5.0	57.8 ±4.3	13.0 ±3.4
Fish [139]	75.3 ±0.5	66.3 ±0.5	85.8 ±0.2	73.3 ±3.9	61.1 ±3.0	13.7 ±3.3
Focal [99]	75.7 ±0.4	65.3 ±1.1	85.7 ±0.3	76.2 ±3.9	68.9 ±3.2	12.6 ±1.9
CBLoss [72]	78.0 ±0.4	68.3 ±2.0	85.0 ±0.1	89.2 ±1.2	83.9 ±2.5	9.3 ±3.9
LDAM [71]	74.7 ±0.9	64.1 ±1.4	85.1 ±0.6	70.8 ±3.5	67.8 ±1.2	11.1 ±2.4
BSoftmax [100]	76.7 ±1.0	65.6 ±1.3	83.4 ±0.8	90.8 ±0.9	78.3 ±3.9	12.6 ±2.4
SSP [22]	78.5 ±0.7	67.3 ±0.4	85.5 ±1.0	87.8 ±0.9	82.6 ±1.2	13.2 ±2.8
CRT [82]	81.6 ±0.1	70.0 ±0.4	**89.7** ±0.2	90.4 ±0.3	83.9 ±0.5	12.9 ±0.0
BoDA$_r$	78.6 ±0.4	68.5 ±0.3	86.4 ±0.1	85.0 ±1.0	80.0 ±0.9	13.7 ±2.1
BoDA-M$_r$	79.4 ±0.6	71.3 ±0.4	88.4 ±0.3	76.2 ±2.7	88.3 ±1.6	14.4 ±1.4
BoDA$_{r,c}$	82.3 ±0.3	68.5 ±0.6	89.2 ±0.2	**92.5** ±0.9	88.3 ±1.2	21.3 ±0.7
BoDA-M$_{r,c}$	**83.0** ±0.4	**74.6** ±0.7	89.2 ±0.2	91.2 ±0.6	**91.7** ±2.0	**21.7** ±1.4
BoDA *vs.* ERM	+7.7	+7.2	+3.6	+22.9	+25.6	+7.3

achieve higher gains (e.g., CRT); Nevertheless, regardless of dataset, `BoDA` outperforms all methods, highlighting its effectiveness for the MDLT task. Finally, compared to ERM, `BoDA` slightly improves the average and many-shot performance, while substantially boosting the performance for the medium-shot, few-shot, and zero-shot pairs. Table 4-7 summarizes the averaged accuracy across all datasets, where `BoDA` brings large overall improvements of $\sim 3\%$.

A Closer Look at Accuracy Gains. We further explore how `BoDA` performs across *all* domain-class pairs. Fig. 4-7 shows the absolute accuracy gains of `BoDA` over ERM on

Table 4-6: **Results on `DomainNet-MLT`.**

Algorithm	Accuracy (by domain)		Accuracy (by shot)			
	Average	Worst	Many	Medium	Few	Zero
ERM [127]	58.6 ±0.2	29.4 ±0.3	66.0 ±0.1	56.1 ±0.1	35.9 ±0.5	27.6 ±0.3
IRM [66]	57.1 ±0.1	27.6 ±0.1	64.7 ±0.1	54.3 ±0.3	33.5 ±0.3	25.8 ±0.3
GroupDRO [134]	53.6 ±0.1	25.9 ±0.2	61.8 ±0.1	49.1 ±0.3	30.7 ±0.7	22.0 ±0.1
Mixup [135]	57.6 ±0.1	28.7 ±0.0	64.9 ±0.2	54.5 ±0.1	35.6 ±0.2	27.3 ±0.3
MLDG [116]	58.5 ±0.0	28.7 ±0.1	66.0 ±0.1	55.7 ±0.1	35.3 ±0.2	26.9 ±0.3
CORAL [111]	59.4 ±0.1	30.1 ±0.4	66.4 ±0.1	57.1 ±0.0	37.7 ±0.6	29.9 ±0.2
MMD [137]	56.7 ±0.0	27.2 ±0.2	64.2 ±0.1	54.0 ±0.0	33.9 ±0.2	25.4 ±0.2
DANN [113]	55.8 ±0.1	26.9 ±0.4	63.0 ±0.1	52.7 ±0.1	34.2 ±0.4	26.8 ±0.4
CDANN [112]	56.0 ±0.1	27.7 ±0.1	63.2 ±0.0	52.7 ±0.2	34.3 ±0.5	27.6 ±0.1
MTL [138]	58.6 ±0.1	29.3 ±0.2	65.9 ±0.1	56.0 ±0.4	35.4 ±0.1	28.2 ±0.3
SagNet [136]	58.9 ±0.0	29.4 ±0.2	66.3 ±0.1	56.4 ±0.0	36.2 ±0.3	27.2 ±0.4
Fish [139]	59.6 ±0.1	29.1 ±0.1	67.1 ±0.1	57.2 ±0.1	36.8 ±0.4	27.8 ±0.3
Focal [99]	57.8 ±0.2	27.5 ±0.1	65.2 ±0.2	55.1 ±0.2	35.8 ±0.1	26.3 ±0.1
CBLoss [72]	58.9 ±0.1	30.1 ±0.1	64.3 ±0.0	61.0 ±0.3	42.5 ±0.4	28.1 ±0.2
LDAM [71]	59.2 ±0.0	29.2 ±0.2	66.6 ±0.0	57.0 ±0.0	37.1 ±0.2	27.8 ±0.3
BSoftmax [100]	58.9 ±0.1	29.9 ±0.1	64.3 ±0.1	60.9 ±0.3	42.4 ±0.6	28.2 ±0.1
SSP [22]	59.7 ±0.0	31.6 ±0.2	64.3 ±0.1	62.6 ±0.1	45.0 ±0.3	30.5 ±0.0
CRT [82]	60.4 ±0.2	31.6 ±0.1	66.8 ±0.0	61.6 ±0.1	45.7 ±0.1	29.7 ±0.1
BoDA$_r$	60.1 ±0.2	32.6 ±0.1	65.7 ±0.2	60.6 ±0.1	42.6 ±0.3	30.5 ±0.2
BoDA-M$_r$	60.1 ±0.2	32.2 ±0.2	65.9 ±0.2	60.7 ±0.1	42.9 ±0.3	30.0 ±0.1
BoDA$_{r,c}$	**61.7** ±0.1	**33.4** ±0.1	**67.0** ±0.1	62.7 ±0.1	46.0 ±0.2	**32.2** ±0.3
BoDA-M$_{r,c}$	**61.7** ±0.2	33.3 ±0.1	**67.0** ±0.1	**63.0** ±0.3	**46.6** ±0.4	31.8 ±0.2
BoDA *vs.* ERM	+3.1	+4.0	+1.0	+6.9	+10.7	+4.6

Table 4-7: **Results over all MDLT benchmarks.**

Algorithm	`VLCS-MLT`	`PACS-MLT`	`OfficeHome-MLT`	`TerraInc-MLT`	`DomainNet-MLT`	Avg
ERM [127]	76.3 ±0.4	97.1 ±0.1	80.7 ±0.0	75.3 ±0.3	58.6 ±0.2	77.6
IRM [66]	76.5 ±0.2	96.7 ±0.2	80.6 ±0.4	73.3 ±0.7	57.1 ±0.1	76.8
GroupDRO [134]	76.7 ±0.4	97.0 ±0.1	80.1 ±0.3	72.0 ±0.4	53.6 ±0.1	75.9
Mixup [135]	75.9 ±0.1	96.7 ±0.2	81.2 ±0.2	71.1 ±0.7	57.6 ±0.1	76.5
MLDG [116]	76.9 ±0.2	96.6 ±0.1	80.4 ±0.2	76.6 ±0.2	58.5 ±0.0	77.8
CORAL [111]	75.9 ±0.5	96.6 ±0.5	81.9 ±0.1	76.4 ±0.5	59.4 ±0.1	78.0
MMD [137]	76.3 ±0.6	96.9 ±0.1	78.4 ±0.4	73.3 ±0.4	56.7 ±0.0	76.3
DANN [113]	77.5 ±0.1	96.5 ±0.0	79.2 ±0.2	68.7 ±0.9	55.8 ±0.1	75.5
CDANN [112]	76.6 ±0.4	96.1 ±0.1	79.0 ±0.2	70.3 ±0.5	56.0 ±0.1	75.6
MTL [138]	76.3 ±0.3	96.7 ±0.2	79.5 ±0.2	75.0 ±0.7	58.6 ±0.1	77.2
SagNet [136]	76.3 ±0.2	**97.2** ±0.1	80.9 ±0.1	75.1 ±1.6	58.9 ±0.0	77.7
Fish [139]	77.5 ±0.3	96.9 ±0.2	81.3 ±0.3	75.3 ±0.5	59.6 ±0.1	78.1
Focal [99]	75.6 ±0.4	96.5 ±0.2	77.9 ±0.0	75.7 ±0.4	57.8 ±0.2	76.7
CBLoss [72]	76.8 ±0.3	96.9 ±0.1	79.8 ±0.2	78.0 ±0.4	58.9 ±0.1	78.1
LDAM [71]	77.5 ±0.1	96.5 ±0.2	80.3 ±0.2	74.7 ±0.9	59.2 ±0.0	77.7
BSoftmax [100]	76.7 ±0.5	96.9 ±0.3	80.4 ±0.2	76.7 ±1.0	58.9 ±0.1	77.9
SSP [22]	76.1 ±0.3	96.9 ±0.2	81.1 ±0.3	78.5 ±0.7	59.7 ±0.0	78.5
CRT [82]	76.3 ±0.2	96.3 ±0.1	81.2 ±0.0	81.6 ±0.1	60.4 ±0.2	79.2
BoDA$_r$	76.9 ±0.5	97.0 ±0.1	81.5 ±0.1	78.6 ±0.4	60.1 ±0.2	78.8
BoDA-M$_r$	77.5 ±0.3	97.1 ±0.1	81.9 ±0.2	79.4 ±0.6	60.1 ±0.2	79.2
BoDA$_{r,c}$	77.3 ±0.2	**97.2** ±0.1	82.3 ±0.1	82.3 ±0.3	**61.7** ±0.1	80.2
BoDA-M$_{r,c}$	**78.2** ±0.4	97.1 ±0.2	**82.4** ±0.2	**83.0** ±0.4	**61.7** ±0.2	**80.5**
BoDA *vs.* ERM	+1.9	+0.1	+1.7	+7.7	+3.1	+2.9

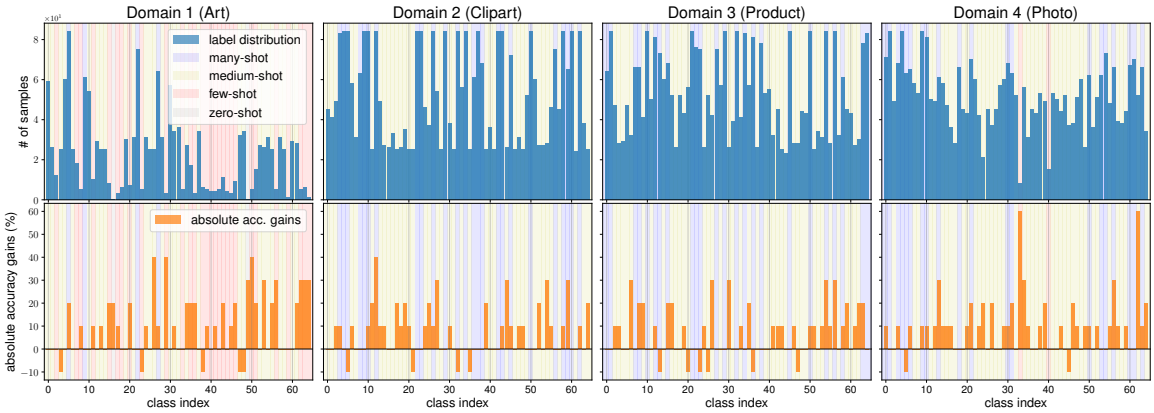


Figure 4-7: **The absolute accuracy improvements of `BoDA` *vs.* ERM over all domain-class pairs on `OfficeHome-MLT`.** `BoDA` establishes large improvements w.r.t. all regions, especially for the few-shot and zero-shot ones. Results for other datasets are in Appendix C.7.2.

`OfficeHome-MLT`, where `BoDA` consistently improves the performance over all domains. The improvements are especially large for domain "Art", where most of the classes lie in the *few-shot* region. For certain classes, `BoDA` can improve up to 50% accuracy, indicating its effectiveness on tackling MDLT.

Ablation Studies on `BoDA` Components (Appendix C.7.1). We study the effects of (1) adding balanced distance (i.e., `BoDA` *vs.* vanilla `DA`), and (2) different choices of distance calibration coefficient $\lambda_{d,c}^{d',c'}$ in `BoDA`. We observe that `BoDA` improves over `DA` by a large margin (2.3% on average over all MDLT datasets), highlighting the importance of using *balanced* distance. Interestingly, as for $\lambda_{d,c}^{d',c'}$, we find that `BoDA` is pretty robust to different

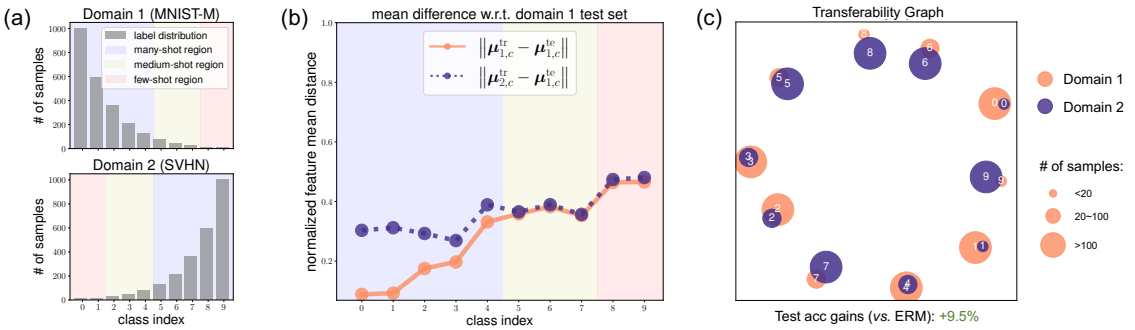


Figure 4-8: BoDA **analysis. (a)** Label distribution setup. **(b)** Distance of feature mean between train and test data. BoDA enables better learned tail (d, c) with smaller feature discrepancy. **(c)** BoDA learns features that are more aligned across domains even in the presence of divergent labels, and significantly improves upon ERM by 9.5%.

Table 4-8: BoDA **bound.**

	$\mathcal{L}_{\texttt{BoDA}}$
Empirical	2.92947 $\pm$7.3e-3
Theoretical	2.92513 $\pm$7.8e-3

choices within a given range, and obtain similar gains (1.9% to 2.9% over ERM).

■ 4.5.2 Understanding the Behavior of BoDA on MDLT

To better understand how the design of BoDA contributes to its ability to outperform other algorithms, we go back to the Digits-MLT dataset, but this time we run BoDA as opposed to ERM.

Better Learned Representations for Minority Data. Similar to Fig. 4-5, we plot in Fig. 4-8b the feature mean distance between training and test data for BoDA on Digits-MLT. The plot shows that BoDA learns better representations with smaller feature discrepancy, especially for minority classes.

Improved Transferability against Severe Imbalance. Fig. 4-8c plots the transferability graph induced by BoDA. It shows that even in the presence of severe and divergent label imbalance (Fig. 4-8a), BoDA still learns transferable features. Further, BoDA learns a *balanced* feature space that separates different classes away. The better learned features translate to better accuracy (9.5% absolute accuracy gains *vs.* ERM in Fig. 4-3c). We provide more related results in Appendix C.7.3 and C.7.5.

Tightness of the Bound. We study whether the BoDA bound derived in Theorem 4 is tight.

Table 4-9: `BoDA` **strengthens performance on Domain Generalization (DG) benchmarks.** Full tables including detailed results for each DG dataset are provided in Appendix C.6.

Algorithm	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Avg
ERM	77.5 ±0.4	85.5 ±0.2	66.5 ±0.3	46.1 ±1.8	40.9 ±0.1	63.3
Current SOTA [111]	**78.8** ±0.6	86.2 ±0.3	68.7 ±0.3	47.6 ±1.0	41.5 ±0.1	64.5
BoDA$_{r,c}$	78.5 ±0.3	**86.9** ±0.4	**69.3** ±0.1	**50.2** ±0.4	**42.7** ±0.1	**65.5**
BoDA$_{r,c}$ + Current SOTA [111]	79.1 ±0.1	87.9 ±0.5	69.9 ±0.2	50.7 ±0.6	43.5 ±0.3	66.2
BoDA *vs.* ERM	**+1.6**	**+2.4**	**+3.4**	**+4.6**	**+2.6**	**+2.9**

We train a ResNet-18 on `Digits-MLT` for 5,000 steps to ensure convergence. We compute the loss over all samples, and combine the results over 3 random seeds. Table 4-8 confirms the bound is empirically tight.

■ 4.6 Beyond MDLT: Imbalanced Domain Generalization

Domain Generalization (DG) refers to learning from multiple domains and generalizing to unseen domains. Since naturally the learning domains differ in their label distributions and may even have class imbalance within each domain, we investigate whether tackling cross-domain data imbalance can further strengthen the performance for DG. Note that all datasets we adapted for MDLT are standard benchmarks for DG, which confirms that data imbalance is an intrinsic problem in DG, but has been overlooked by past works.

We study whether `BoDA` can improve performance for DG. To test `BoDA`, we follow standard DG evaluation protocol [133], and compare to the current SOTA [111]. Table 4-9 reveals the following findings: First, `BoDA` alone can improve upon the current SOTA on four out of the five datasets, and achieves notable average performance gains. Moreover, combined with the current SOTA, `BoDA` further boosts the result by a notable margin across all datasets, suggesting that label imbalance is orthogonal to existing DG-specific algorithms. Finally, similar to MDLT, the gains depend on how severe the imbalance is within a dataset – e.g., `TerraInc` exhibits the most severe label imbalance across domains, on which `BoDA` achieves the highest gains. Detailed results for each DG dataset are provided in Appendix C.6. These intriguing results shed light on how label imbalance can affect out-of-distribution generalization, and highlight the importance of integrating label imbalance for practical DG algorithm design.

4.7 Summary

In this chapter, we formalize the MDLT task as learning from multi-domain imbalanced data, and generalizing to all domain-class pairs. We introduce the domain-class transferability graph, and propose BoDA, a theoretically grounded loss that tackles MDLT. Extensive results on five curated real-world MDLT benchmarks verify its superiority. Furthermore, incorporating BoDA into DG algorithms establishes a new SOTA on DG benchmarks. Our work opens up new avenues for realistic multi-domain learning and generalization in the presence of data imbalance.

CHAPTER 5
A Closer Look at Subpopulation Shift

Machine learning models frequently exhibit drops in performance under the presence of distribution shifts [140]. Constructing machine learning models that are robust to these shifts is critical to the safe deployment of such models in the real-world [141]. One ubiquitous type of distribution shift is *subpopulation shift*, which is characterized by changes in the proportion of some subpopulations between training and deployment [142]. In such settings, models may have high overall performance but still perform poorly in rare subgroups [143, 144].

A well-studied type of subpopulation shift occurs when data contains *spurious correlations* [145] – non-causal relationships between the input and the label which may shift in deployment [146]. For example, image classifiers frequently make use of non-robust features such as image backgrounds [109], textures [147], and erroneous markings [148]. However, there has been little work in defining subpopulation shift in a holistic way, understanding *when* these shifts happen, and *how* state-of-the-art (SOTA) algorithms generalize under diverse and realistic shifts. Subpopulation shift can encompass a much wider array of underlying mechanisms. First, different attributes in data often exhibit skewed distributions, inevitably causing *attribute imbalance* [149]. Moreover, certain labels can have significantly fewer observations, where such long-tailed label distribution induces severe *class imbalance* [69]. Finally, certain attributes may have no training data at all, which motivates the need for *attribute generalization* to unseen subpopulations [150].

In this chapter, we systematically investigate subpopulation shift in realistic evaluation

settings. We first formalize a generic framework of subpopulation shift, which decomposes *attribute* and *class* to enable fine-grained analyses. We demonstrate that this modeling covers and explains the aforementioned common subgroup shifts, which are basic units of building more complex shifts that arise in real data. Using this framework, we can quantify the type and degree of different shift components in each given dataset.

We establish a realistic and comprehensive benchmark of subpopulation shift, consisting of **20** SOTA algorithms that span different learning strategies and **12** real-world datasets in vision, language, and healthcare domains. While existing analysis on subpopulation shift either focus on a single shift type, or have limited severity, our benchmark provides a much larger set of datasets that cover different types of realistic subgroup shifts. Our experimental framework can be easily extended to include new methods, shifts, and datasets.

This chapter also evaluates current algorithms across different settings including attribute availability in training set and/or validation set, model selection strategies, and a wide range of metrics for understanding subpopulation shift in-depth. With the established framework and over 10K trained models, we reveal intriguing observations for future research.

Concretely, in this chapter, we make the following contributions: (i) We formalize a unified framework for subpopulation shift which defines basic types of shift, explains when and why shifts happen, and quantifies their degrees. (ii) We set up a comprehensive and realistic benchmark for systematic subpopulation shift evaluation, with 20 SOTA methods and 12 diverse datasets across various domains. (iii) Based on over 10K trained models, we verify that current algorithms only advance subgroup robustness over certain types of shift identified by our framework, but not others. (iv) We confirm that while successful algorithms rely on the access to group information for model selection, a simple criterion based on worst-class accuracy is surprisingly effective even without group-annotated validation data. (v) We establish the fundamental tradeoff between worst-group accuracy (WGA) and important metrics such as worst-case precision, highlighting the need to rethink evaluation metrics in subpopulation shift beyond WGA.

5.1 Related Work

Subpopulation Shift. Machine learning models frequently experience performance degradation under *subpopulation shift*, where the proportion of some subpopulations differ between the training and test [142, 151]. Depending on the definition of such subpopulations, this could lead to vastly different problem settings. Prior works largely focus on the case of shortcut learning [145], where subpopulations are defined as the product of attributes and labels. In such settings, models trained to minimize overall loss tend to learn spurious correlations, resulting in poor performance in the minority subpopulation [148, 152]. There have been a large set of methods developed to address this scenario, both when the attribute is known [153, 134, 154, 155, 156, 157], and unknown [158, 159, 160, 161].

However, subpopulations may also be defined using only the label. This setting corresponds to class-imbalanced learning, which has also been well studied with extensive proposed methods [22, 3, 71, 72, 104, 162, 163, 164].

Finally, when subpopulations are defined based on a particular attribute (e.g., demographic group) [165, 166], the objective of maximizing performance for the worst-case group then becomes identical to minimax fairness [167, 168].

In this chapter, we present a unified framework of subpopulation shift across these aforementioned scenarios.

Distribution Shift Benchmarks. There have been few prior works which benchmark the performance of subpopulation shift methods. [142] proposed the WILDS benchmark for domain generalization and subpopulation shift, though they only evaluated four methods over five datasets. [169] and [133] proposed the NICO++ and DomainBed benchmarks respectively for domain generalization, and we adapt elements of their benchmark into our subpopulation shift evaluation. [150] proposed the BREEDS benchmark, which consists of multiple datasets constructed from ImageNet [170] using the WordNet hierarchy [171], aiming to evaluate generalization across unseen attributes. Finally, [172] conducted a similar analysis in the general distribution shift setting on four synthetic and two real-world datasets.

Our work differs from these prior works by evaluating a much larger set of algorithms that span different categories on many more real-world datasets. We further define, dissect and quantify the type and degree of shift components in each dataset, and relate it to

Table 5-1: **Formulation summary of basic types of subpopulation shift under our framework.**

Subpopulation Shift Type	Attribute Bias	Class Bias	Impact on Classification Model
Spurious Correlations (SC)	$p_{\text{train}}(a\|y,\mathbf{x}_{\text{core}}) \gg p_{\text{train}}(a\|\mathbf{x}_{\text{core}})$ $p_{\text{test}}(a\|y,\mathbf{x}_{\text{core}}) = p_{\text{test}}(a\|\mathbf{x}_{\text{core}})$	–	$\frac{\mathbb{P}(a\|y,\mathbf{x}_{\text{core}})}{\mathbb{P}(a\|\mathbf{x}_{\text{core}})} \gg 1 \Rightarrow \mathbb{P}(y\|\mathbf{x}) \uparrow$
Attribute Imbalance (AI)	$p_{\text{train}}(a\|y,\mathbf{x}_{\text{core}}) \gg p_{\text{train}}(a'\|y,\mathbf{x}_{\text{core}})$ $p_{\text{test}}(a\|y,\mathbf{x}_{\text{core}}) = p_{\text{test}}(a'\|y,\mathbf{x}_{\text{core}})$	–	$\frac{\mathbb{P}(a\|y,\mathbf{x}_{\text{core}})}{\mathbb{P}(a\|\mathbf{x}_{\text{core}})} \gg \frac{\mathbb{P}(a'\|y,\mathbf{x}_{\text{core}})}{\mathbb{P}(a'\|\mathbf{x}_{\text{core}})} \Rightarrow \mathbb{P}(y\|\mathbf{x}_{\text{core}},a) \gg \mathbb{P}(y\|\mathbf{x}_{\text{core}},a')$
Class Imbalance (CI)	–	$p_{\text{train}}(\mathbf{Y}=y) \gg p_{\text{train}}(\mathbf{Y}=y')$ $p_{\text{test}}(\mathbf{Y}=y) = p_{\text{test}}(\mathbf{Y}=y')$	$\mathbb{P}(y) \gg \mathbb{P}(y') \Rightarrow \mathbb{P}(y\|\mathbf{x}) \gg \mathbb{P}(y'\|\mathbf{x})$
Attribute Generalization (AG)	$p_{\text{train}}(a\|y,\mathbf{x}_{\text{core}}) = 0, \forall a \in \mathbb{A}^{\text{unseen}}$ $p_{\text{test}}(a\|y,\mathbf{x}_{\text{core}}) > 0, \forall a \in \mathbb{A}$	Unconstrained	Generalize to $\mathbb{A}^{\text{unseen}}$

the performance of each method. In addition, we analyze important yet overlooked factors such as model selection criteria and metrics to evaluate against, and reveal intriguing properties in subpopulation shift.

■ 5.2 Unified Framework of Subpopulation Shift

Problem Setup. In the general subpopulation shift setting, given input $\mathbf{x} \in \mathcal{X}$ and label $y \in \mathcal{Y}$, the goal is to learn $f : \mathcal{X} \to \mathcal{Y}$. In addition, there exist attributes $a_1, ..., a_i, ..., a_m$, $a_i \in \mathcal{A}_i$, which may or may not be available when learning f. Then, discrete subpopulations can be defined based on the attribute and label, by some function $h : \mathcal{A} \times \mathcal{Y} \to \mathcal{G}$.

Let $\ell(y, f(\mathbf{x})) \to \mathbb{R}$ be a loss function. Consider the source distribution where $(\mathbf{x}, y)$ are drawn as a mixture of group-wise distributions: $P_{src} = \sum_{g\in\mathcal{G}} \alpha_g P_g$, where $\alpha \in \Delta_{|\mathcal{G}|}$. Further, consider some target distribution which is not observed: $P_{tar} = \sum_{g\in\mathcal{G}} \beta_g P_g$, where $\beta \in \Delta_{|\mathcal{G}|}$. The objective of subpopulation shift is to find [134]:

$$f^* = \arg\min_f \sup_{\beta\in\Delta_{|\mathcal{G}|}} \mathbb{E}_{(\mathbf{x},y)\sim P_{tar}}[\ell(y, f(\mathbf{x}))].$$

This objective is equivalent to minimizing risk for the worst-case group [134], i.e.,

$$f^* = \arg\min_f \max_{g\in\mathcal{G}} \mathbb{E}_{(\mathbf{x},y)\sim P_g}[\ell(y, f(\mathbf{x}))].$$

■ 5.2.1 A Generic Framework for Subpopulation Shift

As motivated earlier, both attribute a and label y can have specific skewed distributions, resulting in distinct types of subpopulation shift. To this end, we propose to decompose the effect of a and y given a multi-group dataset, and characterize general subpopulation

shift into several ***basic shift*** components for fine-grained interpretation.

Specifically, we view each input $\mathbf{x}$ as being fully described or generated from a set of underlying core features $\mathbf{x}_{\text{core}}$ (representing the label) and a list of attributes $\mathbf{a}$ [173, 174]. Here, $\mathbf{x}_{\text{core}}$ denotes the underlying invariant components that are label-specific and support robust classification, whereas attributes $\mathbf{a}$ may have inconsistent distributions and are not label-specific. Such modeling helps us disentangle the attributes and examine how they affect the classification results $\mathbb{P}(y|\mathbf{x})$. Following Bayes' theorem, we can rewrite the classification model as:

$$\begin{aligned}\mathbb{P}(y|\mathbf{x}) &= \frac{\mathbb{P}(\mathbf{x}|y)}{\mathbb{P}(\mathbf{x})} \cdot \mathbb{P}(y) \\ &= \frac{\mathbb{P}(\mathbf{x}_{\text{core}}, \mathbf{a}|y)}{\mathbb{P}(\mathbf{x}_{\text{core}}, \mathbf{a})} \cdot \mathbb{P}(y) \\ &= \underbrace{\frac{\mathbb{P}(\mathbf{x}_{\text{core}}|y)}{\mathbb{P}(\mathbf{x}_{\text{core}})}}_{\text{PMI}} \cdot \underbrace{\frac{\mathbb{P}(\mathbf{a}|y, \mathbf{x}_{\text{core}})}{\mathbb{P}(\mathbf{a}|\mathbf{x}_{\text{core}})}}_{\text{attribute}} \cdot \underbrace{\mathbb{P}(y)}_{\text{class}},\end{aligned} \tag{5.1}$$

where the first term in Eqn. (5.1) represents the pointwise mutual information (PMI) between $\mathbf{x}_{\text{core}}$ and y, the second term corresponds to the potential bias arising in the **attribute** distribution, and the third term explains the potential bias arising in the **class** (label) distribution. Given invariant $\mathbf{x}_{\text{core}}$ between training and testing distributions, we can ignore changes in first term (which is a robust indicator), and focus on how the second and third term, i.e., the *attribute* and *class*, influence the outcomes under subpopulation shift.

More formally, assuming the mutual independence and conditional independence across different attributes a_i [172], we can further decompose the attribute term into a fine-grained version:

$$\frac{\mathbb{P}(\mathbf{a}|y, \mathbf{x}_{\text{core}})}{\mathbb{P}(\mathbf{a}|\mathbf{x}_{\text{core}})} \triangleq \prod_{a_i \in \mathbf{a}} \frac{\mathbb{P}(a_i|y, \mathbf{x}_{\text{core}})}{\mathbb{P}(a_i|\mathbf{x}_{\text{core}})}, \tag{5.2}$$

where each a_i corresponds to an attribute. Note that for benign attributes that are independent of y (i.e., $a_i \perp\!\!\!\perp y, \forall a_i \in \mathbf{a}_{\text{benign}}$), we have $\mathbb{P}(a_i|y, \mathbf{x}_{\text{core}}) = \mathbb{P}(a_i|\mathbf{x}_{\text{core}})$, indicating that the attribute term in Eqn. (5.2) is only driven by *biased* attributes that are label-dependent.

Using the formulation of "attribute-class" decomposition, we can intuitively explain *when* do common subpopulation shifts happen, and *how* they affect the classification results.

Table 5-2: **Overview of the datasets for evaluating subpopulation shift.** Detailed statistics and example data are provided in Appendix D.1.

Dataset	Data type	# Attr.	# Classes	# Train set	# Val. set	# Test set	Max group	Min group	Shift type			
									SC	AI	CI	AG
Waterbirds	Image	2	2	4795	1199	5794	3498 (73.0%)	56 (1.2%)	✓	✓	✓	
CelebA	Image	2	2	162770	19867	19962	71629 (44.0%)	1387 (0.9%)	✓		✓	
MetaShift	Image	2	2	2276	349	874	789 (34.7%)	196 (8.6%)	✓			
ImageNetBG	Image	N/A	9	183006	7200	4050	N/A	N/A				✓
NICO++	Image	6	60	62657	8726	17483	811 (1.3%)	0 (0.0%)		✓	✓	✓
Living17	Image	N/A	17	39780	4420	1700	N/A	N/A				✓
MultiNLI	Text	2	3	206175	82462	123712	67376 (32.7%)	1521 (0.7%)		✓		
CivilComments	Text	8	2	148304	24278	71854	31282 (21.1%)	1003 (0.7%)		✓	✓	
MIMICNotes	Clinical text	2	2	16149	3229	6460	8359 (51.8%)	676 (4.2%)			✓	
MIMIC-CXR	Chest X-rays	6	2	303591	17859	35717	68575 (22.6%)	7846 (2.6%)		✓		
CheXpert	Chest X-rays	6	2	167093	22280	33419	51606 (30.9%)	506 (0.3%)		✓	✓	
CXRMultisite	Chest X-rays	2	2	338134	19891	39781	299089 (88.5%)	574 (0.2%)	✓	✓	✓	

5.2.2 Characterizing Basic Types of Subpopulation Shift

We formally define and characterize four basic types of subpopulation shift using our framework: *spurious correlations*, *attribute imbalance*, *class imbalance*, and *attribute generalization* (see Table 5-1). In practice, we note that dataset often consists of multiple types of shift instead of one. The four cases constitute the *basic* shift units, and are important elements to explain complex subgroup shifts in real data.

Spurious Correlations (SC). Spurious correlations happen when certain a is spuriously correlated with y in training but not in test data. Under our framework, it implies that $p_{\text{train}}(a|y, \mathbf{x}_{\text{core}}) \gg p_{\text{train}}(a|\mathbf{x}_{\text{core}})$, which is not true of p_{test}. As a result, it introduces bias to the *attribute term*, which induces higher prediction confidence for certain label once given its spuriously correlated attribute (details in Table 5-1).

Attribute Imbalance (AI). Attributes often incur biased distributions in the wild. In our framework, it happens when certain attributes are sampled with a much smaller probability than others in p_{train}, but not in p_{test}. To disentangle the effect of labels, we assume no class bias under this basic shift. As such, it again affects the *attribute term* in Eqn. (5.1) where $p_{\text{train}}(a|y, \mathbf{x}_{\text{core}}) \gg p_{\text{train}}(a'|y, \mathbf{x}_{\text{core}})$, causing lower prediction confidence for underrepresented attributes.

Class Imbalance (CI). Similarly, class labels can exhibit imbalanced distributions, causing lower preference for minority labels. Within our framework, CI can be explained by biasing the *class term* in p_{train}, leading to higher prediction confidence for majority classes.

Attribute Generalization (AG). Certain attributes can be totally missing in p_{train}, but present in p_{test}, which motivates the need for attribute generalization. In our framework, this translates to $p_{\text{train}}(a|y, \mathbf{x}_{\text{core}}) = 0, a \in \mathbb{A}^{\text{unseen}}$, yet we have $p_{\text{test}}(a|y, \mathbf{x}_{\text{core}}) > 0$. AG requires learning robust $\mathbf{x}_{\text{core}}$ in order to generalize across unseen attributes, which is harder but more ubiquitous in real data [150].

5.3 Benchmarking Subpopulation Shift

Datasets. We explore subpopulation shift using **12** real-world datasets from a variety of modalities and tasks. First, for **vision** datasets, we use `Waterbirds` [175] and `CelebA` [176], which are commonly used in the spurious correlation literature [158]. Similarly, we use the `MetaShift` cats *vs.* dogs dataset [177]. We further convert the ImageNet backgrounds challenge (`ImageNetBG`) [178], the `NICO++` [169] benchmark, and the `Living17` dataset from the BREEDS benchmark [150] for subpopulation shift. Further, for **language** understanding datasets, we leverage `CivilComments` [179] and `MultiNLI` [180], which are commonly used text datasets in subpopulation shift. Finally, we curate 4 datasets in the **medical** domain. We construct `MIMIC-CXR` [181] and `CheXpert` [182] to predict the presence of any pathology from a chest X-ray. We also construct `MIMICNotes` for mortality classification from clinical notes [183]. Finally, we follow a recent work in evaluating subgroup shift and construct the `CXRMultisite` dataset [184]. Table 5-2 reports the details of each dataset. We leave full information and descriptions for each of the datasets in Appendix D.1.1.

Algorithms. We evaluate **20** algorithms that span a broad range of learning strategies and categories, and relate their performance to different shifts defined in our framework. We believe this is the first work to comprehensively evaluate a large set of diverse algorithms in subpopulation shift. Concretely, these algorithms cover the following areas: (1) *vanilla:* **ERM** [127], (2) *subgroup robust methods:* **GroupDRO** [134], **CVaRDRO** [185], **LfF** [186], **JTT** [158], **LISA** [153], **DFR** [154], (3) *data augmentation:* **Mixup** [187], (4) *domain-invariant feature learning:* **IRM** [66], **CORAL** [111], **MMD** [137], (5) *imbalanced learning:* **ReSample** [188], **ReWeight** [188], **Focal** [99], **CBLoss** [72], **LDAM** [71], **BSoftmax** [100], **CRT** [82], **ReWeightCRT** [82]. Our framework can be easily extended to include new algorithms. We provide detailed descriptions for each algorithm in Appendix D.1.2.

Evaluation Metrics. Existing works on subpopulation shift mainly report *worst-group ac-*

curacy (WGA) as the gold-standard. While WGA faithfully assesses worst-group performance, other important metrics (e.g., worst-case precision, calibration error, etc.) are also essential especially when involving subpopulation shift. Therefore, in our benchmark we include a variety of metrics aiming for a thorough evaluation from different aspects. In particular, besides **Avg Accuracy** and **Worst Accuracy**, we further include **Avg Precision**, **Worst Precision**, **Avg F1-score**, **Worst F1-score**, (Class-)**Balanced Accuracy**, **Adjusted Accuracy** (accuracy on a *group*-balanced dataset), and expected calibration error (**ECE**) [189]. Detailed summaries of all metrics are in Appendix D.1.3.

Attribute Availability. Whether attribute is known in both (1) *training set* and (2) *validation set* has long been a vital factor for almost all subgroup algorithms [154]. Specifically, classic methods (e.g., GroupDRO) assume access to attributes during training to define meaningful groups. Recently, a number of methods (e.g., JTT, LfF, DFR) try to improve worst-group accuracy without knowing the training attributes. Nevertheless, current approaches still require access to group-annotated validation set for model selection and hyperparameter tuning [160].

We systematically investigate this phenomenon by considering three settings in our benchmark: (1) *attributes are known in both training & validation*, (2) *attributes are unknown in training, but known in validation*, and (3) *attributes are unknown in both training & validation*. Note that when training attributes are unknown, methods that operate over *subgroups* degenerate to operate over *classes*. Without further specification, we report results under the third setting, which is the hardest but the most realistic one. We include full results across all settings in Appendix D.4.

Model Selection. As mentioned earlier, model selection becomes essential when attributes are completely unknown. Significant drop (over 20%) in worst-group test accuracy has been observed if using the highest *average* validation accuracy as the model selection criterion without any group annotations [160]. To this end, we provide a rigorous analysis on different model selection strategies, especially when attributes are fully unknown. Further details are provided in Appendix D.1.4.

Implementation. For a fair evaluation, following [133], for each algorithm we conduct a random search of 16 trials over a joint distribution of all hyperparameters (details are provided in Appendix D.2). We then use the validation set to select the best hyperparame-

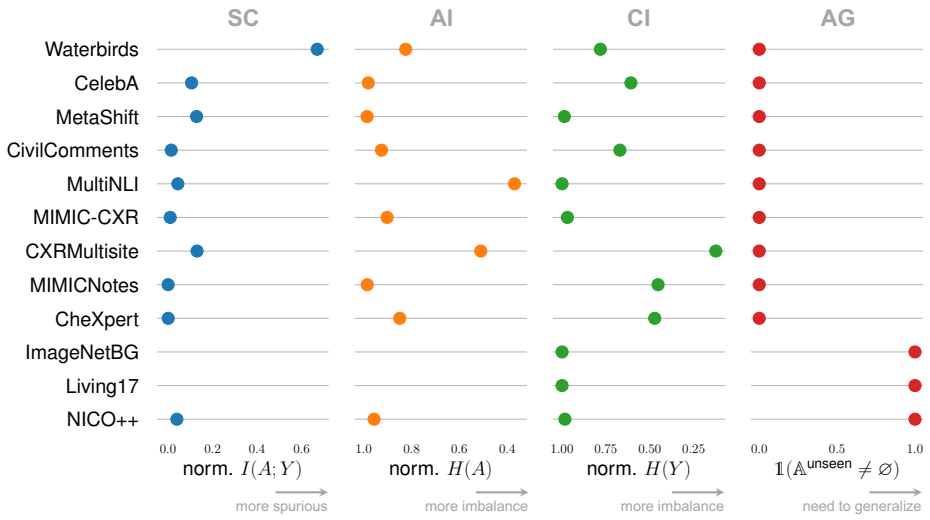


Figure 5-1: **Quantification of the degree of different shifts over all datasets.** Additional metrics are provided in Appendix D.3.1.

ters for each algorithm, fix them and rerun the experiments under three different random seeds to report the final average results with standard deviation. Such process ensures the comparison is best-versus-best, and the hyperparameters are optimized for all algorithms.

5.4 A Fine-Grained Analysis

5.4.1 Quantifying Subpopulation Shift

In order to quantify the degree of each shift for each dataset relative to others, we use several simple metrics. For *spurious correlations*, we use the normalized mutual information between A and Y, where norm $I(A;Y) = 1$ means that the two are perfectly correlated: norm $I(A;Y) = \frac{2I(A;Y)}{H(Y)+H(A)}$.

For *attribute* and *class imbalance*, we use the normalized entropy, where norm $H(Y) = 1$ indicates that the distribution is uniform (i.e., no imbalance): norm $H(Y) = \frac{H(Y)}{\log |\text{supp}(Y)|}$.

For *attribute generalization*, we simply examine whether there exist any subpopulations in the test set which do not appear during training via an indicator function (see Fig. 5-1). We provide several additional metrics in Appendix D.3.1.

We find that different datasets exhibit very different types of shift, and the degrees also greatly vary (Fig. 5-1). To further study how algorithms perform across various types of shift, we categorize each dataset into its most dominant shift type.

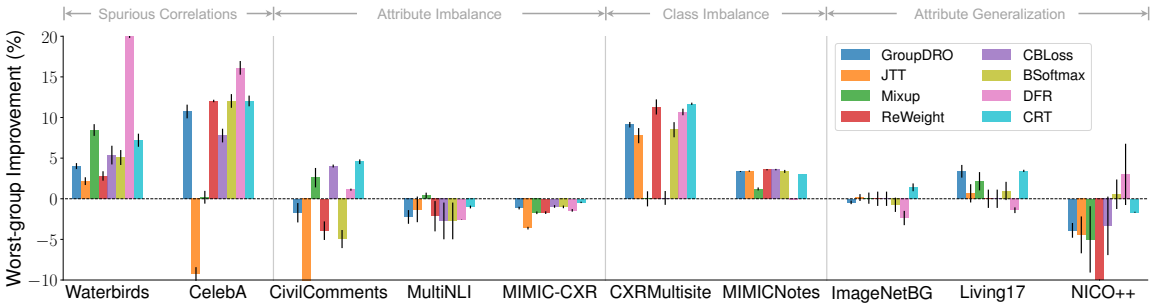


Figure 5-2: **Worst-group improvements over ERM across different datasets when attributes are *unknown* in both training and validation set.** SOTA algorithms only enhance subgroup robustness on certain types of shift (i.e., **SC** and **CI**). Complete results are in Appendix D.3.2.

■ 5.4.2 Performance across Different Types of Shift

As described earlier, we run experiments for all algorithms, datasets, and attribute availability settings. We use *worst-group accuracy* as the model selection criterion, and provide analysis for other metrics in Appendix D.3.3. When attributes are unknown in the validation set, this criterion degenerates to *worst-class accuracy*. Interestingly, we discover that this simple method is surprisingly effective (related results in Sec. 5.4.4). In total, we trained over 10,000 models.

We study model performance over different shifts. Specifically, we report results when attributes are *unknown* in both training and validation. Results for other settings are in Appendix D.3.2. We present main results in Fig. 5-2 and Table 5-3, where we make intriguing observations as follows.

SOTA algorithms only improve subgroup robustness on certain types of shift, but not others. As Fig. 5-2 illustrates, for *spurious correlations* and *class imbalance*, existing algorithms can provide consistent worst-group gains over ERM even in the absence of validation attributes, indicating that progress has been made for tackling these two specific shifts. Interestingly however, when it comes to *attribute imbalance*, little improvement is observed across datasets. In addition, the performance becomes even worse for *attribute generalization*. These findings stress that current advances are only made for specific shifts (i.e., SC and CI), while no progress has been made for the more challenging shifts such as AG.

Methods that decouple representation and classifier are more effective. When further zoom into the performance across all datasets in Table 5-3, a set of methods that decou-

Table 5-3: **Results on all tested subpopulation benchmarks, when attributes are *unknown* in both training and validation set.** Full results for each dataset and other settings are in Appendix D.4. Methods that re-train classifier using a two-stage strategy are marked in gray.

Algorithm	Waterbirds	CelebA	CivilComments	MultiNLI	MetaShift	ImageNetBG	NICO++	MIMIC-CXR	MIMICNotes	CXRMultisite	CheXpert	Living17	Avg
ERM	69.1 ±4.7	57.6 ±0.8	63.2 ±1.2	66.4 ±2.3	82.1 ±0.8	76.8 ±0.9	35.0 ±4.1	68.6 ±0.2	80.4 ±0.2	50.1 ±0.9	41.7 ±3.4	27.7 ±1.1	59.9
Mixup	77.5 ±0.7	57.8 ±0.8	65.8 ±1.5	66.8 ±0.3	79.0 ±0.8	76.9 ±0.7	30.0 ±4.1	66.8 ±0.6	81.6 ±0.6	50.1 ±0.9	37.4 ±3.5	29.8 ±1.8	60.0
GroupDRO	73.1 ±0.4	68.3 ±0.9	61.5 ±1.8	64.1 ±0.8	83.1 ±0.7	76.4 ±0.2	31.1 ±0.9	67.4 ±0.5	83.7 ±0.1	59.2 ±0.3	74.7 ±0.3	31.1 ±1.0	64.5
CVaRDRO	75.5 ±2.2	60.2 ±3.0	62.9 ±3.8	48.2 ±3.4	83.5 ±0.5	74.8 ±0.8	27.8 ±2.3	68.0 ±0.2	65.6 ±1.5	50.2 ±0.9	50.2 ±1.8	27.3 ±1.6	57.8
JTT	71.2 ±0.5	48.3 ±1.5	51.0 ±4.2	65.1 ±1.6	82.6 ±0.4	77.0 ±0.4	30.6 ±2.3	64.9 ±0.3	83.8 ±0.1	57.9 ±2.1	60.4 ±4.8	28.3 ±1.1	60.1
LfF	75.0 ±0.7	53.0 ±4.3	42.2 ±7.2	57.3 ±5.7	72.3 ±1.3	70.1 ±1.4	28.8 ±2.0	62.2 ±2.4	84.0 ±0.1	50.1 ±0.9	13.7 ±9.8	26.4 ±1.3	52.9
LISA	77.5 ±0.7	57.8 ±0.8	65.8 ±1.5	66.8 ±0.3	79.0 ±0.8	76.9 ±0.7	30.0 ±4.1	66.8 ±0.6	81.6 ±0.6	50.1 ±0.9	37.4 ±3.5	29.8 ±1.8	60.0
ReSample	70.0 ±1.0	74.1 ±2.2	61.0 ±0.6	66.8 ±0.5	81.0 ±1.7	77.7 ±1.1	30.6 ±2.3	67.5 ±0.3	82.6 ±0.6	55.0 ±0.2	74.3 ±0.4	31.4 ±0.6	64.3
ReWeight	71.9 ±0.6	69.6 ±0.2	59.3 ±1.1	64.2 ±1.9	83.1 ±0.7	76.8 ±0.9	25.0 ±0.0	67.0 ±0.4	84.0 ±0.1	61.4 ±1.3	73.7 ±1.0	27.7 ±1.1	63.6
SqrtReWeight	71.0 ±1.4	66.9 ±2.2	68.6 ±1.1	63.8 ±2.4	82.6 ±0.4	76.8 ±0.9	32.8 ±3.5	68.0 ±0.4	83.1 ±0.2	61.2 ±0.6	68.5 ±1.6	27.7 ±1.1	64.2
CBLoss	74.4 ±1.2	65.4 ±1.4	67.3 ±0.2	63.6 ±2.4	83.1 ±0.0	76.8 ±0.9	31.7 ±3.6	67.6 ±0.3	84.0 ±0.1	50.2 ±0.9	74.0 ±0.7	27.7 ±1.1	63.8
Focal	71.6 ±0.8	56.9 ±3.4	61.9 ±1.1	62.4 ±2.0	81.0 ±0.4	71.9 ±1.2	30.6 ±2.3	68.7 ±0.4	70.9 ±9.8	50.0 ±0.9	42.1 ±4.0	26.9 ±0.6	57.9
LDAM	70.9 ±1.7	57.0 ±4.1	28.4 ±7.7	65.5 ±0.8	83.6 ±0.4	76.7 ±0.5	31.7 ±3.6	66.6 ±0.6	81.0 ±0.3	50.1 ±0.9	36.0 ±0.7	24.3 ±0.8	56.0
BSoftmax	74.1 ±0.9	69.6 ±1.2	58.3 ±1.1	63.6 ±2.4	82.6 ±0.4	76.1 ±2.0	35.6 ±1.8	67.6 ±0.6	83.8 ±0.3	58.6 ±1.8	73.8 ±1.0	28.6 ±1.4	64.4
DFR	89.0 ±0.2	73.7 ±0.8	64.4 ±0.1	63.8 ±0.0	81.4 ±0.1	74.4 ±1.8	38.0 ±3.8	67.1 ±0.4	80.2 ±0.0	60.8 ±0.4	75.8 ±0.3	26.3 ±0.4	66.2
CRT	76.3 ±0.8	69.6 ±0.7	67.8 ±0.3	65.4 ±0.2	83.1 ±0.0	78.2 ±0.5	33.3 ±0.0	68.1 ±0.1	83.4 ±0.0	61.8 ±0.1	74.6 ±0.4	31.1 ±0.1	66.1
ReWeightCRT	76.3 ±0.2	70.7 ±0.6	64.7 ±0.2	65.2 ±0.2	85.1 ±0.4	77.5 ±0.7	33.3 ±0.0	67.9 ±0.1	83.4 ±0.0	53.1 ±2.3	75.1 ±0.2	33.1 ±0.1	65.4

ple the training of representation and classifier [154, 82] achieve remarkable gains over all other algorithms (highlighted in gray). As prior works also confirmed [154], features learned by ERM seem to be good enough under spurious correlations. These findings inspire us to further understand the role of *representation* and *classifier* in subpopulation shift, especially their behaviors under different subgroup shifts.

■ 5.4.3 The Role of Representation and Classifier

We are motivated to explore the role of representation and classifier in subpopulation shift. In particular, we separate the whole network into two parts: the feature extractor and the classifier. We then employ three training strategies for representation and classifier learning, respectively: (1) *uniform*, which follows the normal ERM training; (2) *balanced sampling*, where balanced samples are drawn from each group (class if attribute not available) during training, and (3) *re-weighting*, where we re-weight all the samples by the inverse of the sample size of their groups (classes). Note that classifier re-balancing resembles CRT [82] and DFR [154]. We train models following the above settings across all datasets, and average the results over datasets according to the type of shift.

Representation & classifier quality play different roles under different shifts. As Fig. 5-3 reveals, for **SC** and **CI**, balanced classifier learning (i.e., both re-sampling and re-weighting) can substantially improve the performance when fixing the representation, whereas different representation learning schemes do not lead to notable gains when fixing the classifier learning manner. Interestingly, for **AI**, balancing the classifier does not lead to better per-

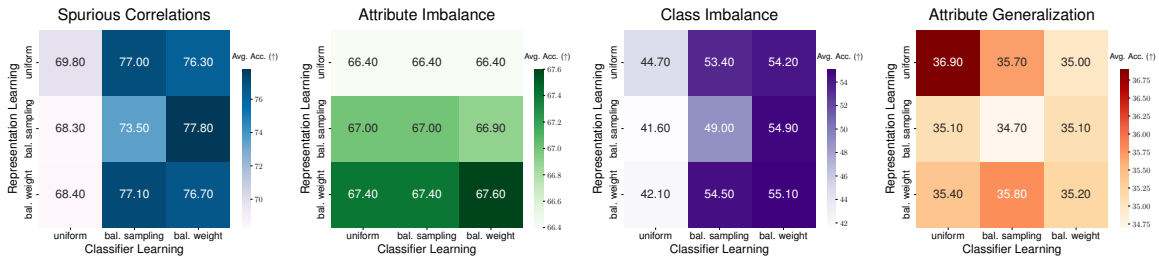


Figure 5-3: **Averaged worst-group accuracy of different manners for representation learning and classifier learning under different shifts.** Within each shift type, we average the results across datasets that belong to this shift to report the final accuracy. As observed, balanced classifier learning substantially improves the results for **SC** and **CI**, while balanced representation learning gives reasonable gains for **AI**; Yet, no stratified learning manners lead to performance gains under **AG** compared to vanilla ERM. Experimental details are in Section 5.4.3.

Table 5-4: **Relative improvements over ERM when using stratified balanced representation or classifier learning under different shifts.**

	SC	AI	CI	AG
REPRESENTATION	**-0.3**	**+1.1**	**-0.2**	**-0.4**
CLASSIFIER	**+8.1**	**+0.0**	**+11.9**	**-0.4**

formance, while balanced representation schemes can bring notable gains.

ERM features are not sufficient for subpopulation shift. Unlike recent works that claim ERM features are sufficient for out-of-distribution generalization [190, 154], our above intriguing findings suggest that features learned via ERM may only be good enough for **certain** shifts. Concretely, improving the feature extractor still leads to notable gains especially for **AI**. The results in turn well explain the performance differences in Fig. 5-2, that SOTA algorithms with two-stage training do not improve worst-case accuracy under **AI** or **AG**.

Stratified balanced learning does not outperform ERM under AG. Finally, no stratified learning manners lead to performance gains under **AG**. As Table 5-4 summarizes, both stratified representation and classifier learning manners even exhibit negative gains for datasets that require **AG**. This reveals the intrinsic limitation of SOTA algorithms [154] against diverse types of subpopulation shift.

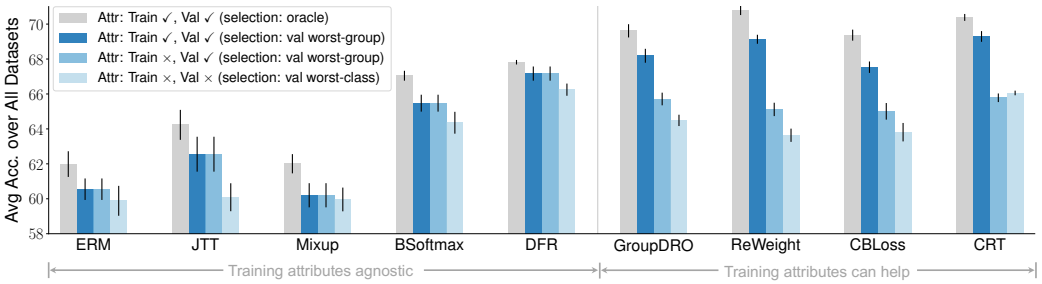


Figure 5-4: **Averaged worst-group accuracy of various algorithms under different model selection and attribute availability settings.**

Table 5-5: **Test-set worst-group accuracy difference (%) between each selection strategy on each dataset, relative to the oracle which selects the best worst-group accuracy.** Complete results across all datasets and all selection strategies are provided in Appendix D.3.3.

Selection Strategy	CelebA	CheXpert	CivilComments	MIMIC-CXR	MIMICNotes	MetaShift	Avg
Max Worst-Class Accuracy	-5.0 ±6.3	**-0.4** ±0.8	**-3.2** ±5.2	**-0.9** ±1.0	**-0.1** ±0.5	**-1.5** ±3.0	**-1.8**
Max Balanced Accuracy	**-4.4** ±5.4	-1.3 ±2.5	-3.5 ±5.8	-2.9 ±4.9	-2.3 ±6.2	-1.7 ±3.0	-2.7
Min Class Accuracy Diff	-6.1 ±9.1	-1.9 ±5.3	-4.1 ±8.0	-1.9 ±5.0	-0.3 ±1.2	-2.2 ±4.6	-2.7
Max Worst-Class F1	-13.4 ±10.4	-5.4 ±6.7	**-3.2** ±3.8	-2.5 ±2.2	-4.4 ±8.7	-1.8 ±3.3	-5.1
Max Overall AUROC	-12.2 ±10.3	-10.4 ±13.0	-8.2 ±9.0	-6.6 ±9.9	-10.0 ±16.5	-3.2 ±7.0	-8.4
Max Overall Accuracy	-18.6 ±12.0	-30.9 ±24.9	-13.7 ±9.5	-5.1 ±6.3	-19.9 ±26.0	-1.9 ±3.3	-15.0

5.4.4 On Model Selection and Attribute Availability

Model selection (e.g., choice of hyperparameters, training checkpoints) and attribute availability affect subpopulation shift evaluation considerably, especially given that almost all SOTA algorithms need access to a group-annotated validation set for model selection [160]. We study this problem in-depth, where we follow three settings mentioned earlier (i.e., the availability of both *training* and *validation* attributes), and summarize the results in Fig. 5-4.

The importance of training attribute availability relies on algorithm properties. As Fig. 5-4 verifies, when training attribute is available, it can greatly boost the performance of algorithms that need group information (e.g., GroupDRO), while it does not bring benefits for attribute-agnostic methods (e.g., ERM, JTT).

Validation attribute may not be necessary once you have a good selection metric. We further investigate the performance without validation attributes. It is widely known that SOTA subpopulation shift methods rely on group labels for validation. Surprisingly however, we observe a relatively small accuracy drop over all methods when using a simple *worst-class accuracy* (degenerated from *worst-group* as attributes are unknown in validation)

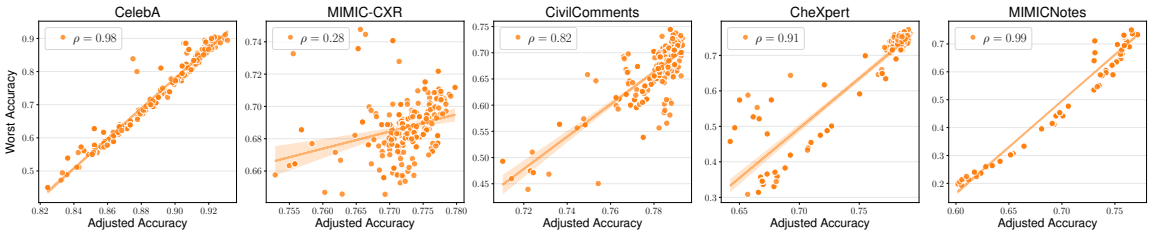
as selection metric. Specifically, comparing the last two bars across all methods in Fig. 5-4, the average accuracy drop is less than merely 2%. This striking finding contrasts with the literature, where large degradation (over 20%) is observed when using *average* accuracy as the metric without validation attributes. This suggests that if carefully choosing a metric for model selection, we can achieve minimal worst-group accuracy loss even in the absence of any attribute information.

Simple selection criterion using worst-class accuracy is surprisingly effective even without validation attribute. We examine different strategies for choosing when to stop during model training when no attribute annotations are available in both training and validation. We select six representative datasets and six representative selection strategies, respectively (full results across all datasets and all selection strategies are in Appendix D.3.3). For each model, we utilize each stopping criterion over the validation set metrics computed throughout training, to determine its corresponding stopping point. We evaluate a variety of selection criteria in this way for a large variety of methods trained on each dataset. We compare each strategy with the oracle selection criteria, summarizing our results in Table 5-5. We observe that simply stopping when the *worst-class accuracy* reaches a maxima achieves the best worst-group accuracy on average. As expected, any selection criterion based on overall performance (e.g., accuracy, AUROC) performs much worse.

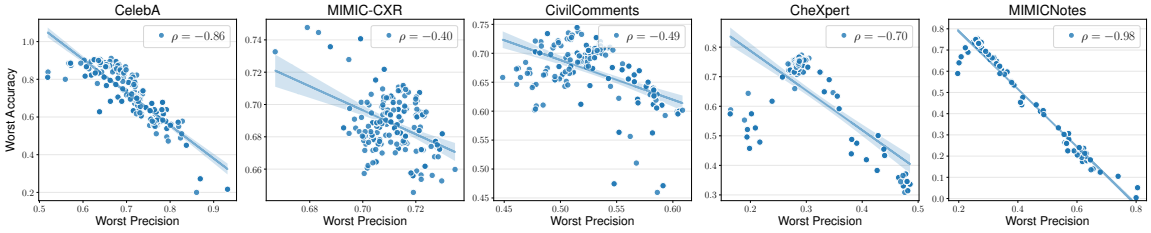
■ 5.4.5 Metrics Beyond Worst-Group Accuracy

Worst-group accuracy (WGA) has long been treated as the gold-standard for assessing the model performance in subpopulation shift. Recent studies also discovered that WGA and model average performance are linearly correlated, a phenomenon called "*Accuracy on the line*" [191, 154]. However, WGA essentially assesses the worst-case (top-1) recall conditioned on attribute [3], which does not reflect other important metrics such as worst-case precision and calibration error. Whether models with high WGA will also perform better across these metrics remains unknown. Therefore, we further examine the relationship between WGA and other evaluation metrics that proposed in our benchmark.

Intrinsic tradeoff: Accuracy can be on the inverse line. Interestingly, we observe that not all metrics are positively correlated with WGA. In particular, we show scatter plots of WGA *vs.* other metrics for representative datasets. As Fig. 5-5(a) confirms, adjusted accuracy is linearly correlated with WGA, which is well aligned with existing observations



(a) **Accuracy on the line:** Adjusted accuracy is *positively* correlated with WGA.



(b) **Accuracy on the inverse line:** Worst-case precision is *negatively* correlated with WGA.

Figure 5-5: **Fundamental tradeoff between WGA and other evaluation metrics.** Complete results for all metrics are in Appendix D.3.4.

[154]. Interestingly however, for worst-case precision, the *positive* correlation does not hold anymore; instead, we observe a strong ***negative*** linear correlation, indicating an intrinsic tradeoff between WGA and worst-case precision. We show in Appendix D.3.4 that many metrics also possess such "*accuracy on the* ***inverse*** *line*" property, further verifying the inherent tradeoff between testing metrics.

Fundamental limitations of WGA as the only metric. The above observations highlight the complex relationship between WGA and other metrics: Certain metrics display high positive correlation, while many others show the opposite case. This finding uncovers the fundamental limitation of using only WGA to assess model performance in subpopulation shift: A well performed model with high WGA can however have low worst-case precision, which is alarming especially in critical applications such as medical diagnosis (e.g., `CheXpert`). Our observations emphasize the need for more realistic evaluation metrics in subpopulation shift.

■ 5.4.6 Further Analysis

Impact of model architecture (Appendix D.3.5). We study the effect of different model architectures on subpopulation shift across various datasets and modalities. In particular, we employ ResNets and vision transformers (ViTs) for the image modality, and five dif-

ferent transformer-based language models for the text modality. We observe that on text datasets, base BERT models are already competitive over other architecture variants (Table D-8). Yet, the results on image datasets are mixed when comparing the worst-group performance for ResNets and ViTs (Tables D-9 and D-10).

Impact of pretraining methods (Appendix D.3.5). We investigate how different pretraining methods affect the model performance under subpopulation shift. We consider both *supervised* and *self-supervised* pretraining using various SOTA methods. Similar to previous findings [154], we observe that supervised pretraining outperforms self-supervised counterparts for most of the experiments. The results may also suggest that better self-supervised schemes could be developed for tackling subgroup shifts.

Impact of pretraining datasets (Appendix D.3.5). Finally, we investigate whether increasing the pretraining dataset size could lead to better subgroup performance. We leverage ImageNet-21K [192] and SWAG [193] in addition to the default ImageNet-1K. Interestingly, we find consistent and significant worst-group performance gains when going from ImageNet-1K to ImageNet-21K to SWAG, indicating that larger and more diverse pretraining datasets seem to increase worst-group performance.

■ 5.5 Limitations and Broader Impacts

Limitations. We acknowledge several limitations of our benchmark and analyses. First, we have used 12 real-world predictive datasets in our benchmark. However, real-world data can have many complexities including potential mislabelling in both attributes and labels. We do not consider this effect, though it would be interesting to examine it in a synthetic setting. Moreover, prior work has shown that in the case of multiple spurious attributes, reducing reliance on one can increase reliance on another [194]. We only consider a single attribute in this benchmark, though an evaluation of this effect in the context of model selection criteria would be an interesting direction of future research.

Potential Negative Impacts. There are several potential negative social impacts of our work. First, we assume throughout the work that we would like to have models that are robust to subpopulation shift. However, in practice, this comes at the cost of overall accuracy on the training distribution. There may be cases where the practitioner would like to maximize overall accuracy regardless of spurious correlations, and thus subpopulation

shift methods would worsen overall performance and potentially cause excess harm. Next, we recognize that the large grid of deep models trained for our evaluations likely resulted in a significant carbon footprint [195]. However, we hope that the insights provided in this chapter will reduce the number of models and training steps (and therefore carbon emissions) required by future practitioners. Finally, we have constructed several models in this chapter that utilize clinical data for clinical predictive tasks. We do not advocate for blind deployment of these models in any way, as there are many issues that need to be verified and resolved before their deployment, such as real-world clinical testing, privacy, fairness, interpretability, and regulatory requirements [196, 197].

5.6 Summary

In this chapter, we systematically study the subpopulation shift problem, formalize a unified framework to define and quantify different types of subpopulation shift, and further set up a comprehensive benchmark for realistic evaluation. Our benchmark includes 20 SOTA methods and 12 real-world datasets across different domains. Based on over 10K trained models, we reveal several intriguing properties in subpopulation shift that have implications for future research, including divergent performance on different shifts, model selection criteria, and metrics to evaluate against. We hope our benchmark and findings will promote realistic and rigorous evaluations and inspire new advances in subpopulation shift.

Part II

Applications: Extending Healthcare Beyond Clinics

CHAPTER 6

Artificial Intelligence-Enabled Detection and Assessment of Parkinson's Disease using Nocturnal Breathing Signals

Parkinson's disease (PD) is the fastest-growing neurological disease in the world [27]. Over one million people are living with PD in the US as of 2020 [28], resulting in an economic burden of $52 billion per year [29]. Today, no drugs can reverse or stop the progression caused by the disease [10]. A key difficulty in PD drug development and disease management is the lack of effective diagnostic biomarkers [198]. The disease is typically diagnosed based on clinical symptoms, mainly related to motor functions such as tremor and rigidity [199]. However, motor symptoms tend to appear several years after the onset of the disease, leading to late diagnosis [10]. Thus, there is a strong need for novel diagnostic biomarkers, particularly ones that can detect the disease at an early stage.

There are also no effective progression biomarkers for tracking the severity of the disease over time [198]. Today, assessment of PD progression relies on patient self-reporting or qualitative rating by a clinician [200]. Typically, clinicians use a questionnaire called the Movement Disorder Society Unified Parkinson's Disease Rating Scale (MDS-UPDRS) [201]. The MDS-UPDRS is semi-subjective and does not have enough sensitivity to capture

small changes in patient status [202, 203, 204]. As a result, PD clinical trials need to last multiple years before changes in MDS-UPDRS can be reported with sufficient statistical confidence [202, 205], which increases cost and delays progress [206].

The literature has investigated a few potential PD biomarkers, among which cerebrospinal fluid [207, 208], blood biochemical [209], and neuroimaging [210] have good accuracy. However, these biomarkers are costly, invasive, and require access to specialized medical centers, and as a result are not suitable for frequent testing to provide early diagnosis or continuous tracking of disease progression.

In this chapter, we present a novel AI-based system that detects PD, predicts disease severity, and tracks disease progression over time using nocturnal breathing. Our system delivers a diagnostic and progression digital biomarker that is objective, non-obtrusive, low-cost, and can be measured repeatedly in the patient's home. A relationship between PD and breathing was noted as early as 1817, in the work of Dr. James Parkinson [211]. This link was further strengthened in later work which reported degeneration in areas in the brainstem that control breathing19 [212], weakness of respiratory muscle function [213], and sleep breathing disorders [214, 215, 216, 217]. Further, these respiratory symptoms often manifest years before clinical motor symptoms [213, 216, 218], which indicates that the breathing attributes could be promising for risk assessment prior to clinical diagnosis.

The AI-based system is illustrated in Fig. 6-1. It takes as input one night of breathing signals, which can be collected using a breathing belt worn on the person's chest or abdomen [30]. Alternatively, the breathing signals can be collected without wearable devices by transmitting a low power radio signal and analyzing its reflections off the person's body [219, 220, 6]. The nocturnal breathing is passed as input to our neural network, which analyses it to produce two outputs: (1) it predicts whether the person has PD, and (2) it estimates the severity of PD in terms of the MDS-UPDRS. The neural network leverages transfer learning and multi-task learning (details in Sec. 6.2). Transfer learning allows the model to transfer knowledge across different data types (i.e., breathing-belt data and radio-based data). Multi-task learning addresses the limited supervision of PD labels (only one bit for a full-night of nocturnal breathing). Specifically, the model is made to learn the auxiliary task of predicting the person's quantitative electroencephalogram (qEEG) from nocturnal breathing, which prevents the model from overfitting and helps interpreting the

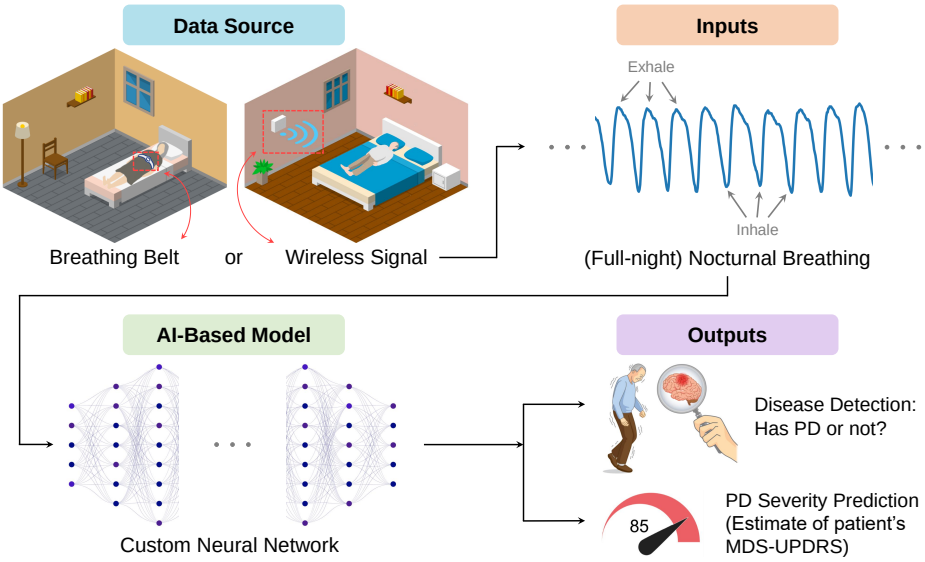


Figure 6-1: **Overview of the AI model for PD diagnosis and disease severity prediction from nocturnal breathing signals.** The system extracts nocturnal breathing signals either from a breathing belt worn by the subject, or from radio signals that bounce off their body while asleep. It processes the breathing signals using a neural network to infer whether the person has PD, and if they do, assess the severity of their PD in accordance with the MDS-UPDRS.

model's output (as explained in Sec. 6.3).

The resulting biomarker is noninvasive and can be collected in the person's home. The measurements can be repeated longitudinally to track changes in disease severity, and hence disease progression. Further, by using a contactless wireless sensor to extract breathing, the measurements can be collected passively and continuously without effort from patients or caregivers.

■ 6.1 Datasets

We use a large and diverse dataset created by pulling multiple datasets from several sources including Mayo Clinic, MGH sleep lab, observational PD clinical trials sponsored by the Michael J. Fox Foundation (MJFF) and the NIH Udall Center, an observational study conducted by MIT, and public sleep datasets from the National Sleep Research Resource such as the Sleep Heart Health Study (SHHS) [30] and the MrOS Sleep Study (MrOS) [221]. The combined dataset contains 11,964 nights with over 120,000 hours of nocturnal breathing signals from 757 PD subjects (mean (SD) age 69.1 (10.4), 27% women) and 6,914 control subjects (mean (SD) age 66.2 (18.3), 30% women). Fig. 6-2 summarizes the datasets.

Data source	Data type	Source of breathing signals	# of PD patients	# of controls	MDS-UPDRS	Hoehn and Yahr stage	# of nights per subject
Mayo Clinic (External test cohort)	PSG sleep study (sampled from the population visiting the Mayo Clinic sleep lab)	Breathing Belt	644	1276	–	PD: 2.2 (1.0) Control: N/A	1 (0)
Sleep Heart Health Study Visit 2 (SHHS)	PSG sleep study (heart disease, sleep disorders)	Breathing Belt	13	2617	–	–	1 (0)
MrOS Sleep Study (MrOS)	PSG sleep study (sleep disorders, vascular disease)	Breathing Belt	48	2827	–	–	1.4 (0.5)
Massachusetts General Hospital (MGH) study	PSG sleep study (sampled from the population visiting the MGH sleep lab)	Breathing Belt	27	120	PD: 39.8 (17.4) Control: N/A	PD: 2.2 (0.4) Control: N/A	1 (0)
Massachusetts General Hospital (MGH) study	Sleep study (sampled from the population visiting the MGH sleep lab)	Wireless	0	8	N/A	N/A	9.5 (4.0)
Udall study	Observational clinical study in PD	Wireless	14	6	PD: 61.1 (20.1) Control: 1.8 (2.0)	PD: 2.3 (0.6) Control: 0.2 (0.4)	86.7 (67.2)
Michael J. Fox Parkinson's study	Observational clinical study in PD	Wireless	11	4	PD: 58.3 (19.3) Control: 7.0 (1.9)	PD: 2.2 (0.7) Control: 0 (0)	35.1 (19.1)
MIT study	Sleep study (healthy volunteers)	Wireless	0	56	N/A	N/A	18.7 (24.4)

Figure 6-2: **Characteristics of the datasets used in this study.** PSG refers to polysomnography; hyphens indicate fields with unavailable data; N/A indicates fields for which the data are inapplicable.

The data is divided to two groups: the breathing belt datasets and the wireless datasets. The first group comes from polysomnography (PSG) sleep studies and uses a breathing belt to record the person's breathing throughout the night. The second group collects nocturnal breathing in a contactless manner using a radio device [219]. The radio sensor is deployed in the person's bedroom, and analyzes the radio reflections from the environment to extract the person's breathing signal [220, 6]. The breathing belt datasets have only one or two nights per person and lack MDS-UPDRS and H&Y scores. In contrast, the wireless datasets include longitudinal data for up to one year and MDS-UPDRS and H&Y scores, allowing us to validate the model's predictions of PD severity and its progression. Since some individuals in the wireless datasets are fairly young (e.g., in their 20's or 30's), when testing on the wireless data, we limit ourselves to the PD patients and their age-matched control subjects (i.e., 10 control subjects from the Udall and MJFF studies and 18

age and gender-matched subjects from the MIT and MGH studies for a total of 28 control individuals). Control subjects missing MDS-UPDRS or H&Y scores receive the mean value for the control group.

6.1.1 Data Pre-Processing

The datasets are divided into two groups. The first group comes from polysomnography (PSG) sleep studies. Such studies use a breathing belt to record the subject's breathing signals throughout the night. They also include EEG and sleep data. The PSG datasets are the Sleep Heart Health Study (SHHS) [30] (n=2,630 nights from 2,630 subjects), the MrOS Sleep Study (MrOS) [221] (n=3,883 nights from 2,875 subjects), and the MGH sleep dataset (n=223 nights from 155 subjects). Further, an external PSG dataset from Mayo Clinic (n=1,920 nights from 1,920 subjects) is held back during the AI model development, and serves as an independent test set. The second group of datasets collects nocturnal breathing in a contactless manner using a radio device developed by our team at MIT [219]. The data is collected by installing a low-power radio sensor in the subject's bedroom, and analyzing the radio reflections from the environment to extract the subject's breathing signal as described in our prior work [220, 6]. This group includes the Michael J. Fox dataset (n=526 nights from 15 subjects), the Udall dataset (n=1,734 nights from 20 subjects) and the MIT dataset (n=1,048 nights from 56 subjects). The wireless datasets have multiple nights per subject and information about PD severity such as the MDS-UPDRS and/or the Hoehn and Yahr stage [222].

We process the data to filter out nights shorter than 2 hours. We also filter out nights where the breathing signal is distorted or non-existent, which occurs when the person does not wear the breathing belt properly for breathing belt data, and when a source of interference (e.g., fans or pets) exists near the subject for wireless data. We normalize the breathing signal from each night by clipping values larger than a particular range (we use $[-6, +6]$), subtracting the mean of the signal, and dividing by the standard deviation. The resulting breathing signal is a 1-D time series $x \in R^{1 \times f_b T}$, with a sampling frequency f_b of 10 Hz, and a length of T seconds.

We use the following variables to determine whether a participant has PD: *"Drugs Used To Treat Parkinson's"* for SHHS, and *"Has a doctor or other health care provider ever told you that you had Parkinson's disease?"* for MrOS. The other datasets explicitly report whether the

person has PD, and for those who do have PD, they provide their MDS-UPDRS and H&Y stage.

In the experiments involving distinguishing Parkinson's from Alzheimer's disease (AD), we use the following variables to identify AD patients: *"Acetylcholine Esterase Inhibitors For Alzheimer's"* for SHHS, and *"Has a doctor or other health care provider ever told you that you had dementia or Alzheimer's disease?"* for MrOS.

6.2 Methods

6.2.1 Training and Testing Protocols

Subjects used in training the neural network were not used for testing. We performed k-fold cross-validation (k=4) for PD detection, and leave-one-out validation for severity prediction. We also assessed cross-institution prediction by training and testing the model on data from different medical centers. Furthermore, data from Mayo Clinic was kept as external data, never seen during development or validation, and used only for a final test.

6.2.2 Sensing Breathing using Radio Signals

By capturing breathing signals using radio signals, our system can run in a completely contactless manner. To do so, we leverage past work on extracting breathing signals from radio frequency (RF) signals that bounce off people's bodies. The RF data is collected using a multi-antenna Frequency-Modulated Continuous Waves (FMCW) radio, which is commonly used in passive health monitoring [220, 6]. The radio sweeps the frequencies from 5.4 GHz to 7.2 GHz and transmits at sub-milliwatt power in accordance with FCC regulations, and captures the reflections from the environment. The radio reflections are processed to infer the subject's breathing signals. Past work shows that respiration signals extracted in this manner are highly accurate, even when multiple people sleep in the same bed [220, 219]. In this chapter, we extract the participant's breathing signal from the RF signal using the method developed by [220, 6], which has been shown to work well even in the presence of bed partners, producing an average correlation 0.914 with an FDA-approved breathing belt on the person's chest. We have further confirmed their accuracy results in a diverse population by collecting wireless signals and breathing belt data from 326 subjects attending the MGH sleep lab, and running the above method to extract breath-

ing signals from RF signals. The RF-based breathing signals have an average correlation of 0.91 with the signals from a breathing belt on the subject's chest.

6.2.3 The AI-Based Model

We use a neural network to predict whether a subject has PD, and the severity of their PD in terms of the MDS-UPDRS. The neural network takes as input a night of nocturnal breathing. The neural network consists of a breathing encoder, a PD encoder, a PD classifier and a PD severity predictor.

- **Breathing Encoder:** We first use a breathing encoder to capture the temporal information in breathing signals. The encoder $E(\cdot)$ uses eight layers of 1-D bottleneck residual blocks [64], followed by three layers of simple recurrent units (SRU) [223].
- **PD Encoder:** We then use a PD encoder to aggregate the temporal breathing features into a global feature representation. The PD encoder $G(\cdot)$ is a self-attention network [64]. It feeds the breathing features into two convolution layers with a stride of 1 followed by a normalization layer to generate the attention scores for each breathing feature. It then calculates the time average of the breathing features weighted by the corresponding attention scores as the global PD feature $G(E(x)) \in R^{d\times 1}$, where d is the fixed dimension of the global feature.
- **PD Classifier:** The PD classifier $M(\cdot)$ is composed of three fully-connected layers and one sigmoid layer. The classifier outputs the PD diagnosis score $M(G(E(x))$, which is a number between 0 and 1. The person is considered to have PD if the score exceeds 0.5.
- **PD Severity Predictor:** The PD severity predictor $N(\cdot)$ is composed of four fully connected layers. It outputs the PD severity estimation $N(G(E(x))$, which is an estimate of the subject's MDS-UPDRS score.

Multi-task Learning. To tackle the sparse supervision from PD labels (i.e., only one label for $\sim$10 hours of nocturnal breathing signals), we introduce an auxiliary task of predicting a summary of the patient's quantitative electroencephalogram (qEEG) during sleep. The auxiliary task provides additional labels (from the qEEG signal), which help regularize the model during training. We chose qEEG prediction as our auxiliary task because EEG is

related to both PD [224, 225] and breathing [226]. Further, the datasets collected during sleep studies have EEG signals, making the labels accessible.

To generate the qEEG label, we first transform the ground-truth time series EEG signals into the frequency domain using the short-time Fourier transform (STFT) and the Welch's periodogram method [227]. We extract the time series EEG signals from the C4-M1 channel, which is commonly used and available in sleep studies [30, 221]. We then decompose the EEG spectrogram into the Delta (0.5–4 Hz), Theta (4–8 Hz), Alpha (8–13 Hz), and Beta (13–30 Hz) bands [225, 224, 228], and normalize the power to obtain the relative power in each band every second.

- **qEEG Predictor:** The qEEG predictor $F(\cdot)$ takes as input the encoded breathing signals, and predicts the relative power in each EEG band at that time. It consists of three layers of 1-D deconvolution blocks, which up-sample the extracted breathing features to the same time resolution as the qEEG signal, and two fully connected layers. Each 1-D deconvolution block contains three deconvolution layers followed by batch normalization, ReLU activation, and a residual connection. We also use a skip connection by concatenating the output of SRU layers in the breathing encoder to the deconvolution layers in the qEEG predictor, which follows the UNet structure [64, 229]. The predicted qEEG is $F(E(x))$.

Transfer Learning. Our model leverages transfer learning to enable a unified model that works with both a breathing belt and a contactless radio sensor of breathing signals, and transfers the knowledge between different datasets.

- **Domain-Invariant Transfer Learning.** Note that our breathing signals are extracted from both breathing belts and wireless signals. There could exist a domain gap between these two data types, which makes jointly learning both of them less effective. To deal with this issue, we adversarially train the breathing encoder to ensure that the latent representation is domain invariant [230]. Specifically, we introduce a discriminator $D_{PD}(\cdot)$ that differentiates features of breathing belt from features of wireless signals, for PD patients. We then add an adversarial loss to the Breathing Encoder that makes the features indistinguishable by $D_{PD}(\cdot)$. Similarly, we introduce a second discriminator $D_{Control}(\cdot)$ with a corresponding adversarial loss for control subjects. We use two discriminators

because the ratio of PD to control individuals is widely different between the wireless datasets and the breathing belt datasets (59% of the wireless data is from individuals with PD, whereas less than 2% of the breathing belt data comes from individuals with PD). If one uses a single discriminator, the discriminator may end up eliminating some features related to PD as it tries to eliminate the domain gap between the wireless dataset and the breathing-belt dataset.

- **Transductive Consistency Regularization.** For PD severity prediction (i.e., predicting the MDS-UPDRS), since we have multiple nights for each subject, the final PD severity prediction for each subject can further leverage the information that PD severity does not change over a short period (e.g., one month). Therefore, the prediction for one subject across different nights should be consistent, i.e., the PD severity prediction for different nights should be the same. To enforce this consistency, we add a consistency loss on the predictions of different nights (samples) for the same subject.

Distribution Calibration. Since the percentage of individuals with PD is quite different between the wireless data and breathing belt data, we further calibrate the output probability of the PD classifier $M(\cdot)$ to ensure that all data types have the same threshold for PD diagnosis (i.e., 0.5). Specifically, during training, we randomly split training samples into four subsets of equal size, and we use three of them for training, and the remaining one for calibration. We apply Platt Scaling [231] to calibrate the predicted probability for PD diagnosis. After training a model using three subsets, we use the remaining calibration subset to learn two scalars $A, B \in R$ and calibrate the model output by $\widehat{y}_c = \sigma(A\widehat{y} + B)$, where $\widehat{y}$ is the original model output, $\widehat{y}_c$ is the calibrated result, and $\sigma(\cdot)$ is a sigmoid function [232]. The cross-entropy loss between $\widehat{y}_c$ and y is minimized in the calibration subset. This process is repeated four times, with each subset used once for calibration, leading to four calibrated models. Our final model is the average ensemble of these models.

Training Details. At each epoch, we randomly sampled a full-night nocturnal breathing signal as a mini-batch of the input. The total loss in general contains a weighted cross-entropy loss of PD classification, a weighted regression loss of MDS-UPDRS regression, an L_2 loss of qEEG prediction, a discriminator loss of which domain the input comes from, and a transductive consistency loss of minimizing the difference of the severity prediction across all nights from the same subject. For each specific input nocturnal breathing sig-

nal, its total loss depends on the existing labels for this night. If one kind of label is not available, the corresponding loss term is excluded from the total loss. During training, the weights of the model are randomly initialized, and we use Adam optimizer [64] with a learning rate of 1e-4. The neural network model is trained on multiple NVIDIA TITAN Xp graphical processing units using the PyTorch deep learning library.

6.2.4 Statistical Analysis

PD Diagnosis and PD Severity Prediction. Intraclass correlation coefficient (ICC) was used to assess test-retest reliability for both PD diagnosis and PD severity prediction. To evaluate PD severity prediction, we assessed the correlation between our model predictions (median value from all nights used) and clinical PD outcome measures (MDS-UPDRS total score) at the baseline visit using a Pearson correlation. We further compared the aggregated mean values among groups with different Hoehn and Yahr stages using Kruskal-Wallis test ($\alpha = 0.05$).

Risk Assessments Prior to Clinical Diagnosis. In addition, we assessed the capability of our AI-based system to identify high-risk individuals prior to actual diagnosis. For PD diagnosis, we compared the aggregated predictions between the prodromal group and the control group using one-tailed Wilcoxon rank-sum test ($\alpha = 0.05$). For PD severity prediction, we again used one-tailed Wilcoxon rank-sum test ($\alpha = 0.05$) to assess the PD severity prediction between the prodromal group and the control group.

Longitudinal Disease Progression Analysis. We evaluated the AI model predictions on the disease severity across longitudinal data. To assess the disease progression over one year, we aggregated the one-year MDS-UPDRS change values over all patients, and used one-tailed one-sample Wilcoxon signed-rank test ($\alpha = 0.05$) to assess the significance of 6-month and 12-month MDS-UPDRS change for both clinician assessment and our model prediction. For continuous severity prediction across one year, we further compared the aggregated model predictions with an interval length of one month using Kruskal-Wallis test ($\alpha = 0.05$).

qEEG and Sleep Statistics Comparison Between PD and Control Subjects. Finally, we assessed the distribution difference between control and PD subjects using aggregate attention score associated with different EEG bands and sleep status. To do so, we used a

one-tailed Wilcoxon rank-sum test ($\alpha = 0.05$) for statistical analysis between the PD group and the control group.

All statistical analyses were performed with Python version 3.7 (Python Software Foundation) and R version 3.6 (R Foundation).

6.2.5 Evaluation Methods

To evaluate the performance of PD severity prediction, we use the Pearson correlation. To evaluate the performance of PD classification, we use sensitivity, specificity, receiver operating characteristic (ROC) curves, and area under the ROC curve (AUC). When reporting the sensitivity and specificity, we use a classification threshold of 0.5 for both data from breathing belt, and data from wireless signals. We follow standard procedures to calculate the 95% confidence interval for sensitivity and specificity [233].

We also evaluate the test-retest reliability. This is a common test for identifying the lower bound on the amount of data aggregation necessary to achieve a desirable statistical confidence in the repeatability of the result. The test-retest reliability is evaluated using the intraclass correlation coefficient (ICC) [234]. To compute the ICC, we divide the longitudinal data into time windows. We use the month immediately after the baseline visit. Using more than a month of data is undesirable since a key requirement for test-retest reliability analysis is that for each patient, the disease severity and symptoms have not changed during the period included in the analysis. We choose one month because this period is short enough to assume that the disease has not changed, and long enough to analyze various time windows for assessing reliability. From that period, we include all available nights.

6.3 Results

6.3.1 Evaluation of Parkinson's Disease Diagnosis

We evaluate the accuracy of diagnosing PD from one night of nocturnal breathing. Fig. 6-3a and 6-3b show the receiver operating characteristic (ROC) curves for data from breathing belt and data from wireless signals, respectively. The AI model detects PD with high accuracy. For nights measured using a breathing belt, the model achieves an AUC of 0.889 with a sensitivity of 80.22% (95% CI [70.28%, 87.55%]) and specificity of 78.62% (95% CI [77.59%, 79.61%]). For nights measured using wireless signals, the model achieves an AUC of 0.906

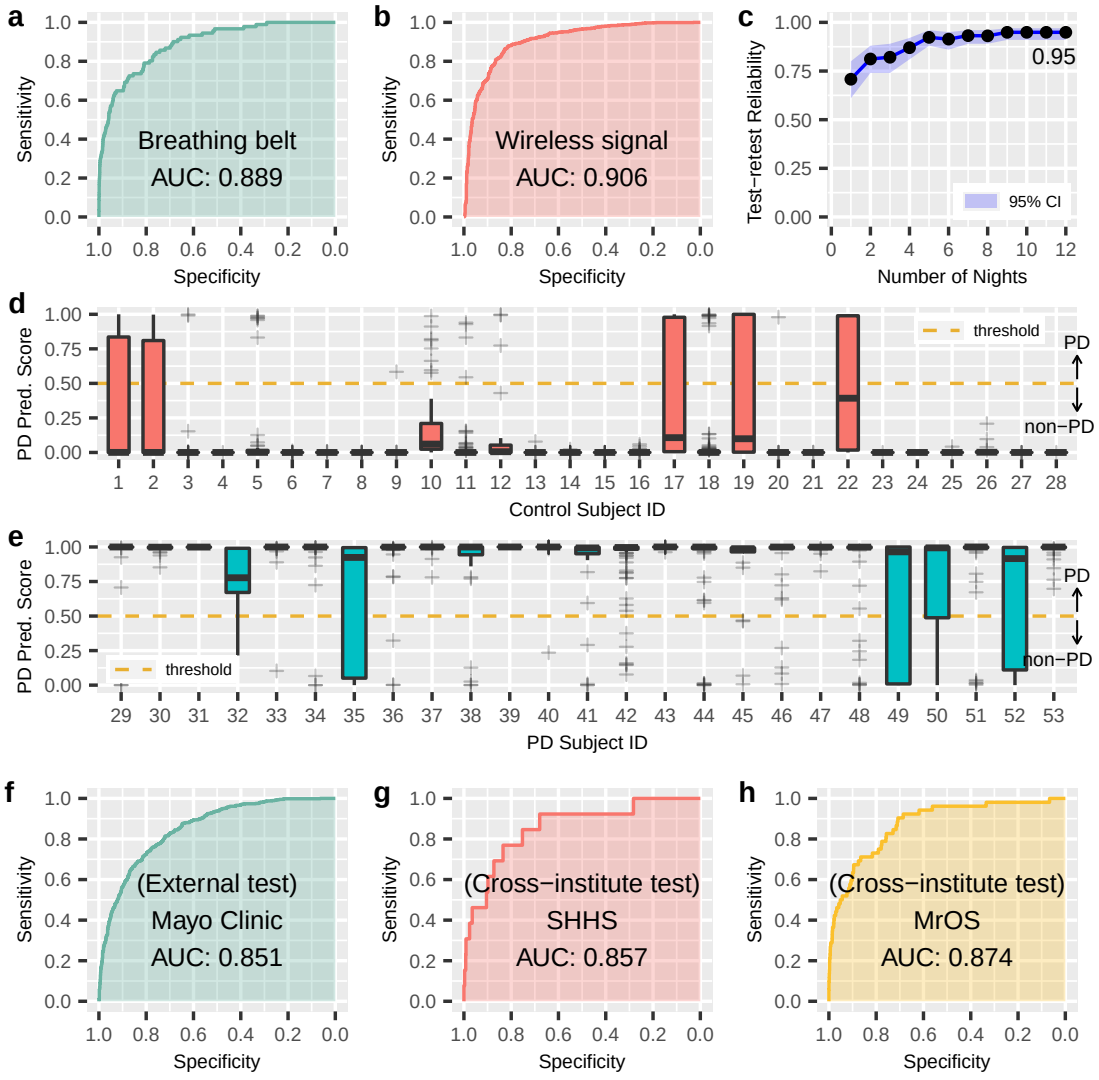


Figure 6-3: **PD diagnosis from nocturnal breathing signals. a, b,** Receiver operating characteristic (ROC) curves for detecting PD from breathing belt data (n=6,660 nights from 5,652 subjects) and wireless data (n=2,601 nights from 53 subjects) respectively. **c,** Test-retest reliability of PD diagnosis as a function of the number of nights used by the AI model. The test was performed on one month of data from each subject in the wireless dataset (n=53 subjects). The dots and the shadow denote the mean and 95% CI, respectively. The model achieved a reliability of 0.95 (95% CI [0.92, 0.97]) with 12 nights of data. **d, e,** Distribution of PD prediction scores for subjects with multiple nights (n1=1,263 nights from 25 PD subjects and n2=1,338 nights from 28 age- and gender-matched controls). The graphs show a boxplot of the prediction scores as a function of the subject ids. **f,** ROC curves for detecting PD on an external test set from Mayo Clinic (n=1,920 nights from 1,920 subjects). The model has an AUC of 0.851 with a sensitivity of 80.12% and specificity of 72.65%. **g, h,** Cross-institution PD prediction performance. In this analysis, all data from one institution was held back as test data, and the AI model was retrained excluding all data from that institution. Cross-institution prediction achieved an AUC of 0.857 with a sensitivity of 76.92% and specificity of 83.45% on SHHS, and an AUC of 0.874 with a sensitivity of 82.69% and specificity of 75.72% on MrOS.

with a sensitivity of 86.23% (95% CI [84.08%, 88.13%]) and specificity of 82.83% (95% CI [79.94%, 85.40%]).

We further investigate whether the accuracy improves by combining multiple nights from the same individual. We use the wireless datasets since they have multiple nights per subject (mean (SD) 61.3 (42.5)), and compute the model prediction score for all nights. The PD prediction score is a continuous number between 0 and 1, where the subject is considered to have PD if the score exceeds 0.5. We use the median PD score for each subject as the final diagnosis result. As Fig. 6-3d and 6-3e show, with multiple nights considered for each subject, both sensitivity and specificity of PD diagnosis further increase to 100% for the PD and control subjects in this cohort.

Next, we compute the number of nights needed to achieve a high test-retest reliability31. We use the wireless datasets, and compute the test-retest reliability by averaging the prediction across consecutive nights within a time window. The results show that the reliability improves when we use multiple nights from the same subject, and reaches 0.95 (95% CI [0.92, 0.97]) with only 12 nights (Fig. 6-3c).

■ 6.3.2 Generalization to External Test Cohort

To assess the generalizability of our model across different institutions with different data collection protocols and patient populations, we validated our AI model on an external test dataset (n=1,920 nights from 1,920 subjects out of which 644 have PD) from an independent hospital not involved during model development (Mayo Clinic). Our model achieved an AUC of 0.851 (Fig. 6-3f). The performance indicates that our model can generalize to diverse data sources from institutions not encountered during training.

We also examined the cross-institution prediction performance by testing the model on data from one institution, but training it on data from the other institutions excluding the test institution. For breathing belt data, and as highlighted in Fig. 6-3g and 6-3h, the model achieved a cross-institution AUC of 0.857 on SHHS and 0.874 on MrOS. For wireless data, the cross-institution performance was 0.892 on MJFF, 0.884 on Udall, 0.974 on MGH, and 0.916 on MIT. These results show that the model is highly accurate on data from institutions it never seen during training. Hence, the accuracy is not due to leveraging institution related information, or misattribution of institution-related information to the disease.

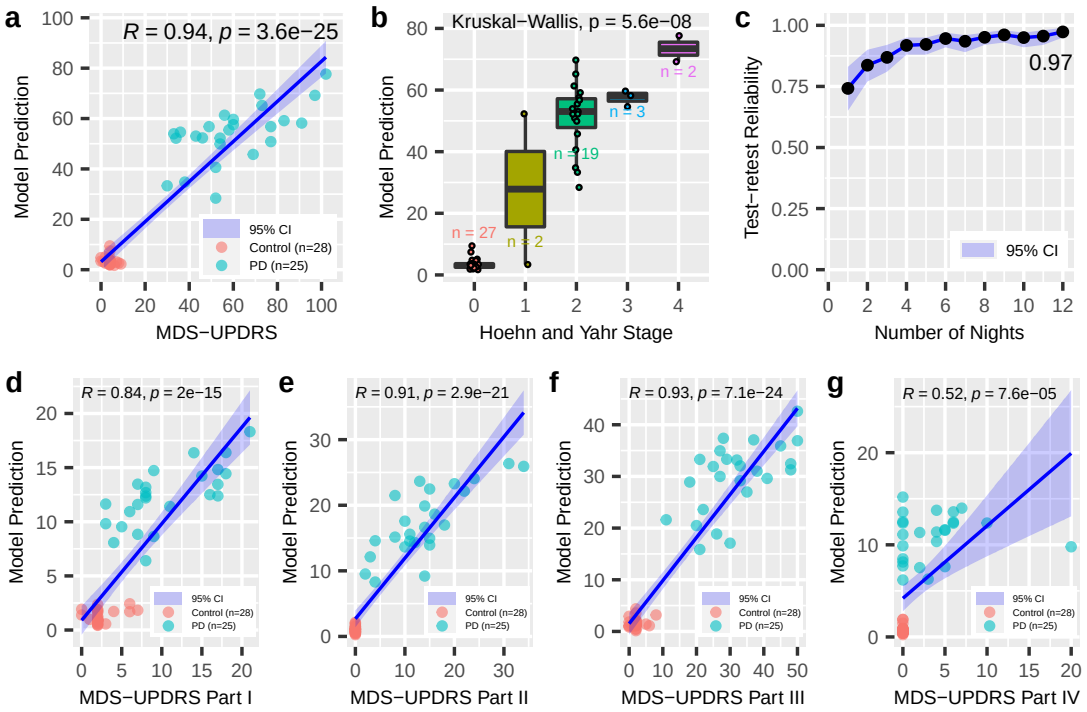


Figure 6-4: **PD severity prediction from nocturnal breathing signals. a,** The Pearson correlation coefficient of the model with MDS-UPDRS is R = 0.94 (p = 3.6e-25, two-sided t-test). The center line and the shadow denote the mean and 95% CI, respectively. **b,** Severity prediction distribution of the model with respect to the Hoehn and Yahr stage; a higher Hoehn and Yahr stage indicates increased PD severity (p = 5.6e-08, Kruskal-Wallis test). On each box, the central line indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to 1.5 times the interquartile range. **c,** Test-retest reliability of PD severity prediction as a function of the number of nights per subject. The dots and the shadow denote the mean and 95% CI, respectively. The model achieved a reliability of 0.97 (95% CI [0.95, 0.98]) with 12 nights of data. **d-g,** Correlations of the AI model predictions with different sub-parts of MDS-UPDRS (Part I: R = 0.84, p = 2e-15; Part II: R = 0.91, p = 2.9e-21; Part III: R = 0.93, p = 7.1e-24; Part IV: R = 0.52, p = 7.6e-05; all using two-sided t-test). The center line and the shadow denote the mean and 95% CI, respectively. Data in all panels is from the wireless dataset (n=53 subjects).

■ 6.3.3 Evaluation of Parkinson's Disease Severity Prediction

Today the MDS-UPDRS is the most common method for evaluating PD severity, with higher scores indicating more severe impairment. Evaluating MDS-UPDRS requires effort from both patients and clinicians: patients are asked to visit the clinic in-person and evaluations are performed by trained clinicians who categorize symptoms based on quasi-subjective criteria [202].

We evaluate the ability of our model to produce a PD severity score that correlates well with the MDS-UPDRS simply by analyzing the patients' nocturnal breathing at home. We

use the wireless dataset where MDS-UPDRS assessment is available, and each subject has multiple nights of measurements (n=53 subjects, 25 PD subjects with a total of 1,263 nights and 28 controls with a total of 1,338 nights). We compare the MDS-UPDRS at baseline with the model's median prediction computed over the nights from the one-month period following the subject's baseline visit. Fig. 6-4a shows strong correlation between the model's severity prediction and the MDS-UPDRS (R = 0.94, p = 3.6e-25), providing evidence that the AI model can capture PD disease severity.

We also study the feasibility of predicting each of the four sub-parts of MDS-UPDRS (i.e., predicting subparts I, II, III, and IV). This is done by replacing the module for predicting the total MDS-UPDRS by a module that focuses on the sub-part of interest, while keeping all the other components of the neural network unmodified. Fig. 6-4d-g show the correlation between the model's prediction and the different sub-parts of MDS-UPDRS. We observe a strong correlation between the model's predictions and Part I (R = 0.84, p = 2e-15), Part II (R = 0.91, p = 2.9e-21), and Part III (R = 0.93, p = 7.1e-24) scores. This indicates that the model captures both non-motor (e.g., Part I), and motor symptoms (e.g., Part II and III) of PD. The model's prediction has mild correlation with Part IV (R = 0.52, p = 7.6e-05). This may be caused by the large overlap between PD and control subjects in Part IV scores (i.e., most of the PD patients and control subjects in the studied population have a score of 0 for Part IV).

We also compare our model's severity prediction with the Hoehn and Yahr (H&Y) stage [222], another standard for PD severity estimation. The H&Y stage uses a categorical scale, where a higher stage indicates worse severity. Again, we use the Udall and the MJFF datasets since they report the H&Y scores and have multiple nights per subject. Fig. 6-4b shows that even though the model is not trained using H&Y, it can reliably differentiate patients in terms of their H&Y stages (p = 5.6e-08, Kruskal-Wallis test).

Finally, we compute the test-retest reliability of PD severity prediction on the same datasets in Fig. 6-4c. Our model provides consistent and reliable predictions for assessing PD severity with its reliability reaching 0.97 (95% CI [0.95, 0.98]) with 12 nights per subject.

■ 6.3.4 PD Risk Assessment

Since breathing and sleep are impacted early in the development of PD [216, 10, 218], we anticipate that our AI model can potentially recognize individuals with PD before their

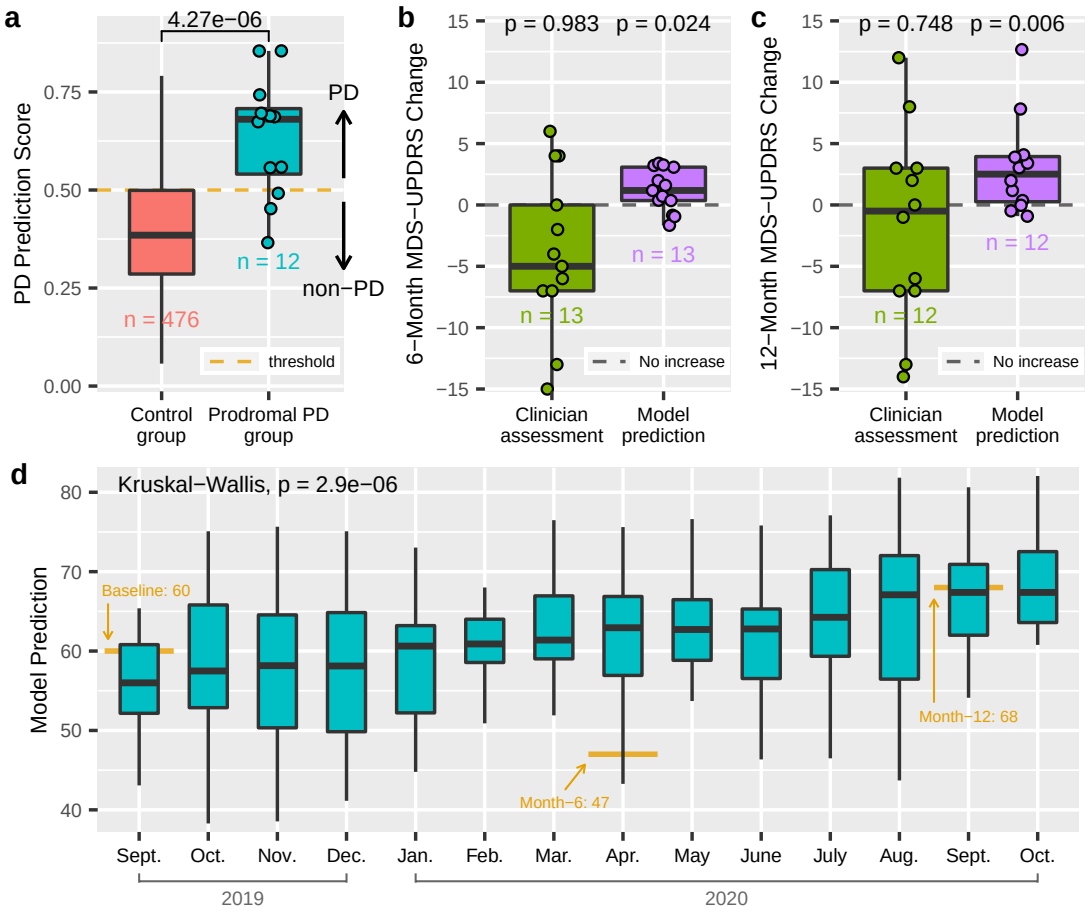


Figure 6-5: **Model evaluation for PD risk assessment prior to actual diagnosis, and disease progression tracking using longitudinal data. a,** The model prediction scores for the prodromal PD group (i.e., undiagnosed individuals that eventually were diagnosed with PD) and the age- and gender-matched control group (p = 4.27e-06, one-tailed Wilcoxon rank-sum test). **b, c,** The AI model assessment of the change in MDS-UPDRS over 6-month and 12-month periods (p = 0.024 for 6 months, p = 0.006 for 12 months, one-tailed one-sample Wilcoxon signed-rank test) and the clinician assessment of the change in MDS-UPDRS over the same periods (p =0.983 for 6 months, p = 0.748 for 12 months, one-tailed one-sample Wilcoxon signed-rank test). **d,** Continuous severity prediction across one year for the patient with maximum MDS-UPDRS increase (p = 2.9e-06, Kruskal-Wallis test; n=365 nights from 09/01/2019 to 10/31/2020). For each box in all sub-figures, the central line indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to 1.5 times the interquartile range.

actual diagnosis. To evaluate this capability, we leverage the MrOS dataset [221], which includes breathing and PD diagnosis from two different visits, separated by approximately 6 years. We consider subjects who were diagnosed with Parkinson's by their second visit, but had no such diagnosis by their first visit, and refer to them as the "prodromal PD group" (n=12). To select the "control group", we sample subjects from the MrOS dataset

who did not have a PD diagnosis in the first visit nor in the second visit, occurring six years later. For each of the subject in the prodromal group, we sample up to 40 control subjects that are age and gender matched, resulting in 476 qualified control subjects. We evaluate our model on breathing data from the first visit when neither the prodromal group nor the control group had a PD diagnosis. Fig. 6-5a shows that the model gives the prodromal group (i.e., subjects eventually diagnosed with PD) much higher PD scores than the control group (p = 4.27e-06, one-tailed Wilcoxon rank-sum test). Indeed, the model predicts 75% of them as individuals with PD prior to their reported PD diagnosis.

6.3.5 PD Disease Progression

Today, assessment of PD progression relies on MDS-UPDRS, which is semi-subjective and does not have enough sensitivity to capture small, progressive changes in patient status [202, 203]. As a result, PD clinical trials need to last for multiple years before changes in MDS-UPDRS can be reported with sufficient statistical confidence [202, 205], which creates a great challenge for drug development. A progression marker that captures statistically significant changes in disease status over short intervals can shorten PD clinical trials.

We evaluate disease progression tracking on data from the Udall study, which includes longitudinal data from participants with PD 6 months (n=13) and 12 months (n=12) into the study. For those individuals, we assess their disease progression using two methods. In the first method, we use the difference in the clinician-scored MDS-UPDRS at baseline and at month 6, or month 12. In the second method, we use the change in their predicted MDS-UPDRS over 6 months, or 12 months. To compute the change in the predicted MDS-UPDRS, we take the data from the one month following the baseline and compute its median MDS-UPDRS prediction, and take the month following the month-6 visit and compute its median MDS-UPDRS prediction. We then subtract the median at month 6 from the median at baseline. We repeat the same procedure for computing the prediction difference between month 12 and baseline. We plot the results in Fig. 6-5b and 6-5c. The results show both the 6-month and one-year changes in MDS-UPDRS as scored by a clinician are not statistically significant (6-month p = 0.983, 12-month p = 0.748, one-tailed one-sample Wilcoxon signed-rank test), which is consistent with prior observations [202, 203, 205]. In contrast, the model's estimates of changes in MDS-UPDRS over the same periods are statistically significant (6-month p = 0.024, 12-month p = 0.006, one-tailed one-sample Wilcoxon

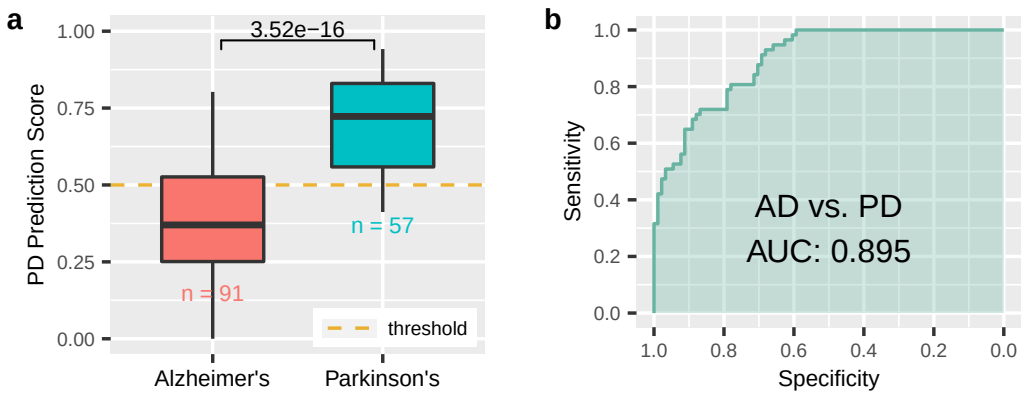


Figure 6-6: **Performance of the AI model on differentiating subjects with Parkinson's disease (PD) from subjects with Alzheimer's disease (AD). a,** The model's output scores differentiate PD subjects from AD subjects (p = 3.52e-16, one-tailed Wilcoxon rank-sum test). **b,** Receiver operating characteristic (ROC) curves for detecting PD subjects against AD subjects (n=148). The model achieves high AUC for differentiating PD from AD (AUC = 0.895).

signed-rank test).

To provide more insight, we examine continuous severity tracking over one year for the patient in our cohort who exhibited the maximum increase in MDS-UPDRS over this period (Fig. 6-5d). The results show that the AI model can achieve statistical significance in tracking disease progression in this patient from one month to the next (p = 2.9e-06, Kruskal-Wallis test). The figure also shows that the clinician-scored MDS-UPDRS is noisy; the MDS-UPDRS at month-6 is lower than at baseline, though PD is a progressive disease and the severity should be monotonically increasing.

Finally, we note that the above results persist if one controls for changes in symptomatic therapy. Specifically, we repeated the above analysis while limiting it to patients who had no change in symptomatic therapy. The changes in the model-predicted MDS-UPDRS are statistically significant (6-month p = 0.049, 12-month p = 0.032, one-tailed one-sample Wilcoxon signed-rank test), whereas the changes in the clinician-scored MDS-UPDRS are statistically insignificant (6-month p = 0.894, 12-month p = 0.819, one-tailed one-sample Wilcoxon signed-rank test).

■ 6.3.6 Distinguish Parkinson's Disease from Alzheimer's Disease

We additionally test the model's ability to distinguish between PD and Alzheimer's disease (AD), the two most common neurodegenerative diseases. To evaluate this capability,

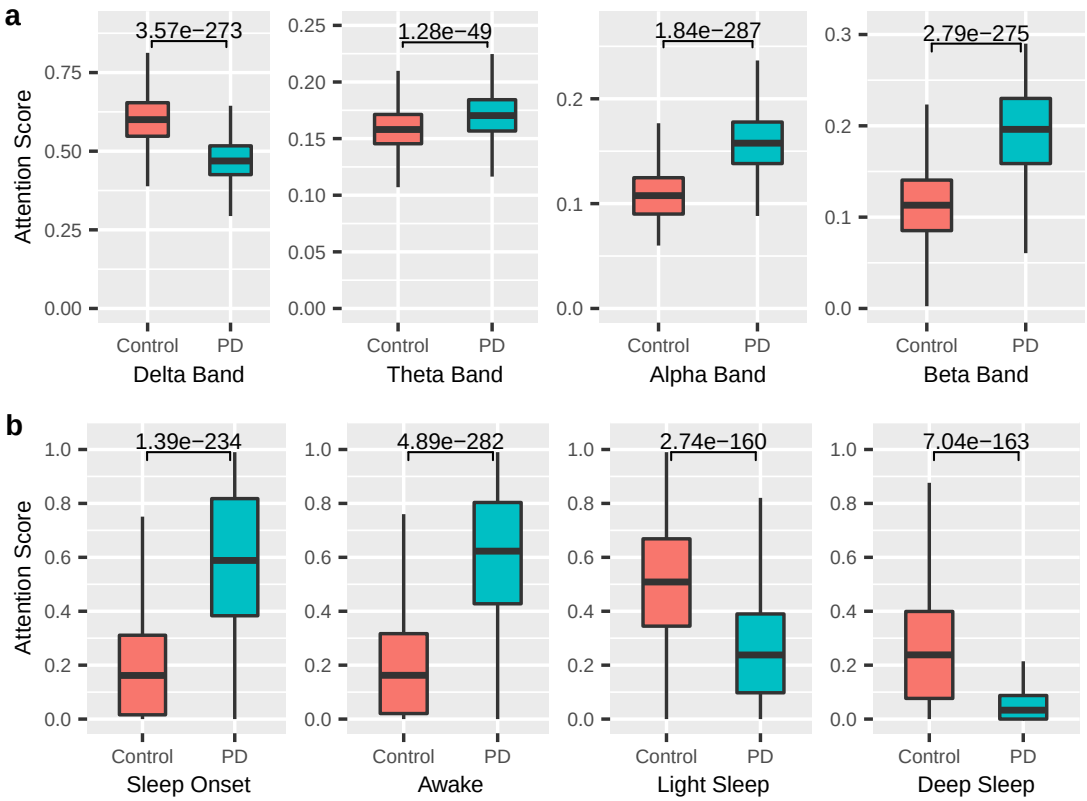


Figure 6-7: **Interpretation of the output of the AI model with respect to EEG and sleep status. a, b,** Attention scores were aggregated according to sleep status and EEG bands for PD patients (n=736 nights from 732 subjects) and controls (n=7,844 nights from 6,840 subjects). Attention scores were normalized across EEG bands or sleep status. **a,** Attention scores for different EEG bands between PD patients and control individuals (Delta band p = 3.57e-273, Theta band p = 1.28e-49, Alpha band p = 1.84e-287, Beta band p = 2.79e-275, one-tailed Wilcoxon rank-sum test). **b,** Attention scores for different sleep status between PD patients and control individuals (sleep onset p = 1.39e-234, awake period p = 4.89e-282, light sleep p = 2.74e-160, deep sleep p = 7.04e-163, one-tailed Wilcoxon rank-sum test). On each box in all sub-figures, the central line indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to 1.5 times the interquartile range.

we leverage the SHHS [30] and MrOS [221] datasets, which contain subjects identified with AD (details in Sec. 6.2). In total, 99 subjects are identified with AD, and 9 of them also reported to have PD. We exclude subjects with both AD and PD, and evaluate our model's ability to distinguish the PD group (n=57) from the AD group (n=91). Fig. 6-6 shows that the model achieves an AUC of 0.895 with a sensitivity of 80.70% and specificity of 78.02% in differentiating PD from AD, and reliably distinguish PD from AD subjects (p = 3.52e-16, one-tailed Wilcoxon rank-sum test).

6.3.7 Model Interpretability

Our AI model employs a self-attention module [64], which scores each interval of data according to its contribution to making a PD or Non-PD prediction. Since the SHHS and MrOS datasets include EEG signals and sleep stages throughout the night, we can analyze the breathing periods with high attention scores, and the corresponding sleep stages and EEG bands. Such analysis allows for interpreting and explaining the results of the model.

The analysis shows that the model's attention focuses on periods with relatively high qEEG Delta activity for control individuals, while focusing on periods with high activities in Beta and other bands for PD patients (Fig. 6-7a). Interestingly, these differences are aligned with prior work which observed that PD patients have reduced power in Delta band and increased power in Beta and other EEG bands during non-REM sleep [235, 14]. Further, comparing the model's attention to the person's sleep stages shows that the model recognizes control subjects by focusing on their light/deep sleep periods, while attending more to sleep onset and awakenings in PD patients (Fig. 6-7b). This is consistent with the medical literature which reports that PD patients have significantly less light and deep sleep, and more interruptions and wakeups during sleep [235, 236], and the EEG in PD patients during sleep onset and awake periods show abnormalities in comparisons with non-PD individuals [224, 225, 228].

6.4 Discussion

This chapter provides evidence that AI can identify people who have Parkinson's disease from their nocturnal breathing and accurately assess their disease severity and progression. Importantly, we were able to validate our findings in an independent external PD cohort. The results show the potential of a new digital biomarker for PD. This biomarker has multiple desirable properties. It operates both as a diagnostic and progression biomarker. It is objective and does not suffer from the subjectivity of either patient or clinician. It is noninvasive and easy to measure in the person's own home. Further, by using wireless signals to monitor breathing, the measurements can be collected every night in a touchless manner.

Our results have multiple implications. First, our approach has the potential of reducing the cost and duration of PD clinical trials, and hence facilitating drug development.

The average cost and time of PD drug development are approximately $1.3 billion and 13 years, which limits the interest of many pharmaceutical companies in pursuing new therapies for PD [206]. PD is a slowly progressing disease, and the current methods for tracking disease progression are insensitive and cannot capture small changes [202, 203, 205, 206]. Hence, they require several years to detect progression [202, 203, 205, 206]. In contrast, our AI-based biomarker has shown potential evidence of increased sensitivity to progressive changes in PD. This can help shorten clinical trials, reduce cost, and speed up progress. Our approach can also improve patient recruitment and reduce churn because the measurements can be collected at home with no overhead to patients.

Second, today, about 40% of individuals with PD do not receive care from a PD specialist [237]. This is because PD specialists are concentrated in medical centers in urban areas, while patients are spread geographically, and have problems traveling to such centers due to old age, and limited mobility. By providing an easy and passive approach for assessing disease severity at home and tracking changes in patient status, our system can reduce the need for clinic visits and help extend care to patients in underserved communities.

Third, our system could also help in early detection of PD. Today's diagnosis of PD is based on the presence of clinical motor symptoms [199], which are estimated to develop after 50-80% of dopaminergic neurons have already degenerated [238]. Our system shows initial evidence that it could potentially provide risk assessment prior to clinical motor symptom.

We envision that the system could eventually be deployed in the homes of PD patients and individuals at high risk for PD (e.g., those with LRRK2 gene mutation) to passively monitor their status and provide feedback to their provider. If the model detects severity escalation in PD patients, or conversion to PD in high-risk individuals, the clinician could follow up with the patient to confirm the results either via telehealth or a visit to the clinic. Future research is required to establish the feasibility of such use pattern, and the potential impact on clinical practice.

Our study also has some limitations. PD is a non-homogeneous disease with many subtypes [239]. We did not explore subtypes of PD and whether our system works equally well with all subtypes. Another limitation of the chapter is that both the progression analysis and preclinical diagnosis are validated in a small number of participants. Future studies with larger populations are required to further confirm those results. Also, while we have

confirmed that our system could separate PD from AD, we did not investigate the ability of our model to separate PD from broader neurological diseases. Further, while we have tested the model across institutions and using independent datasets, further studies can expand the diversity of datasets and institutions. Additionally, our empirical results highlight a strong connection between PD and breathing and confirm past work on the topic; however, the mechanisms that lead to the development and progression of respiratory symptoms in PD are only partially understood and require more studies.

Finally, our work shows that advances in AI can support medicine by addressing important unsolved challenges in neuroscience research and allowing for the development of novel biomarkers. While the medical literature has reported several PD respiratory symptoms, such as weakness of respiratory muscles [213], sleep breathing disorders [214, 215, 216, 217], and degeneration in the brain areas that control breathing [212], without our AI-based model, no physician today can detect PD or assess its severity from breathing. This shows that AI can provide new clinical insights that otherwise may be inaccessible.

CHAPTER 7

In-Home Monitoring of Sleep Posture with Wireless Signals

Each of us has our favorite sleep postures: sleeping on the right side, left side, facing up, or facing down. Significant clinical research has shown that sleep posture is a valuable marker of disease progression, and has a significant impact on health. For instance, patients with Parkinson's disease often suffer from loss of axial movement; and less frequent nocturnal turnovers and longer periods spent recumbent or supine (*i.e.*, facing up) are associated with deterioration in the condition of Parkinson's patients [240]. Similarly, infrequent changes in sleep posture can lead to pressure ulcers in the elderly and post-surgery patients [241]. Studies have also demonstrated that sleeping in a supine position can reduce back pain since it is the position in which the muscles have the least amount of work to do to maintain one's posture against the force of gravity [242]. In contrast, if one has obstructive sleep apnea (OSA), the supine position becomes the worst posture because it imposes unfavorable airway geometry and reduces lung volume. Studies have shown that more than half of all OSA cases can be classified as supine related [243, 244]. Improper sleep posture can even be fatal – sleeping on the stomach can boost the risk of sudden infant death syndrome (SIDS) [245] and sudden death in epilepsy patients [246, 11]. These examples highlight the importance of continuous and fully automatic sleep posture monitoring. Such monitoring can provide doctors with information to better manage patient conditions; it can also provide people themselves information to adjust their posture and

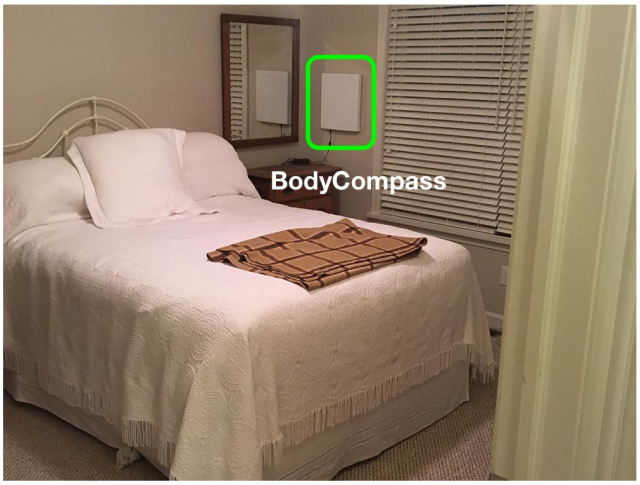


Figure 7-1: **BodyCompass in one of our deployments.** The white box mounted on the wall is the radio. It uploads the RF signals to the cloud where the model processes them to extract sleep posture.

reduce their health risks.

Unfortunately, today, there is no good way to provide such sleep posture monitoring. Doctors typically resort to asking patients about their sleep posture, an error-prone mechanism since people routinely, and unknowingly, change their postures while sleeping. Automated monitoring systems primarily fall into two categories. The first category is vision-based. These methods use a camera to monitor the user's sleep, then extract postures from recorded videos with a machine learning system. Deploying cameras in people's bedrooms, however, is privacy-intrusive. Furthermore, cameras have difficulties tracking body posture if the person is covered or lighting is bad, both of which are typical scenarios when sleeping. The second category uses various kinds of on-bed sensors. Such methods require the user to fix the sensor to the surface of the mattress, which can affect sleep comfort.

Ideally, one desires a system that is non-contact, non-intrusive, and works even in dark scenarios typical of sleeping conditions. In this chapter, we present *BodyCompass*, an RF-based system for sleep posture monitoring. BodyCompass analyzes the reflections of RF signals to infer subjects' sleep postures. It does so without requiring users to wear or be in contact with any sensors. It is not invasive of privacy, and can also work in the dark. Unlike much previous work, BodyCompass has also been demonstrated to work in the wild - with real subjects sleeping in their own homes, and can generalize to new environments with minimal additional training. Fig. 7-1 shows BodyCompass deployed in the home of one of

our users.

But how can one extract the sleep posture from radio signals? Our idea is to use the multipath effect, a known phenomenon in RF communication systems that refers to the fact that RF signals bounce off different objects and obstacles in the environment and reach the receiver through multiple paths. Past work has shown that the human body acts as a reflector in the low-GHz frequencies, commonly used in commodity radios [247]. As the RF signal is incident upon the human body, it reflects from the body based on the body orientation and bounces off the surrounding objects and walls creating a multipath signature indicative of the body posture. Our objective is to learn an inverse map that receives the reflected multipath profile and attempts to infer the body posture. A key challenge in delivering this idea is that the RF signal bounces off many objects in the environment, not just the human body. Only a subset of the signal path involves reflections from the human body, and hence is relevant to the sleep posture. Thus, one has to extract only the RF reflections that bounced off the human body either directly or indirectly in order to determine the sleep posture.

To address this challenge, we leverage past work that shows how to extract a person's breathing from RF signals. Our intuition is that all paths that bounce off a person's trunk (e.g. chest and belly) during their sleep are modulated by the person's breathing, and hence we can use this property to disentangle these reflections from the rest of the reflections. Specifically, we use standard techniques to separate signals along different paths (FMCW and angle of arrival [248]), and correlate these separated signals individually with the subject's breathing signal to identify the specific signals corresponding to the person in bed. We further design a neural network model that takes this breathing filtered multipath profile, and predicts the sleep posture of the person.

A key question with such a system is how well the neural network model works with different people and in different homes. While RF reflections and the multipath effect naturally depend on the environment, one would hope that with proper design, the model would be able to transfer some of the knowledge across environments. Such a model would learn the underlying features that identify each sleep posture, and tune them to a new environment with a small amount of additional labeled data from that environment. To address this issue, we design our model to be easily transferable. Specifically, given a set of source domains *i.e.,* a number of people and their sleep postures in the training set,

and a target domain *i.e.*, a new person in his own home, the model can use a small amount of labeled data (16 minutes to one night) from the new home to optimize its performance for this new environment.

Our model delivers high accuracy. Specifically, our basic sleep posture model using multipath, when trained and tested on the same person and home, achieves an accuracy of 94.1%. The transfer learning model to a new person and a new home has an accuracy of 86.7% with one night of labeled data, and 83.7% with a labeled dataset comprising 8 examples, where in each example, the person lies down in one of his typical sleep postures for a duration of 2 minutes.

To summarize, this chapter makes the following contributions: (i) We present BodyCompass, the first RF-based system that provides accurate sleep posture monitoring in users' own homes. It achieves high accuracy without sacrificing privacy and sleep comfort. (ii) BodyCompass can transfer its model to new homes and users with very little additional training data. (iii) We implement and evaluate BodyCompass extensively in real world settings using data from 26 homes with 26 different subjects and more than 200 nights of sleep.

7.1 Related Work

Past work on sleep posture monitoring can be divided into two major categories: 1) systems with on-body sensors, and 2) non-contact monitoring systems.

(a) On-body Solutions: On-body sensors can monitor sleep postures accurately [249, 250, 251]. For example, one may attach an accelerometer to the person's chest to monitor their sleep posture. Since gravity always points downwards, the accelerometer's orientation can be calculated by combining the acceleration along three different axes [250, 251]. However this method is cumbersome and uncomfortable since the accelerometer needs to be fixed on the user's body during their sleep.

(b) Non-Contact Solutions: Contactless systems are more comfortable for the user compared to on-body sensors. Work in this class falls in the following categories. First, vision-based systems [253, 254, 255] deploy RGB or infra-red cameras to record videos of the user's sleep, then process those videos using convolutional neural networks to predict sleep postures. However, cameras, particularly in people's bedrooms, are privacy-intrusive.

Figure 7-2: **Pressure sensitive bedsheet from [252].**

Further, the accuracy of camera systems decreases significantly in dark settings and when people are covered with a blanket or comforter [253, 254, 255].

Second, on-bed sensors cover the mattress with an array of pressure sensors [256, 252, 257, 258, 259] or RFID Tags [260, 261]. These solutions are more privacy-preserving than camera systems. However, on-bed sensors, shown in Fig. 7-2, change the feel of the bed and thus affect the sleep comfort of the subject. Further, most of these systems are evaluated in the lab, as opposed to overnight testing in people's own homes [256, 257, 258, 259, 260, 261].

Third, a few papers have proposed the use of RF signals for monitoring sleep posture [262, 263, 264]. The approach in those papers is intrinsically different from ours; they analyze the signal power as measured by the RSSI (received signal strength indicator) [263] or the power of the frequency sub-channels extracted from the CSI (channel state information) [262, 264]. As studied in [265], they all inherently suffer from interference. That is, they have no ability to separate changes in the signal that are due to the sleeping person from those due to other sources of motion (e.g. a fan, or a person moving in a neighboring room). Such extraneous motion brings randomness and will greatly hamper the robustness of the system in the wild. As a result, all previous papers are evaluated in a single lab environment with one or two subjects consciously performing specified postures.[1] In contrast, we study the spatial pattern of reflections –i.e., the multipath – and ignore the power by re-normalizing the power distribution of each path (see Section 7.3). Therefore, our system can provide accurate sleep posture monitoring overnight in users' homes and can be easily transferred to new environments.

[1]We note that while the authors of [264] test their vital sign algorithms outside the lab, the sleep posture is only tested in the lab and in one setting.

We also note that past work has demonstrated the feasibility of inferring the human skeleton using only RF reflections [247, 266]. It might seem that one could use such models to infer the skeleton of the person lying in bed and hence their sleep posture. However, due to RF specularity, such models rely on people walking and moving around to achieve good accuracy [247, 266]. Specifically, as described in those papers, a snapshot of RF signal reflections does not capture the full body; Any snapshot captures only a few limbs or body parts that reflect signals directly towards the radio. Hence, their neural networks rely on people moving and walking to expose different body parts in each snapshot so that the network can combine those body parts to create the human skeleton. In contrast, when the person is asleep in bed, the person is mostly static and hence there is not enough motion to allow the neural network to fill in the gaps and combine body parts across different snapshots. To deal with this challenge, our system not only takes the direct reflections towards the radio, but also all the indirect reflections due to multipath. By taking all the multi-path reflections as input, our system estimates the sleep posture accurately even when the person remains static.

Finally, this chapter belongs to a growing body of research that focuses on passive monitoring using radio signals. Researchers have demonstrated that by carefully analyzing RF reflections off the human body, they can monitor people's location [248, 267, 268], gait [269, 270, 16], breathing [271, 220, 272], heart rate [219, 273, 274], falls [275, 276, 277], and sleep quality and stages [230, 265, 278, 279]. Our work builds on this foundation and leverages past work on inferring the breathing signal as a sub-component in our system [220].

■ 7.2 BodyCompass

BodyCompass is the first RF-based system that provides accurate sleep posture monitoring in the wild, *i.e.*, with subjects sleeping in their own beds in their homes, and it generalizes to new subjects and homes with minimal additional effort. It can be used by healthy individuals interested in monitoring their sleep behavior, or can be provided either to patients to help them modify their sleep posture, or to doctors to assist them in understanding disease prognosis and patient health.

BodyCompass leverages measurements from an FMCW radio equipped with an an-

tenna array [280]. Such radios are commonly used in passive health monitoring using RF signals [220, 278, 265]. They work by transmitting a low power radio signal, and observing its reflections from the surrounding environment. The use of an antenna array combined with FMCW enables the radio to resolve RF reflections from multiple points in space. Specifically, at each instance in time, the radio outputs an array of signal values from various voxels in space, which we refer to as an *RF-snapshot*.

BodyCompass takes a sequence of RF-snapshots from an FMCW radio across a whole night, and produces the sleep postures for the night. A sleep posture is described by an angle between two normal vectors, one of the bed surface and one of the user's anterior trunk surface, as shown in Fig. 7-3(a). For example, 0° represents the user facing upwards and 90° represents the user facing rightwards. Defining sleep posture in terms of angle allows us to differentiate between a slight tilt of the trunk to the right and someone sleeping on their right side. This enables a finer granularity definition of sleep postures that encompasses and expands beyond common posture classes (supine, left side, right side, prone). A fine granularity in posture estimation is important for applications that aim to detect changes in postures, such as tracking the progression of Parkinson's patients by monitoring the frequency of their change of sleep posture.

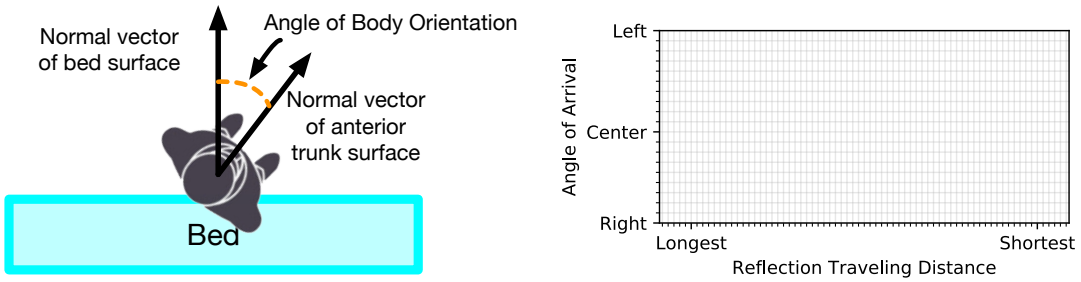
BodyCompass computes sleep postures using three components:

- A filtered multipath profile feature extractor to estimate the RF reflections that bounced off the person directly or indirectly.
- A source-specific neural network that utilizes the multipath profile features to estimate the sleep posture of a specific person in a specific home.
- A transfer learning model that adapts the source-specific models to estimate the sleep posture of a new person in a new home with minimal additional labeled data.

Below, we describe these components in detail.

7.3 Filtered Multipath Feature Extractor

In this section, we describe how BodyCompass extracts filtered multipath features specific to a person from RF-snapshots produced by a FMCW antenna array.



(a) Angle of body orientation.

(b) Coordinates of RF-snapshots.

Figure 7-3: **Illustration of body orientation and coordinates of RF-snapshots.** In Fig. 7-3(b), we represent all RF voxels in a Cartesian coordinate system. The rightmost pixels are closest to the device, and have the shortest distance. Therefore the direct-path reflections should be to the right of the indirect-path reflections since they travel the shortest path between the user and the device.

An RF-snapshot consists, for each point of space (RF voxel), of the magnitude of the RF reflection from that point of space. An RF voxel is represented by two coordinates, its distance from the device, and the angle of that position relative to the normal from the device. Specifically, an RF voxel at coordinate (i, j) represents a small cube around the point at traveling distance d_j from the device, and an angle of arrival of α_i, as shown in Fig. 7-4. We divide the space into N angles, and M distances, and therefore each RF-snapshot is an $N \times M$ matrix. For better visualization, we plot voxels in a standard Cartesian coordinate system, as shown in Fig. 7-3(b), instead of a polar coordinate system.

■ 7.3.1 Stable Sleep Periods

We first note that sleep postures are not independent over time, since people typically sleep in a posture for some period of time, followed by a movement, after which they settle into a different sleep posture, and so on. BodyCompass therefore first segments the night into a series of stable sleep periods. During each stable period, the orientation of the body is approximately constant, and BodyCompass extracts a single sleep posture from that period. BodyCompass leverages prior work [220] to identify motion events from RF-snapshots, and defines the intervals between such motion events as stable sleep periods.

■ 7.3.2 Filtered Multipath profile

Next, BodyCompass extracts the multipath profile for each stable sleep period, with the objective of learning the sleep posture from the multipath profile. Recall that the multipath

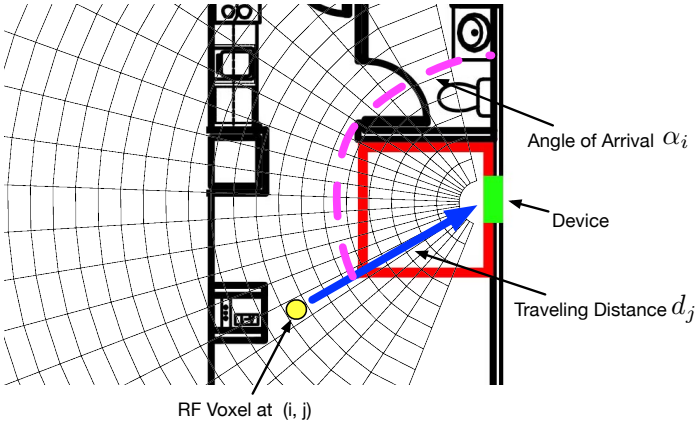


Figure 7-4: **Illustrative example of an RF voxel.** The green rectangle represents the radio location and the red rectangle represents the bed location. Our FMCW antenna array divides the space into small grids. The coordinates of the grid represent the distance from the radio and the angle of arrival.

profile captures the pattern of spatial reflections, *i.e.*, how the RF signals bounce around in space before they reach our radio. We represent the multipath profile of a particular stable period with the relative signal power along each path. Thus, the multipath profile of a particular stable period can be computed by taking the RF-snapshots corresponding to the stable sleep period and computing the variance in each voxel.

The multipath profile is affected by the sleep posture of the person, and is therefore informative about their orientation. For instance, when the person is supine, *i.e.*, lying flat on their back, a significant portion of the signal reflects towards the ceiling and bounces off other objects in the environment before reflecting back to the radio, and as a result the multipath profile shows significant dispersion. In contrast, when the person is sleeping on their side, the direct RF reflections will be significantly stronger than the indirect reflections, and the multipath profile will therefore show high concentration.

However, one cannot directly use the overall multipath profile in a stable sleep period to infer sleep posture. This is because such multipath profile contains reflections both from the environment and from the subject. While reflections from static reflectors (*e.g.*, walls, tables) can be removed,[2] reflections from moving objects cannot be easily disentangled, and their contributions can confound the system for two reasons. First, even within a single home, those contributions can change over time even when the sleep posture does not change, for instance, because of movements from a fan, people walking in the environ-

[2]We remove static reflections by subtracting the average RF-snapshot for each stable period before computing the multipath profile [220].

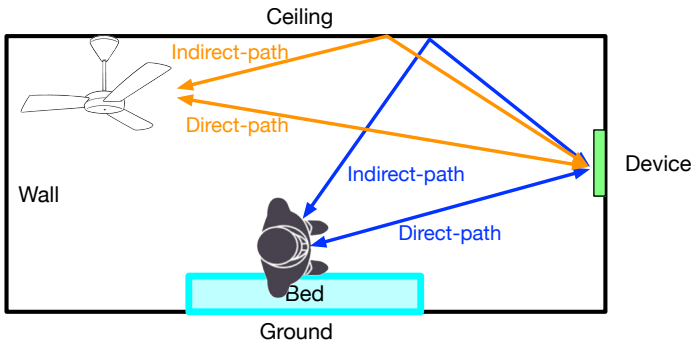
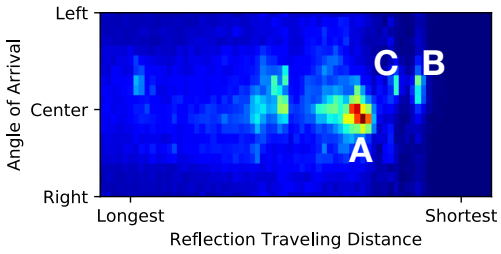


Figure 7-5: **An illustrative example of signal reflections.** In this case the multipath profile contains reflections from the subject (in blue) and a ceiling fan (in orange). Correct processing requires eliminating the fan reflections. Also the figure illustrates how the posture could affect the multipath. When the user is facing the device, the reflections along the direct path have the largest signal variations because the chest movements are most significant in that direction. In contrast, the signal variations along the indirect path are much smaller because the side of the body is not moving significantly.

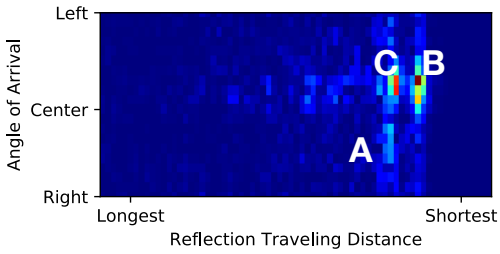
ment, or heating, ventilation, and air conditioning (HVAC) systems. Since these changes are not correlated with the sleep posture of the person, they will adversely affect the ability of BodyCompass to infer sleep posture. Furthermore, such reflections are highly specific to each home, and incorporating them into the multipath profile will prevent BodyCompass from generalizing to new homes.

So, how does one filter out environmental contributions while still retaining the multipath contributions from the sleeping subject? Our idea is inspired by the following observation: when breathing, the chest and belly area of the human body move forward and backward. These motions will change the multipath contributions corresponding to the human body in a manner correlated with the breathing signal, while other environment related multipath contributions will not change in a manner correlated with the person's breathing. Fig. 7-5 shows an illustrative example of this point with a person sleeping facing the device, and a nearby fan.

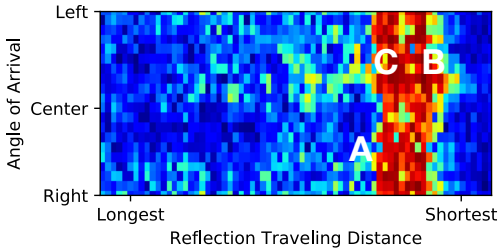
Using breathing also allows BodyCompass to identify the orientation of the person, *i.e.*, whether the person is facing up/down when supine, and whether the person is facing towards/away from the device when on their side. This is because during breathing, only the front of the human body moves significantly, whereas the back does not, therefore breaking symmetry between the orientations, and changing the *filtered* multipath profiles in the two cases.



(a) Original multipath profile.



(b) Filtered multipath profile.

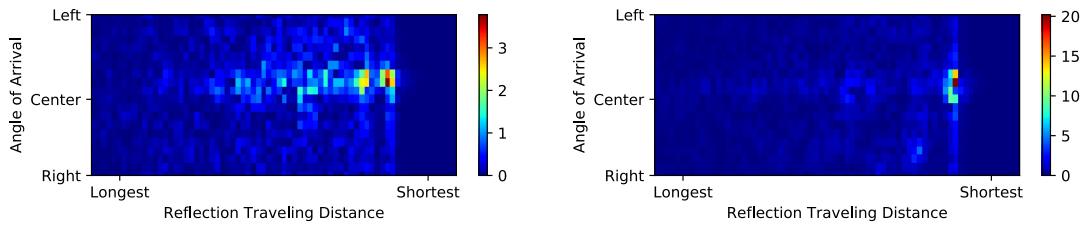


(c) Filter used to extract the multipath associated with the subject. The filter consists of the correlation coefficients with breathing.

Figure 7-6: **Visualization of one stable period.** The value of each RF-voxel represents the corresponding attribute of the RF reflection. The visualization is color-coded, the redder the pixel, the higher the relative value of that pixel, and the bluer the pixel, the lower the relative value. Points A, B, C highlight three different kinds of reflections: environmental movement reflections, breathing reflections along direct-path, breathing reflections along indirect-path.

BodyCompass uses DeepBreath [220], to extract the breathing signal of the subject in bed from the RF-snapshots in the stable period. The breathing signal is a time series that reflects the scaled chest displacement of the subject over time. Then, for each RF-voxel, BodyCompass correlates this extracted breathing signal with the time series of signal magnitudes for this RF-voxel obtained from the RF-snapshots. Specifically, for each voxel, BodyCompass computes the absolute value of the Pearson correlation coefficient between the person's breathing and the magnitude of the RF signal received from that voxel, as expressed in the sequence of RF-snapshots. This correlation provides a spatial filter that allows us to extract the voxels in the multipath profile whose signal is highly correlated with the person's breathing.

Next, BodyCompass multiplies the multipath profile with the above filter to extract the filtered multipath profile, which focuses on the signals that bounced directly or indirectly off the subject. The filtered multipath profile emphasizes pixels with a significant contribution from the subject's breathing while still retaining the relative power contributions



(a) The user facing up.

(b) The user facing the device.

Figure 7-7: **Two typical examples of filtered multipath profiles of the user facing up (Fig. 7-7(a)) and facing towards the device (Fig. 7-7(b)).** Compared to Fig. 7-7(b), in Fig. 7-7(a), we can see much higher power in further away pixels. This is because when the user is facing up, he deflects the signal towards the ceiling causing indirect reflections.

from direct and indirect paths corresponding to that breathing.

To help understand the process, we plot an example in Fig. 7-6. Specifically, Fig. 7-6(a) shows the relative power in each voxel in space in the original multipath profile. Fig. 7-6(c) shows the breathing filter, and Fig. 7-6(b) shows the filtered multipath profile. As we can see, pixel A is very bright in the original multipath profile (Fig. 7-6(a)), meaning that it has very a high reflection power relative to other pixels. However, since it has a very low correlation coefficient with breathing (Fig. 7-6(c)), it is removed in the final filtered multipath profile (Fig. 7-6(b)). In this case, pixel A was contributed by environmental movements from a different person walking in the environment. In contrast, while pixels B and C have lower power compared to A, they are emphasized in the filtered multipath profile since they exhibit strong correlation with the breathing signal. It is also worth noting that pixel B is actually the direct-path reflection, and pixel C is one of the indirect-path reflections.

Next we show two typical filtered multipath profiles of two different postures in Fig. 7-7. When the user is sleeping in a supine position, the filtered multipath profile shows more dispersion because the subject reflects a significant part of the signal towards the ceiling causing more indirect reflections. In contrast, when the user is facing the device most of the signal is directly reflected from the user to the device and hence the power in the filtered multipath profile is concentrated at the user's location, i.e. the direct path.

7.4 Source-Specific Sleep Posture Model

Having computed the filtered multipath profile for each stable sleep period corresponding to a source (*i.e.* a user in a specific home), BodyCompass then uses a neural network to predict the sleep posture for that source during each stable sleep period.

Our model uses a multi-layer fully-connected neural network. We deliberately choose a fully-connected neural network instead of the commonly used convolutional neural network (CNN). CNNs are more suitable for natural images because one typically needs to compare each pixel with the pixels in its neighborhood. In contrast, to capture the multipath profile, one needs to compare pixels globally. (See Sec. 7.7.4 for an empirical comparison of the performance of a fully-connected network and a CNN on this task.)

For training the neural network, we ask subjects to wear accelerometers to collect the ground-truth angular orientation of the body. Detailed ground truth collection process is described in Sec. 7.6.2. Recall that we express the sleep position in terms of the angle specifying the trunk rotation with respect to the bed. BodyCompass averages the angular values that the accelerometer measures during a stable period to obtain the ground truth sleep posture of the subject in that period.

BodyCompass trains the neural network to predict the sleep angle associated with each filtered multipath profile. To train the network we need to compare the predicted angle with the ground truth angle. Directly comparing angles however leads to discontinuity since angles wrap around, i.e., 0° and 360° are the same angle, but simply computing their difference will yield a large loss.

Therefore, in order to ensure smoothness of the loss function, BodyCompass's model predicts complex numbers, and uses the phase of the complex number as the angle prediction. Specifically, we define *Circular Loss* as follows:

$$\mathcal{L}_c(\theta) = \mathbb{E}_{x,y\sim p(x,y)}\, arccos\left(\frac{Re(F(x,\theta)\cdot e^{-iy})}{|F(x,\theta)|}\right) \tag{7.1}$$

where x is the input feature vector (i.e., the filtered multipath profile of a stable segment), y is the ground-truth angle, $F(\cdot,\theta)$ is the model that maps a feature vector into a complex number, θ is the model parameters (the wights of the neural network), $\arccos$ denotes the arc cosine function, and $\mathbb{E}$ is the expectation.

The operand of $arccos$: $\frac{Re(F(x,\theta)\cdot e^{-iy})}{|F(x,\theta)|}$ can be interpreted as the cosine similarity between

two vectors: one is our prediction $(Re(F(\cdot)), Im(F(\cdot))$ and the other one represent the unit vector of the ground truth angle $(cos(y), sin(y))$. The similarity reaches its maximum when the predicted vector has the same angle as the ground truth (an $arccos$ of 0). And it reaches its minimum when these two vectors are diametrically opposed (an $arccos$ of π radians). This loss function solves the discontinuity problem since it computes the angle difference between our prediction and the ground truth in a differentiable way.

7.5 Transferring the Model to New Users

In the previous section, we explained how to train a model to predict a user's sleep posture accurately given abundant labeled data from the that user. However, data collection is a laborious and time-consuming task for both the user and the operator of the system. Ideally, we would like our system to perform well on new users with minimal effort.

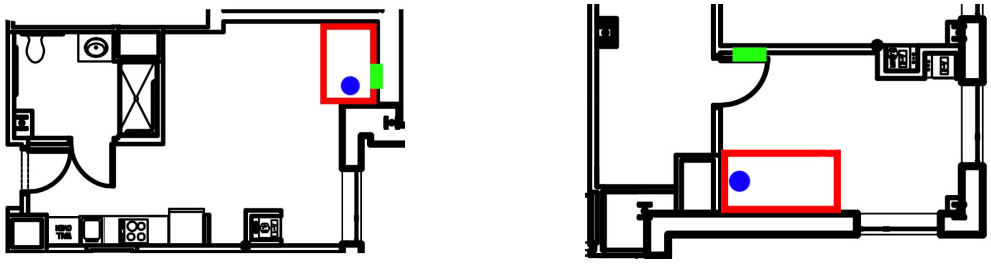
Since the properties of RF signals (power, phase, and multipath) depend on the environment, transfer between different homes is a challenging task. In order to achieve satisfactory performance while reducing the burden on the user, we assume that only limited labeled data from a new user is available. We refer to such labeled examples (where an example is a filtered multipath profile and its correct sleep angle) as *Calibration Points*. As described in Sec. 7.7.2, the number of calibration points can be as few as 8 examples, each lasting for 2 minutes, for a total of only 16 minutes.

Given the scarcity of the calibration points, it is not practical to train a model entirely based on those points. Instead, we formulate the task as a semi-supervised domain adaptation problem: we have multiple source users, each with abundant labeled data, and a target user for which we have a few calibration points. We would like the system to achieve high accuracy on the target user given the above information.

7.5.1 Overview of the Transfer Model

Given a set of source domains i.e. a number of people and their sleep postures in the training set, and a target domain i.e. a new person in their own home, we design a model that learns from the training data of the source domains how to infer sleep posture in the target domain, with a small number of calibration points.

At a high level, our transfer model first preprocesses the training data to ensure that



(a) User A's bedroom layout. (b) User B's bedroom layout.

Figure 7-8: **Bedroom layouts of two users.** Green rectangle shows the location of our device. Red rectangles shows the location of the bed and blue circle shows the position of the pillow.

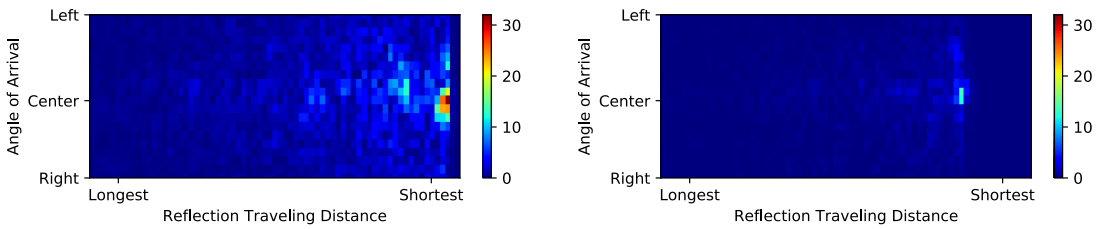
the probability distributions of the source domains look as close as possible to the target domain. Next, given that the amount of data from the target domain is not sufficient to train a model, we try to augment the data from the target by selecting data points from the source domains that look similar to the target data, both in terms of its feature map (*i.e.*, filtered multipath) and the corresponding posture. We use this augmented data to create a virtual target which is similar to the original target but has much more labeled data. Now we can adapt the model from each source domain to work well on the virtual target. The final prediction is then performed by majority voting over all of these adapted models.

In the following sub-sections, we expand on this high-level description, providing the details of the three key components of our transfer model:

- **Distribution Alignment**: We explicitly align the data distributions across users, whose differences are caused by different room layouts.
- **Data Augmentation**: We generate augmented data by picking data points from source subjects that resemble the calibration points.
- **Ensemble Learning**: We use Majority Voting to generate one final prediction that is robust and accurate.

■ 7.5.2 Distribution Alignment

Without any alignments, the model's generalization ability will be greatly hampered by the distribution shift between the source user and the target user. One major reason of this distribution shift is caused by different room layouts. For example, we have User A and



(a) User A's profile when facing to the device.

(b) User B's profile when facing to the device.

Figure 7-9: **Examples showing how the bed position with respect to the radio affects the signal's strength and location.** The figures show that due to differences in the position of the bed with respect to the radio, User A's direct path signal is much stronger and closer to the radio compared to User B's.

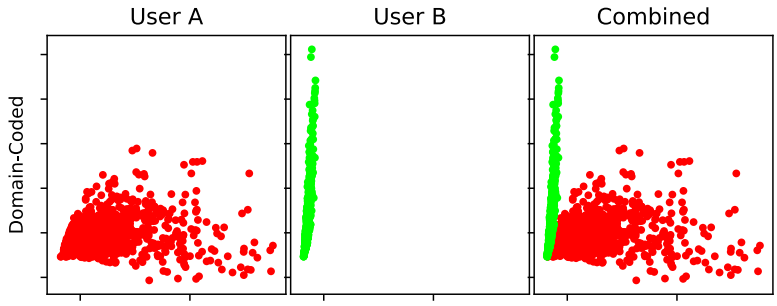
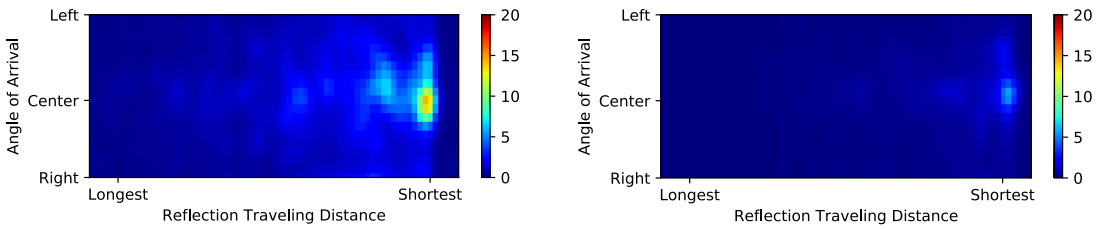


Figure 7-10: **Visualization of data distribution of User A (red) and User B (green).** Since the feature map (i.e., the filtered multipath profile) is high-dimensional, to visualize the data in a two-dimensional space we perform joint Principal Component Analysis (PCA) on all the feature vectors (all filtered multipath profiles) from both A and B using the same set of basis, and plot the data with respect to the two largest principle components. We plot the data of the source and target separately in the left two figures and combined in the right figure. As we can see, the distributions of two users are mismatched significantly.

User B with their floor plan visualized in Fig. 7-8. User A's bed is very close to the device (the device is right above the bed), and in comparison, User B's bed is far from the device. We show the multipath profiles of the two users in in Fig. 7-9 (a) and (b). As clear from these figures, the difference in the bed location with respect to the radio impacts the filtered multipath profile in two ways. First, the direct-path reflection of User B needs to travel a longer distance compared to User A. Thus pixels at the same location in the multipath profiles are not directly comparable. Second, the power of RF reflections decreases as the traveling distance increases. Therefore the breathing powers of User B's pixels are much smaller than User A. As a result, and as shown in Fig. 7-10, the data distributions of User A and User B are significantly mismatched. Below, we explain the two methods we use to align the distributions.



(a) User A's profile after aligning bed location. (b) User B's profile after aligning bed location.

Figure 7-11: **The multipath profiles in Fig. 7-9 after aligning bed locations.** Now the direct path pixels of both Users A and B are at the same location.

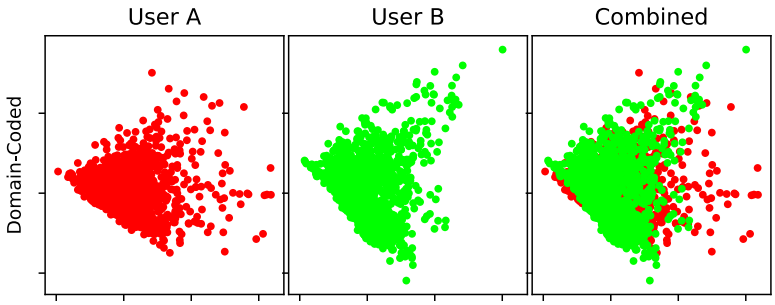


Figure 7-12: **Visualization of data distribution of User A (red) and User B (green) after bed alignment and power normalization.** Compared to Fig. 7-10, we can see that the two distributions are much better aligned.

(a) Aligning Bed Locations: We first align the relative location between the bed and radio across all users. While not all differences can be eliminated, this way, we ensure that all direct-path reflections have the same traveling distance. And since we cannot ask every user to move their bed, we can do this alignment virtually by reducing or increasing traveling distances of all the RF reflections in the multipath profile. This method brings us two advantages: 1) it keeps the original RF reflection pattern, and 2) we only need to know the bed location to perform this alignment.

Identifying the bed position can be done manually, however it is tedious and error-prone. Instead, we propose a robust and accurate way of measuring bed location for aligning. For each user, we do a pixel-wise summation for all of his filtered multipath profiles. Since the direct signal path has the shortest traveling distance, usually it has the highest breathing power in the filtered multipath profiles. Therefore, the pixel with the highest sum will give us an accurate estimation of the bed location. We additionally apply a Gaussian filter with a sigma of 1 to erase small location mismatches.

Looking back to Fig. 7-8, there is another mismatch. In User A's case, the radio device

is on the right-hand side of the user, and for User B, the device is on his left-hand side. This indicates that, when two users are both facing the device, they are actually facing different directions (to the right in Fig. 7-9(a) and to the left in Fig. 7-9(b)). Therefore we also align directions by flipping the angles. In all of the following discussions and results, -90° (left) represents the direction facing the device (to the right for User A and to the left for User B), and 90° (right) represents the opposite direction.

Fig. 7-11 shows the filtered multipath profiles of Users A and B above after aligning the bed location. We can see that the direct path pixels of both A and B are now at the same location.

(b) Power Normalization: RF signals attenuate with distance. Hence, the power in a particular pixel in the filtered multipath profile depends on the path length, more so than the sleep posture, as shown in Fig. 7-11. This dependence can prevent model generalization to a new target user if the target's room layout differs from the source user. Thus we would like to eliminate this dependence. To deal with this issue, we normalize the power distribution in each pixel of the filtered multipath profile (i.e., for each data point, we subtract the mean of the distribution and divide by its standard deviation).

Data distributions for User A and B after both aligning the bed location and normalizing the power are plotted in Fig. 7-12, and they are aligned much better compared to Fig. 7-10.

■ 7.5.3 Target Data Augmentation

Given the small number of calibration points, our information about the target user is limited. One solution is to perform data augmentation. In computer vision tasks, researchers have long been using augmentation techniques such as cropping, rotating, and horizontal flipping to help the model to capture data invariances. After those augmentations, images are still valid images. However, this is not true for our multipath profile. For example, flipping means that the furthest pixel becomes the closest, and the longest indirect reflections becomes the direct-path reflection. Therefore such standard augmentation techniques for images will break the spatial structure of the multipath profile.

Instead, we use data points from the source users that are similar to the calibration points from the target user. Specifically, our augmentation process contains the following steps: First, we align all the data points from all the users (including the calibration points),

as described in the previous section. Then given one calibration point (x_0, y_0), we first select all the points (x_i, y_i) that satisfy the condition that the angle difference between y_0 and y_i is smaller than a certain threshold (the default is 20 degrees). Then for those selected points, we further sort them based on the similarity of their multipath profiles to the calibration point as captured by the L2 distance: $\|x_i - x_0\|_2$. Finally, we pick the data points most similar to the calibration point (specifically we pick the 30 most similar source points to each calibration point). We refer to the set of augmented data points as the virtual target.

When adapting the neural network model from a particular source user to the target user, we combine the augmented data points with the labeled data from that source user to train the model and improve the model's performance on the target user. Note that we do not use the calibration points in training any of the adapted models. We hold the calibration points and use them to select the most effective adapted models as explained in the next section.

■ 7.5.4 Majority Voting

So far, we adapted each of the source models to transfer its knowledge to the virtual target. However, some of the source users in our training set may be very different from the target user, and despite adaptation, their knowledge may not translate well to the target user. Thus we need a mechanism to detect models that are well adapted to the target and combine their predictions.

We define validation accuracy as the model's accuracy on the target's calibration points. This validation accuracy is an estimate of the adapted model's true accuracy on the target data. Thus, we use this validation accuracy to evaluate the source's compatibility with the target. We filter out models that have bad validation accuracy (accuracy worse than 10% compared to the best model), and perform a majority vote among the models with good accuracy.

The majority vote is performed as follows: We create a histogram of the predictions where each angle degree has its own bin. Then we smooth this histogram with a Gaussian filter with a standard deviation of 20. Finally we pick the angle that has the highest value after smoothing as our final predicted angle. It is worth mentioning that, the smoothing is performed in a circular way, *i.e.*, the last bin and the first bin are connected.

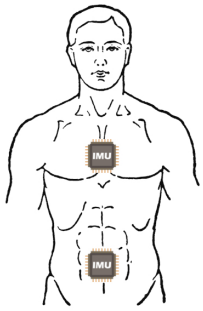


Figure 7-13: **The placement of accelerometers on the subject's body.** The accelerometers are used to collect the ground truth posture.

7.6 Experiment Setup

7.6.1 Data Collection

All experiments with human subjects were approved by our IRB, and we have obtained informed consent from every subject. In total, we have collected 224 nights of data from 26 subjects (17 male subjects and 9 female subjects).[3] Subjects' bed sizes cover most common sizes, from twin-size (1m wide) to king-size (2m wide). Each subject sleeps alone in his/her own bedroom. For each subject, we install the radio device on the wall to the side of the bed. The distance between the bed and the radio ranges from 0 meters (right above the bed) to 4 meters (on the other side of the bedroom).

7.6.2 Ground Truth Collection

To collect the ground truth postures, for each night, we ask the subject to wear two accelerometers, one on the chest and one on the abdomen. Both accelerometers are fixed on the body using sport tapes to prevent sliding during sleep. In Fig. 7-13, we show the placement of accelerometers on the body.

We align both accelerometers so that the accelerometer's x's positive points towards the subject's head, and z's positive points opposite to the body. Then the angle of body orientation y can be calculated using the following equation [281]:

$$y = atan2(a_y, a_z)$$

where $atan2$ is the 2-argument arctangent function.

[3] After data cleaning described in Sec. 7.6.2

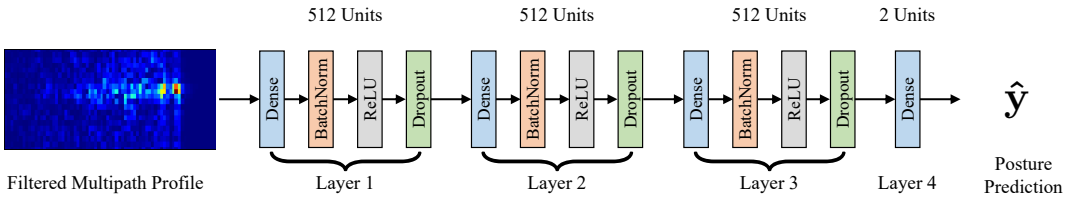


Figure 7-14: **Architecture of the neural network.**

We use two accelerometers for mutual validation. There are many factors that can lead to poor data quality. Most commonly, sensors may fall off from the body during sleep, or the subject may fail to align the sensors correctly. By having two accelerometers, we can identify bad data points because readings from those two will no longer be equal. Such bad data points are then excluded from training and testing.

7.6.3 Radio Specification

We use a standard FMCW radio similar to the one used in past work [282, 220, 219, 283]. The radio sweeps the frequencies from $5.4\ GHz$ to $7.2\ GHz$ and transmits at sub-milliwatt power in accordance with the FCC regulations. The radio is equipped with two antenna arrays: a horizontal array and a vertical array. Each array is capable of dividing space into 20 (angle) by 71 (distance) pixels, with an angular resolution of $\sim 8^{\circ}$ and a distance resolution of $\sim 20cm$. In total we have $2 \times 20 \times 71$ different pixels.

7.6.4 Model Implementation

We use a fully-connected neural network with 4 layers. After each hidden layer, there is a layer of Batch Normalization [284], a layer of ReLU and a layer of Dropout. Fig. 7-14 illustrates the neural network architecture.

7.7 Evaluation

In this section, we empirically evaluate the performance of BodyCompass.

7.7.1 Metrics

We define *Angle Error* as follows: For a single data point (stable segment) with a ground truth angle of y and a predicted angle of y', its *Angle Error* is the difference between y and

y', as defined in Eq. 7.2. Angle error is always within the range from $0°$ to $180°$.

$$e(y, y') = |(y - y' + 180) \bmod 360 - 180| \tag{7.2}$$

We use following two metrics that are based on angle error to evaluate the performance of our system.

- **Average Angle Error** (Error): Average Angle Error is the weighted average of all the angle errors for all stable segments in the testing dataset, where the weight of each stable segment is set its time the duration.
- **Threshold Accuracy** (Accuracy): Since most past work computes accuracy with respect to a few key postures (supine, right side, left side, and prone), we similarly estimate accuracy as the percentage of time that the angle error between the prediction and the ground truth is smaller than $45°$. This gives us an intuitive understanding of the percentage of time we predict the direction of the user correctly.

■ 7.7.2 Evaluation Setting

Depending on the amount of data available from the target subject, we present results under three different evaluation settings:

- **1-Week**: If we collected enough data from the target subject, we can directly train on the target's data, without transfer learning. Under this setting, for each subject in our dataset, we report the result with leave-one-night-out cross-validation, i.e., using one night for testing and the remaining nights for training, and repeat this process until all the nights have been tested.

- **1-Night**: In this setting we limit ourselves to only one night of labeled data from the target subject, and the rest of data from the target is unlabeled. One night is not enough for training. Therefore, we use this one night of data for our calibration points as discussed in Sec. 7.5.

- **16-Minutes**: To further reduce the effort required from users, we present the results with only 8 labeled data points from the target subject. We assume that those 8 points cover

Table 7-1: **Evaluation results under three different settings with different methods (BodyCompass, k-NN, Random Forest (RF), XGBoost [285] (XGB)).** Baseline methods are evaluated under two scenarios: All (A): trained with data from all the subjects; Target (T): trained with data from the target subject only. Note that BodyCompass significantly outperforms all three baselines under all settings.

	BodyCompass	k-NN (A)	k-NN (T)	RF (A)	RF (T)	XGB (A)	XGB (T)
Angle Error (1-week)	$15.3° \pm 4.4°$	NA	$31.3° \pm 9.7°$	NA	$33.8° \pm 13.0°$	NA	$33.8° \pm 13.3°$
Accuracy (1-week)	$94.1\% \pm 4.3\%$	NA	$77.7\% \pm 9.8\%$	NA	$75.4\% \pm 12.0\%$	NA	$75.5\% \pm 12.9\%$
Angle Error (1-night)	$25.6° \pm 6.7°$	$43.1° \pm 11.0°$	$40.6° \pm 11.0°$	$52.5° \pm 17.0°$	$45.4° \pm 15.1°$	$53.9° \pm 16.2°$	$49.2° \pm 13.1°$
Accuracy (1-night)	$86.7\% \pm 6.7\%$	$65.2\% \pm 10.5\%$	$67.8\% \pm 10.2\%$	$54.8\% \pm 14.5\%$	$62.2\% \pm 13.8\%$	$53.5\% \pm 14.2\%$	$59.9\% \pm 10.5\%$
Angle Error (16-min)	$28.3° \pm 8.7°$	$59.1° \pm 19.0°$	$60.6° \pm 19.0°$	$58.4° \pm 20.2°$	$55.0° \pm 18.9°$	$60.7° \pm 20.1°$	$65.1° \pm 13.1°$
Accuracy (16-min)	$83.7\% \pm 6.8\%$	$50.3\% \pm 14.6\%$	$46.4\% \pm 17.0\%$	$51.0\% \pm 14.9\%$	$52.2\% \pm 15.0\%$	$48.7\% \pm 15.8\%$	$42.8\% \pm 11.4\%$

most common positions of that user. This can be achieved by asking the user to emulate sleeping in his common sleep postures. In our evaluation, We select those calibration points from the existing sleep dataset by clustering the subject's sleep postures, and picking the center points for each cluster. If the resulting stable segment is longer than 2 minutes we use only a window of two minutes. Thus, collecting these 8 calibration points can be done in 16 minutes.

7.7.3 Evaluation of BodyCompass's Performance

To evaluate the effectiveness of BodyCompass, we compare its performance with three baselines: k-NN (k-Nearest Neighbors), Random Forest and XGBoost [285]. Note that all baselines and BodyCompass take filtered multipath profiles as input. Since the baselines do not have transferability capability like BodyCompass it is not clear how to train them when the available labeled data from the target subject is limited, i.e., in the 1-night and 16-minutes settings. Thus, for these settings, we evaluate the baselines' performance under two different training setups: 1. using data from all the available subjects; 2. using data only from the target subject.

Table 7-1 compares BodyCompass with the baselines for three different amounts of labeled data from the target subject: 1-week, 1-night, and 16-minutes. The table shows that BodyCompass significantly outperforms all three baselines under all settings. Specifically, BodyCompass and the baselines achieve their best performance when there is sufficient data from the target user. In such setting, BodyCompass's accuracy is 94%, whereas best accuracy across all baseline methods is only 77.7%.

The table also shows that when the amount of labeled data from the target subject is

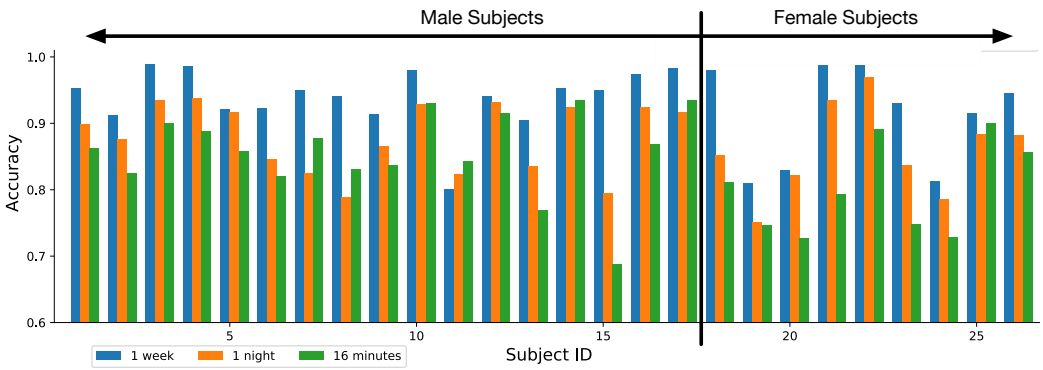
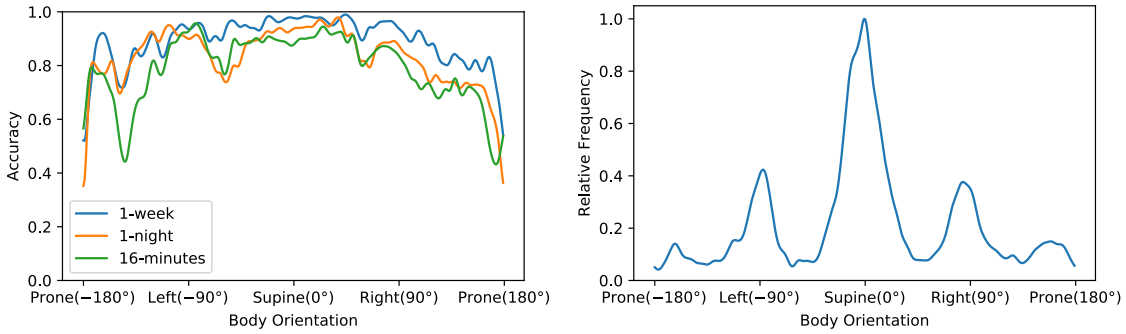


Figure 7-15: **Accuracy for each of our subjects under three different test settings.** Subjects are separated by their gender.

limited (e.g., in the 1-night setting and 16-minutes setting), the accuracies of all baselines are significantly reduced – 27.4% reduction for k-NN, 24.4% reduction for Random Forest, and 26.8% reduction for XGBoost. This is because of the natural variability in sleep postures. For example, even in the same body orientation, a slight change in arm or leg positions can cause a change in the pattern of RF reflections. All three baselines do not have the ability to handle such variability in the absence of a large amount of labeled data from the target. And since differences between different subjects are large, data from other subjects cannot help the baselines improve their robustness (e.g. under 1-night setting, data from other subjects is detrimental to the final performance). In contrast, BodyCompass aligns the distribution across different users, and performs data augmentation to battle this variability. As a result, BodyCompass can sustain high accuracy of 83.7% even with only 16 minutes of labeled data from the target user.

Next, we zoom in on BodyCompass and check the accuracy for each target user. In Fig. 7-15, we plot the average accuracy for each subject with our system under three different settings: 1-week, 1-night, 16-minutes. One can see that while transfer learning gives good accuracy across subjects, not all subjects have equal accuracy. Subjects who are more different from the source subjects will naturally have a lower accuracy when they have only 1-night of labeled data or 8 calibration points. We notice that, in our dataset, on average the accuracy on male subjects is higher than female subjects. Given the small number of subjects, it is not clear whether this accuracy gap is due to gender differences or is specific to the individuals in our dataset.

Another interesting aspect is that we see a slight increase of the accuracy of Subject #11



(a) Accuracy w.r.t. Orientation.

(b) Amount of labeled data as a function of orientation (normalized by the amount of data for the supine case).

Figure 7-16: **Accuracy and amount of labeled data for each body orientation.**

when moving from 1-week setting to a transfer setting like 1-night or 16-minutes. This is because while collecting data from Subject #11, the accelerometers often fell off during sleep. As a result, a significant part of each night from this subject had bad data with no accurate labels and was ignored. In this case, transfer learning can potentially have higher accuracy because it leverages labeled data from other users.

Finally, Fig. 7-16(a) presents the accuracy as a function of body orientation under all three settings. Our system has a high accuracy across all body orientations except for some corner cases where we lack enough training data (See the amount of labeled data for each angle in Fig. 7-16(b)). For example, at angle $-180°$, our system has low accuracy due to the fact that the amount of labeled data for this posture is $1/20$ amount of labeled data for the supine case, *i.e.*, $0°$.

■ 7.7.4 Evaluating the Components of BodyCompass

We evaluate the contribution of each component in our system by evaluating the performance of the system without that component. Specifically, we evaluate the contribution of using multipath profiles, breathing-filtering multipath profiles, distribution alignment, data augmentation, majority voting, fully-connected neural network, and circular loss. Removing the contribution of the breathing-filtering multipath profiles, data alignment and data augmentation components is straightforward, yet for the rest, we present the following substitutions:

- **Substitution for the Multipath Profiles**: Instead of taking the multipath profile as input,

Table 7-2: **Evaluation of the various components of BodyCompass.** The table shows the accuracy under a 1-night setting for the whole system and for the system without a particular component.

	Angle Error	Threshold Accuracy
Full System	$25.6° \pm 6.7°$	$86.7\% \pm 6.7\%$
CNN instead of Fully-Connected Network	$29.5° \pm 10.1°$	$82.5\% \pm 10.0\%$
Direct Path instead of Multipath	$43.0° \pm 11.8°$	$67.7\% \pm 9.5\%$
L2 loss instead of Circular Loss	$33.1° \pm 12.4°$	$77.7\% \pm 13.2\%$
w/o Breathing Filtering	$30.0° \pm 10.2°$	$81.7\% \pm 9.3\%$
w/o Distribution Alignment	$33.4° \pm 11.9°$	$78.5\% \pm 10.8\%$
w/o Data Augmentation	$30.7° \pm 10.2°$	$81.5\% \pm 8.6\%$
w/o Majority Voting	$31.2° \pm 11.2°$	$80.5\% \pm 9.7\%$

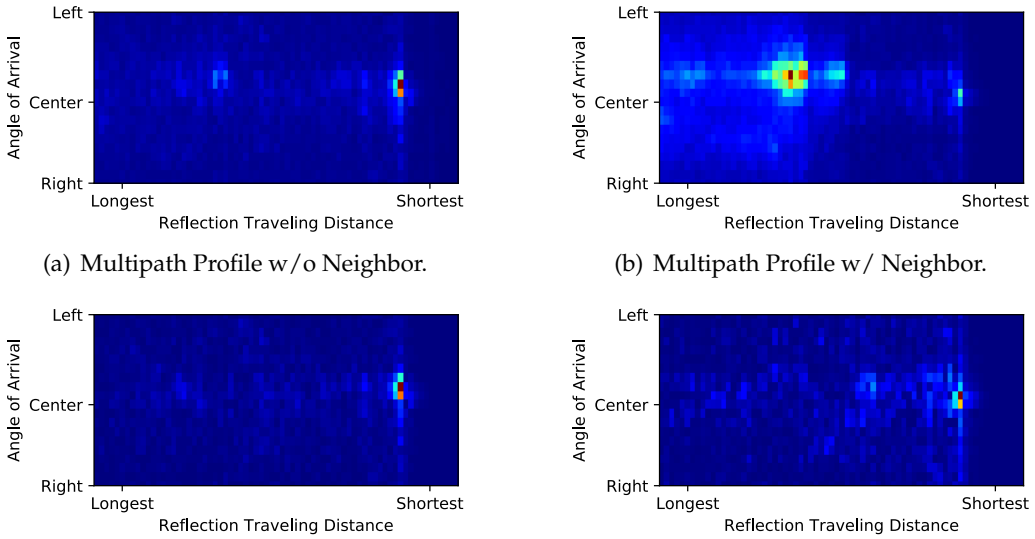
we zoom in on the voxels from the bed area, and take only those voxels. This is equivalent to focusing on the direct path only, and ignoring any indirect paths that involve signals that bounced off the person and other objects in space.

- **Substitution for the Fully-Connected Network**: We substitute the fully-connected neural network with a convolutional neural network (CNN) and evaluate the resulting performance. The CNN model follows the AlexNet model [286].

- **Substitution for the Circular Loss**: We can directly regress the angle and use the standard L2 loss.

- **Substitution for Majority Voting**: Instead of training on each source subject and performing majority voting, we can combine labeled data from all source subjects and train only one model.

Table 7-2 provides the evaluation results for BodyCompass's components. All experiments are conducted under 1-night setting. The first row shows BodyCompass's accuracy 1-night setting when all components are active. Comparing the first row with the other rows in the table shows that each of BodyCompass's components offers a considerable improvement to the overall performance, and the removal of any component results in reduced accuracy.



(a) Multipath Profile w/o Neighbor.

(b) Multipath Profile w/ Neighbor.

(c) Filtered Multipath Profile w/o Neighbor.

(d) Filtered Multipath Profile w/ Neighbor.

Figure 7-17: **Robustness to moving neighbors.** The figures in the top row show the multipath profile without and with a moving neighbor. Figures in the bottom row *filtered* multipath profile without and with a moving neighbor. The figures show that filtering the multipath profile eliminates extraneous movements from other people, hence boosts the robustness of the system.

■ 7.7.5 Sensitivity Study

In this section, we test our system's robustness to various factors such as the presence of other people in the environment and their movements, whether the subject has shallow or deep breathing, and the exact location of the radio. We note that when collecting data in the wild, *i.e.*, in people's own homes during their natural overnight sleep, we have no control over the above factors. In fact, in all of the experiments reported in the previous sections, we leave the radio in the home of the subject for about a week to collect the data. We have no control over when the subject goes to bed, where in the bed they sleep, whether their sleep location changes from one night to the next, how they breathe, and who else is at home and how they move while the subject is asleep. Thus, we cannot run sensitivity tests in the wild. We run these tests in a controlled environment where the subject is lying in bed but they are not asleep. In each case, the subject lies in bed at a particular body orientation, while we change the parameter we want to study, and measure BodyCompass's accuracy. The subject then changes his body orientation and we repeat the measurements while varying the parameter of interest.

Table 7-3: **Performance w/ neighbor movements.**

	Angle Error	Accuracy
Full System	12.7°	100%
w/o Filtering	53.6°	58.3%

Table 7-4: **Performance when subjects breathe at different strengths.**

Strength	Angle Error	Accuracy
Deep	$13.6° \pm 13.5°$	$97.2\% \pm 4.8\%$
Shallow	$8.6° \pm 4.8°$	$100\% \pm 0\%$

Sensitivity to Movements by Other People

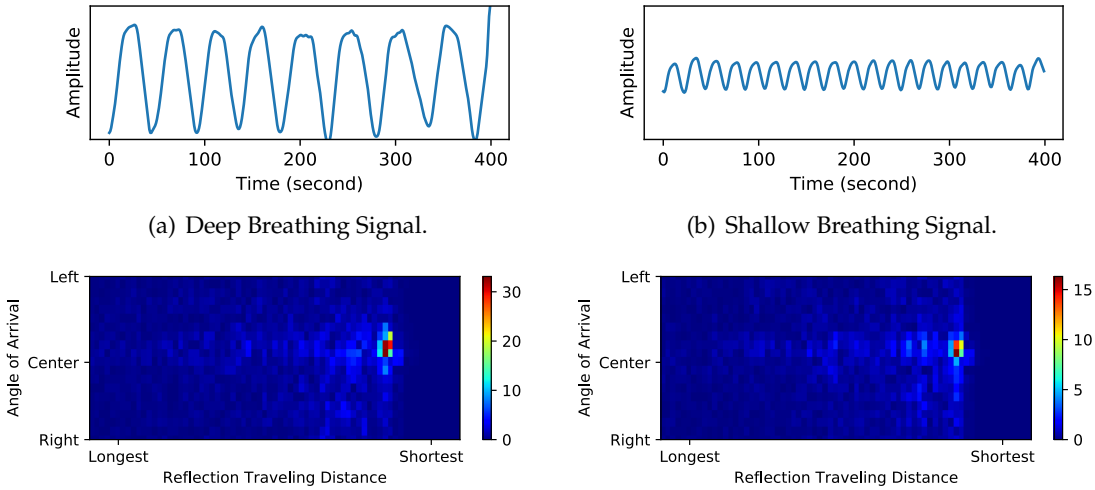
We evaluate our system's performance when there is a neighbor moving in an adjacent room. The subject sleeps approximately 3 meters away from the radio, whereas the neighbor is about 7 meters away from the radio. We ask the subject to lay down for 2 minutes in each of the following postures: supine, facing left, facing right, and prone. We train the system with examples in which the subject was alone without anyone moving in the neighboring room. We use 6 examples for each sleep posture for training. During testing, we bring a second person to the adjacent room and ask them to move at will, while the subject is lying in bed. We collect 3 examples of each sleep posture for testing.

Fig. 7-17 shows that our filtering of the multipath profile makes the system robust to extraneous movements such as the presence of a neighbor. The figure plots the multipath profile as well as the filtered multipath profile. As the figure shows, the neighbor's movements have a significant impact on the unfiltered multipath profile (Fig. 7-17(b)), yet its impact is removed after filtering the multipath profile using the method in Sec 7.3.2 (Fig. 7-17(d)).

Table. 7-3 presents BodyCompass's accuracy results averaged across the test examples. It shows that our system is robust against extraneous movements by other people.

Sensitivity to Breathing Strength

In this section, we investigate BodyCompass's robustness to variations in breathing strength, *i.e.*, to people having shallow vs. deep breathing. As in the previous section, we ask the subject to lie in bed in various sleep postures and for each posture, vary their breathing



(a) Deep Breathing Signal.

(b) Shallow Breathing Signal.

(c) Multipath Profile w/ Deep Breathing.

(d) Multipath Profile w/ Shallow Breathing.

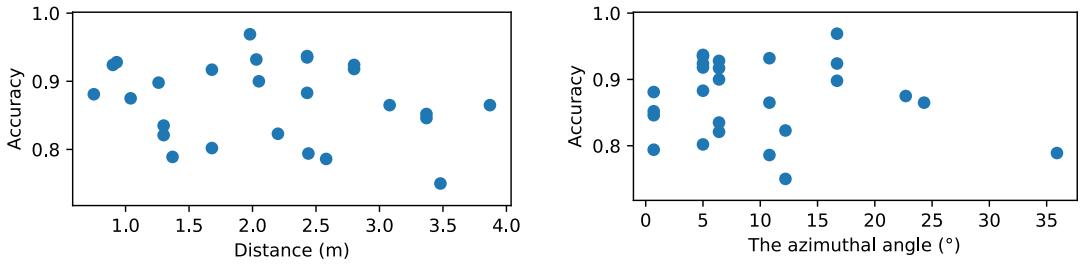
Figure 7-18: **Breathing signals and their corresponding multipath profiles.** Although the amplitude and frequency of the two breathing signals are quite different, their multipath profiles have similar patterns.

depth and rate. We repeat the test for each sleep posture, and with 3 different subjects. During testing, we ask the subject to perform 2 groups of experiments. Each group contains 4 postures (supine, facing left, facing right, and prone) and each posture is repeated 3 times. For the first group, we ask the subject to breathe deeply and slowly, and for the second group, we ask the subject to breathe shallowly and quickly. In Fig. 7-18, we show the breathing signals as well as their corresponding multipath profiles. Table. 7-4 reports the average accuracy for different breathing strength. It shows that BodyCompass has high accuracy for both shallow and deep breathing.

We also note that the accuracy results in these sensitivity tests are higher than the numbers presented in Table. 7-1 for testing in the wild. This is because when the subject is awake, the subject is able to accurately control his posture; In contrast, when the subject is asleep, the limbs can take various positions; Also the subjects may use pillows to support their bodies, and their use of pillows may change across days. Thus, overall there is much more variability in the wild.

Sensitivity to Device Location

Because the radio has directional antennas and the breathing signal is relatively weak, the device should face the bed to get a good SNR. However, we do not require the device to



(a) Distance from the radio to the subject.

(b) The azimuthal angle from the device to the subject's chest.

Figure 7-19: **Scatter plots of accuracy w.r.t. difference location settings.** As two plots show, our system can accommodate a wide range of location settings.

be exactly facing the chest of the person, nor do we require the device to be at a specific distance from the person.

In this section, we show that BodyCompass is fairly robust to variability in device distance and deviation from facing the chest of the person. In contrast to the previous two sensitivity tests, our in-the-wild deployments exhibit significant diversity in terms of device distance and azimuthal angle with respect to the person's chest. Specifically, in terms of distance between the chest and the device, our deployments cover a range from $0.5m$ to $4m$. In terms of the angle between the device and the person's chest, our deployments cover a range up to plus/minus $35°$, where a zero degree means that the device is facing the chest of the person.

Fig. 7-19 plots the accuracy of BodyCompass as a function of the distance to the person, and the angle to person's chest. The results in the figure show that BodyCompass works reliably for different location settings.

■ 7.7.6 Example Application: Monitoring the Frequency of Posture Changes

The frequency of posture shift (moving from one posture to another) during sleep is an important sleep-related metric. The literature shows that posture shift frequency is correlated with aging [287, 288] and sleep qualities [288]. Further for patients with Parkinson's disease, less frequent nocturnal turnovers reflects a deterioration in the disease condition [240].

Note that not every motion is a posture change. For example, when the user just changes his arm position, his body angle remains the same. To thoroughly evaluate our system's performance, we adopt the definition of posture change in [287]: to be quantified

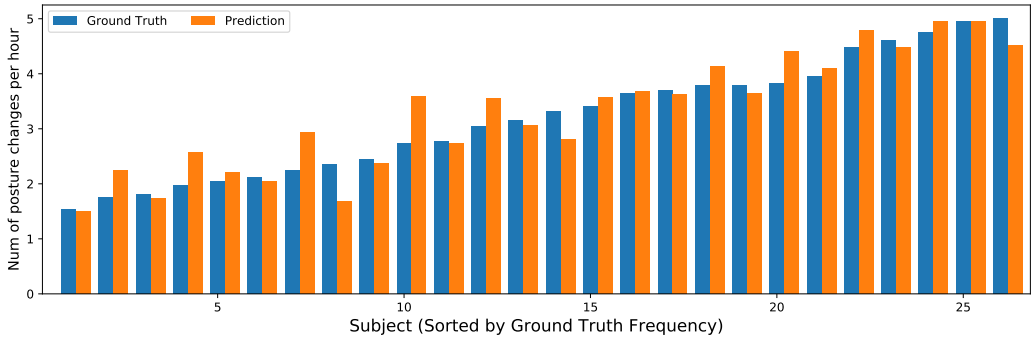


Figure 7-20: **Ground truth and predictions of posture shift frequency for each subject.**

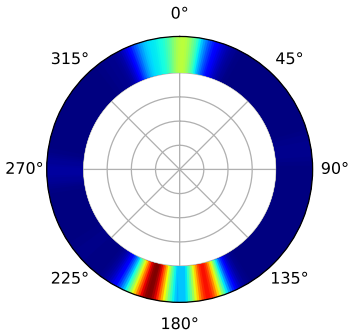


Figure 7-21: **Angle histogram of Subject #15.** Histogram is color-coded into the ring, as a more reddish color represent more occurrence.

as a posture change, a body angle change of at least 30° is required.

Fig. 7-20 plots the frequency of posture shifts for all 26 subjects, ordered in ascending order. The figure compares the ground truth and our model with 1-night of training data from the target user. The figure shows that our model is capable of tracking the frequency of posture shifts accurately, with an average relative error of only 10.3%. These results indicate that our model can be used to track changes in posture shift in Parkinson's patients.

■ 7.7.7 Failure Case Analysis

Looking back at Fig. 7-15, we can see that even when the amount of labeled data is scarce, our system is still able to deliver satisfactory accuracies for most subjects. However, there are few exceptions where we can see a significant reduction in performance when the amount of labeled data is reduced. For example, for Subject #15, we have the largest reduction in performance, from 95.0% to 68.7%. To understand the reason, we plot his angle histogram in Fig. 7-21. We can see that most of his time is spent in prone position (sleep on

stomach), and he never sleeps facing left or right. Recall in Sec. 7.5.2, we explicitly align the distribution to be the same. However this subject's intrinsic distribution is not similar to any other subjects. Therefore the alignment cannot fully succeed, which causes bad transfer performance. We expect to see an increase of our system's performance on this subject if we have more subjects with similar sleep postures.

7.8 Summary

In this chapter, we present BodyCompass, a wireless system that provides accurate sleep posture monitoring in the wild. By explicitly extracting RF reflections from the user and designing appropriate machine learning algorithms, our system can accurately capture the user's posture and is able to transfer its knowledge to a new home with minimal additional data. A user study in 26 different homes with 26 subjects and more than 200 nights shows that BodyCompass is highly accurate, with an accuracy of 94% using 1 week of data from the user, and 83.7% using only 16 minutes of data. We believe that this work can serve as a practical sleep posture monitoring system, enabling easy adoption and helping doctors and patients address this unmet need.

CHAPTER 8

Towards Fair Medical Imaging AI across Environments and Subgroups

As AI models are increasingly deployed in real-world clinical settings, it is crucial to evaluate not only model performance, but also potential biases towards specific demographic groups [197, 289]. While deep learning has achieved human-level performance in numerous medical imaging tasks [290], existing literature indicates a tendency for these models to manifest existing biases in the data, causing performance disparities between protected subgroups [4, 291]. For instance, chest X-ray classifiers trained to predict the presence of disease systematically underdiagnose Black patients [291], potentially leading to delays in care. To ensure the responsible and equitable deployment of such models, it is essential to understand the source of such biases and, where feasible, take actions to correct them [292].

Recent studies have unveiled the surprising ability of deep models to predict demographic information such as self-reported race, sex, and age from medical images [293], achieving performance far beyond that of radiologists. These insights raise the concern of disease prediction models leveraging demographic features as heuristic "shortcuts" [294] – correlations that are present in the data, but have no real clinical basis, for instance deep models using the hospital as a shortcut for disease prediction [295].

In this work, we investigate four questions. First, whether disease classification models also utilize demographic information as shortcuts, and whether such demographic short-

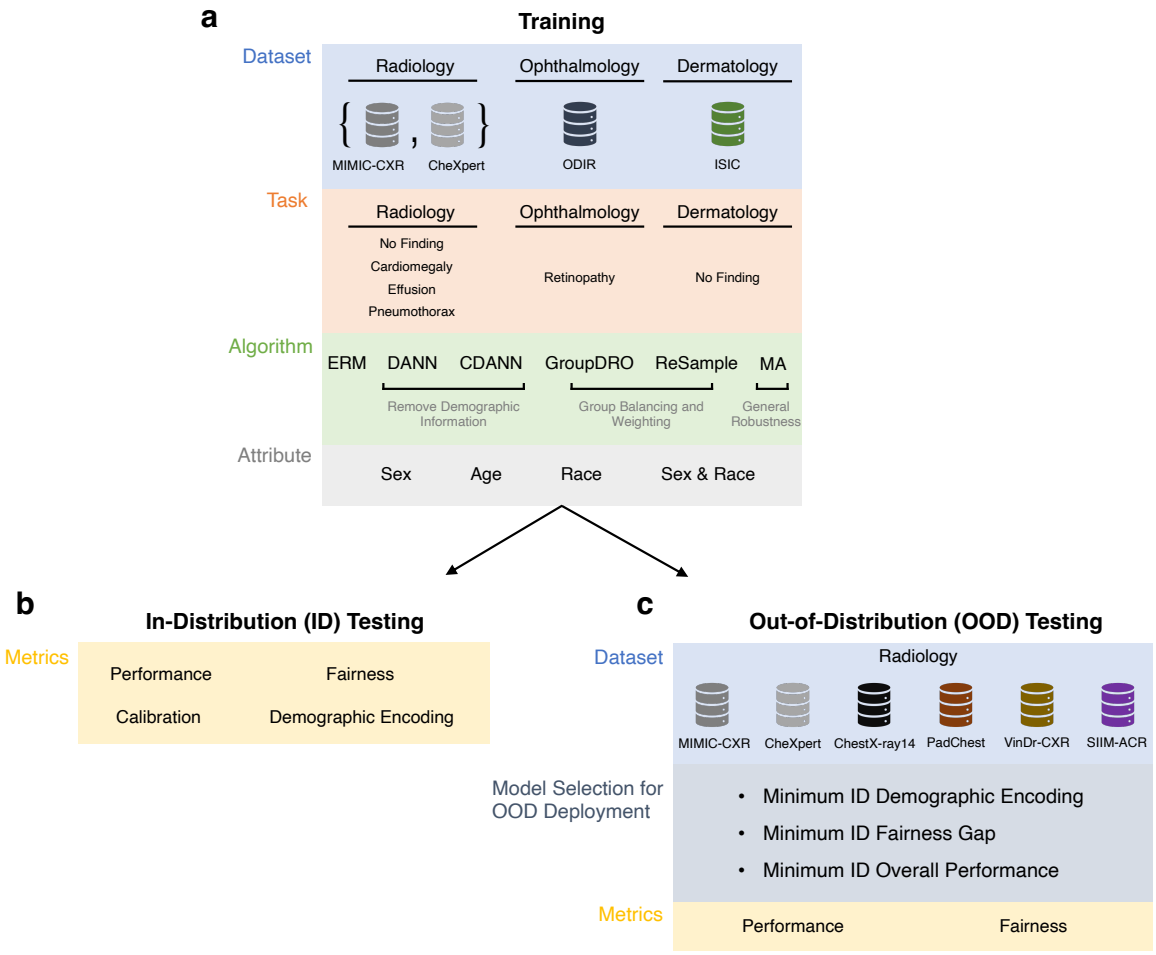


Figure 8-1: **Overall experimental pipeline. a,** We train a grid of deep learning models on medical images from a variety of modalities on several clinical tasks. We apply a variety of state-of-the-art algorithms to mitigate shortcuts, for up to four demographic attributes (where available). **b,** We evaluate each model in-distribution (i.e., on the same dataset where it is trained), along the axis of performance, fairness, amount of demographic encoded, and calibration. **c,** We evaluate the performance and fairness of chest X-ray classification models on out-of-distribution (OOD) domains. To mimic a realistic deployment setting where OOD samples are not observed, we choose the "best" model based on several in-distribution selection criteria.

cuts result in biased predictions. Second, we evaluate the extent to which state-of-the-art methods can remove such shortcuts and create “locally optimal” models that are also fair. Third, we consider real-world clinical deployments settings where shortcuts may not be valid in the out-of-distribution data, in order to dissect the interplay between algorithmic fairness and shortcuts when data shifts. Finally, we explore which algorithms and model selection criteria can lead to “globally optimal” models that maintain fairness when deployed in an out-of-distribution setting.

We perform a systematic investigation into how medical AI leverages demographic shortcuts through these questions, with an emphasis on fairness disparities across both in-distribution training and external test sets. Our primary focus is on chest X-ray (CXR) prediction models, with further validation in dermatology and ophthalmology. Our X-ray analysis draws upon six extensive, international radiology datasets: MIMIC-CXR [181], CheXpert [182], NIH [296], SIIM [297], PadChest [298], and VinDr [299]. We explore fairness within both individual and intersectional subgroups spanning race, sex, and age [291]. Our assessment uncovers compelling new insights on how medical AI encodes demographics, and the impact this has on various fairness considerations, especially when models are applied outside their training context during real-world domain shifts, with actionable insights on what models to select for fairness under distribution shift.

8.1 Datasets

We utilize six publicly available chest X-ray datasets as described in Fig. 8-2. We focus on four binary classification tasks that have been shown to have disparate performance between protected groups [291, 300]: "No Finding", "Effusion", "Pneumothorax", and "Cardiomegaly".

We also examine medical AI applications in dermatology and ophthalmology. Specifically, we use the ISIC dataset [301] with "No Finding" as the task for dermatological imaging, and the ODIR dataset [302] with "Retinopathy" as the task for ophthalmology images.

All datasets used in this study are publicly available. The MIMIC-CXR and VinDr-CXR datasets are available from PhysioNet after the completion of a data use agreement and a credentialing procedure. The CheXpert dataset, along with associated race labels, is available from the Stanford AIMI website. The ChestX-ray14 (NIH) dataset is available to download from the National Institute of Health Clinical Center. The PadChest dataset can be downloaded from the Medical Imaging Databank of the Valencia Region. The SIIM-ACR Pneumothorax Segmentation dataset can be downloaded from its Kaggle contest page. The ISIC 2020 dataset can be downloaded from the SIIM-ISIC Melanoma Classification Challenge page. The ODIR dataset can be obtained from the ODIR 2019 challenge hosted by Grand Challenges.

Unless otherwise stated, we train models on MIMIC-CXR [181], and evaluate on an

		MIMIC	CheXpert	NIH	SIIM	PadChest	VinDr
	Location	Boston, MA	Stanford, CA	Bethesda, MD	Bethesda, MD	Alicante, Spain	Hanoi, Vietnam
	# Images	357,167	222,792	112,120	11,582	144,478	6,354
	% Frontal	64.5	85.5	100.0	100.0	69.1	100.0
	Sample Image						
Sex (%)	**Female**	47.8	40.7	43.5	44.6	50.4	43.1
	Male	52.2	59.3	56.5	55.4	49.6	56.9
Race (%)	**Asian**	3.1	10.5	-	-	-	-
	Black	15.6	5.4	-	-	-	-
	White	61.0	56.4	-	-	-	-
	Other	20.3	27.8	-	-	-	-
Age (%)	**0-18**	-	-	4.8	5.0	3.7	21.8
	18-40	13.8	13.9	27.7	27.3	9.2	16.0
	40-60	31.1	31.1	43.9	42.9	26.5	27.1
	60-80	40.0	39.0	22.7	23.9	38.0	30.0
	80-100	15.1	16.0	0.9	0.9	22.6	5.1
Intersection (%)	**Asian Female**	1.5	4.5	-	-	-	-
	Asian Male	1.6	6.0	-	-	-	-
	Black Female	9.3	2.6	-	-	-	-
	Black Male	6.3	2.7	-	-	-	-
	White Female	27.3	22.2	-	-	-	-
	White Male	33.8	34.1	-	-	-	-
	Others Female	9.8	11.3	-	-	-	-
	Others Male	10.5	16.5	-	-	-	-
Task Prevalence (%)	**No Finding**	39.8	10.0	53.8	-	34.9	41.2
	Effusion	20.0	38.6	11.9	-	5.9	7.5
	Pneumothorax	3.4	8.7	4.7	28.4	0.3	0.7
	Cardiomegaly	14.9	12.1	2.5	-	9.5	22.6

Figure 8-2: **Demographic and label characteristics of the six X-ray datasets used in this study.**

OOD dataset created by merging CheXpert [182], NIH [296], SIIM [297], PadChest [298], and VinDr [299]. We include all images (both frontal and lateral), and split each dataset into 70% train, 15% validation, 15% test sets. Note that only MIMIC-CXR and CheXpert have patient race information available. For MIMIC-CXR, demographic information was obtained by merging with MIMIC-IV [303]. For CheXpert, separate race labels were obtained from the Stanford AIMI website. Where applicable, we drop patients with missing values for any attribute.

For all datasets, we exclude samples where the corresponding patient has missing age or sex. For ODIR and ISIC, we drop samples from patients younger than 18 and older than 80 due to small sample sizes (i.e., smaller than 3% of the total dataset).

We scale all images to 224×224 for input to the model. We apply the following image augmentations during training only: random flipping of the images along the horizontal axis, random rotation of up to 10 degrees, and a crop of a random size (70% to 100%) and a random aspect ratio (3/4 to 4/3).

8.2 Methods

8.2.1 Model Training

We train a grid of deep convolutional neural networks [304] on MIMIC-CXR (radiology), ODIR (ophthalmology), and ISIC (dermatology), varying the classification task. Our approach follows prior work which achieves state-of-the-art performance in these tasks [291] using standard training or Empirical Risk Minimization (ERM) [305]. We also evaluate five algorithms designed to remove spurious correlations, or increase model fairness during training. We categorize these algorithms into those that (1) reweight samples based on their group to combat underrepresentation (ReSample [160], GroupDRO [134]), (2) adversarially remove group information from model representations (DANN [113], CDANN [306]), and (3) more generically attempt to improve model generalization (MA [307]). In total, our analysis encompassed a total of 3,456 models trained on MIMIC-CXR, corresponding to the cartesian product of 4 tasks, 4 demographic attributes, 6 algorithms, 12 hyperparameter settings, and 3 random seeds.

8.2.2 Assessing the Fairness of ML Models

In order to assess the fairness of ML models, we evaluate the metrics described above for each demographic group, as well as the difference in the value of the metric between groups. Equality of TPR and TNR between demographic groups is known in the algorithmic fairness literature as equal odds [308]. As the models we study in this work are likely to be used as screening or triage tools, the cost of a False Positive (FP) may be different from the cost of a False Negative (FN). In particular, for No Finding prediction, FPs (corresponding to underdiagnosis [291]) would be more costly than FNs, and so we focus on the FPR (or TNR) for this task. For all remaining disease prediction tasks, we focus on the FNR (or TPR) for the same reason. Equality in one of the class conditioned error rates is an instance of equal opportunity [309].

Finally, we also examine the per-group ECE and ECE gap between groups. Note that zero ECE for both groups (i.e., calibration per group) implies the fairness definition known as sufficiency of the risk score [308]. We emphasize that differences in calibration between groups is a significant source of disparity, as consistent under or over-estimation of risk for a particular group could lead to under or over-treatment for that group at a fixed operating

threshold relative to the true risk [310].

8.2.3 Evaluation Methods

To evaluate the performance of disease classification in medical imaging, we use the following metrics: the area under the ROC curve (AUC), True Positive Rate (TPR), True Negative Rate (TNR), and Expected Calibration Error (ECE).

We also reported AUC, which is the area under the corresponding ROC curves showing an aggregate measure of detection performance. Finally, we report the Expected Calibration Error (ECE) [189], which we compute using the *netcal* library [311].

8.2.4 Training Details

We train DenseNet-121 [304] models on each task, initializing with ImageNet [312] pretrained weights. We evaluate six algorithms: empirical risk minimization (ERM [305]), resampling to equalize group size (Resample [160]), group distributionally robust optimization (GroupDRO [134]), domain adversarial training (DANN [113]), domain adversarial training conditioned on the label (CDANN [306]), and weight averaging (MA [307]).

For each combination of task, algorithm, and demographic attribute, we conduct a random hyperparameter search [313] with 15 runs. During training, for a particular attribute, we evaluate the validation set worst-group validation AUROC every 1,000 steps, and early stop if this metric has not improved for 5 evaluations. We tune the learning rate and weight decay for all algorithms, and also tune algorithm specific hyperparameters as mentioned in the original works. We select the hyperparameter setting that maximizes the worst-attribute validation AUROC. Confidence intervals are computed as the standard deviation across three different random seeds for each hyperparameter setting.

To obtain the level of demographic encoding within representations, we first compute representations using a trained disease prediction model. We freeze these representations, and train a multi-class multinomial logistic regression model to predict the demographic group using the training set using the scikit-learn library [314]. We vary the L_2 regularization strength between 10^{-5} and 10, and select the model with the best macro-averaged AUROC on the validation set. We report the macro-averaged AUROC on the test-set.

8.2.5 Decomposing Out-of-Distribution Fairness

Here, we present a first approach towards decomposing the fairness gap in an out-of-distribution environment as a function of the in-distribution fairness gap, and the impact that the distribution shift has in each group. In particular, let D_{src} and D_{tar} be the source and target datasets, respectively. Let $g \in G$ be a particular group from a set of groups. Let $L_f(g, D)$ be an evaluation metric for a model f, which is decomposable over individual samples, i.e., $L_f(g, D) = \sum_{(x,y,g') \in D; g'=g} l(f(x), y)$. Examples of such metrics are the accuracy, TPR, or TNR. Then, we can decompose:

$$\begin{aligned} L_f(g_1, D_{tar}) - L_f(g_2, D_{tar}) = &[L_f(g_1, D_{src}) - L_f(g_2, D_{src})] + \\ &[L_f(g_2, D_{src}) - L_f(g_2, D_{tar})] - \\ &[L_f(g_1, D_{src}) - L_f(g_1, D_{tar})] . \end{aligned}$$

The left-hand term is the fairness gap in the out-distribution environment, and the three terms on the right are (1) the fairness gap in the in-distribution data, (2) the impact of the distribution shift on g_2, and (3) the impact of the distribution shift on g_1. We note that to achieve a low fairness gap in the out-of-distribution environment, it is important not only to minimize the in-distribution fairness gap (term 1), but also to minimize the difference in how the distribution shift impacts each group (term 2 - term 3).

8.2.6 Statistical Analysis

Correlation. To calculate the correlations between variables, we used Pearson correlation coefficients and their associated p-value (two-sided t-test, $\alpha = 0.05$). 95% CI for the Pearson correlation coefficient was calculated.

Increase in OOD fairness gap. One-tailed Wilcoxon rank-sum test ($\alpha = 0.05$) was used to assess the increase in OOD fairness gap compared to oracle models.

Confidence intervals. We use the non-parametric bootstrap sampling to generate confidence intervals: random samples of size n (equal to the size of the original dataset) are repeatedly sampled 1,000 times from the original dataset with replacement. We then estimate the increase in OOD fairness gap compared to oracle using each bootstrap sample ($\alpha = 0.05$).

All statistical analysis was performed with Python version 3.9 (Python Software Foundation).

8.3 Results

8.3.1 Algorithmic encoding of protected attributes leads to model fairness gaps

We separately train deep learning models for our four distinct CXR prediction tasks ("No Finding", "Cardiomegaly", "Effusion", "Pneumothorax"), as well as "Retinopathy" in ophthalmology, and "No Finding" in dermatology. Each model consists of a feature extractor followed by a disease prediction head. We then employ a transfer learning approach, wherein we keep the weights of the feature extractor frozen and retrain the model to predict sensitive attributes (e.g., race). This allows us to assess the amount of attribute-related information present in the features learned by each model as measured by the area under the ROC curve (AUC) for attribute prediction (details in the Methods section). We extend prior work [315] demonstrating that deep models trained for disease classification encode demographic attributes, and test across a wider range of settings. As Fig. 8-3a, 8-3c, and 8-3e confirms, the penultimate layer of different disease models contains significant information about four demographic attributes (age, race, sex, and the intersection of sex and race), and that is consistent across different tasks and medical imaging modalities.

We then assess the fairness of these models across demographic subgroups as defined by equal opportunity [309], i.e., discrepancies in the model's false negative rate (FNR) or false positive rate (FPR) for demographic attributes. We focus on underdiagnosis [291], i.e., discrepancies in FPR for "No Finding" and discrepancies in FNR for other diseases. For each demographic attribute, we identify two key subgroups with sufficient sample sizes: age groups "80-100" (n=8,063) and "18-40" (n=7,319); race groups "White" (n=32,732) and "Black" (n=8,279); sex groups "female" (n=25,782) and "male" (n=27,794); sex & race groups "White male" (n=18,032) and "Black female" (n=5,027). In all tasks, we observe that the models displayed biased performance within the four demographic attributes, as evidenced by the FNR disparities (Fig. 8-3b). The observed gaps can be as large as 30% for age. The same results hold for the other two imaging modalities (Fig. 8-3d, 8-3f).

We further investigate the degree to which demographic attribute encoding "shortcuts"

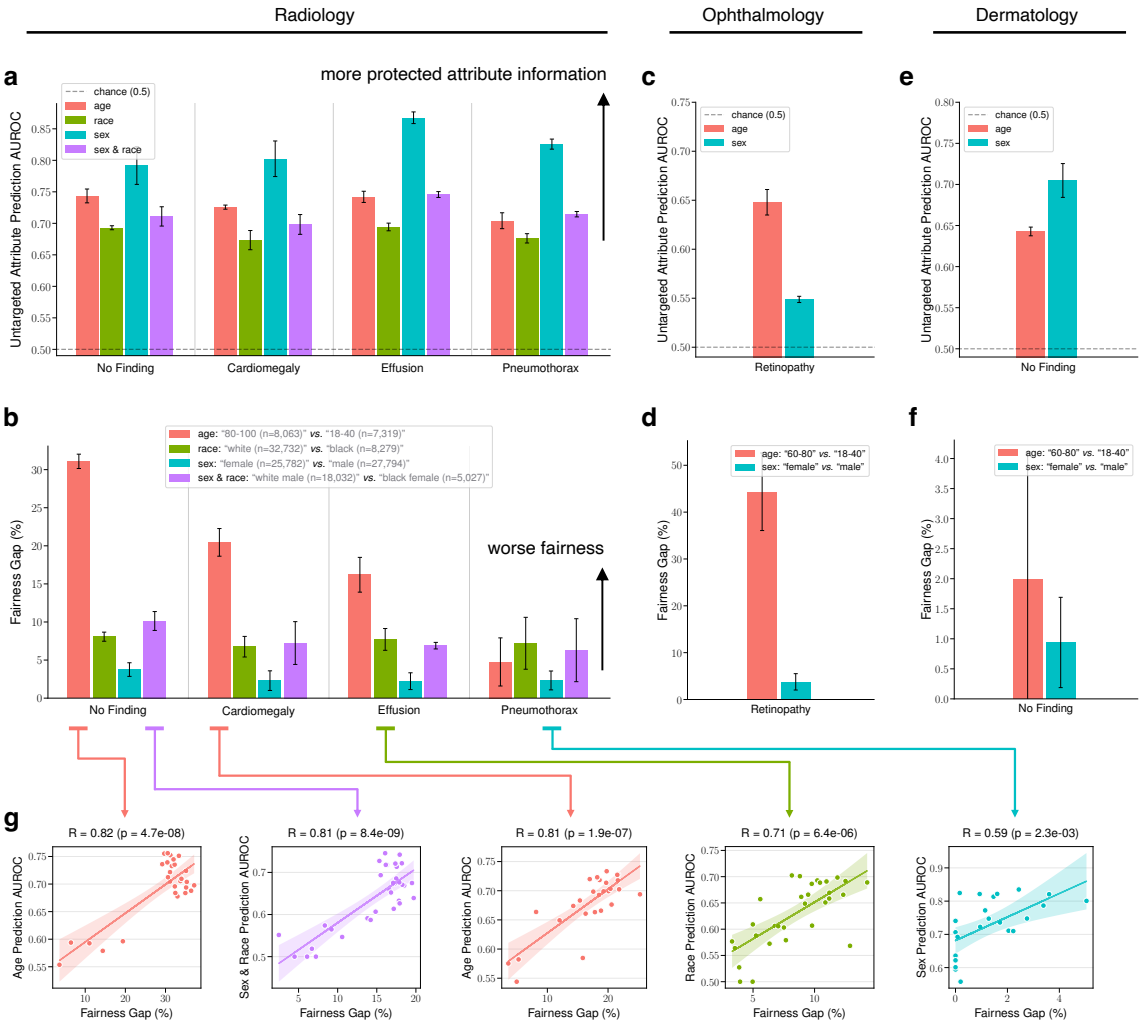


Figure 8-3: **Medical imaging models encode sensitive attributes and are unfair across subgroups.** **a,** The area under the ROC curve (AUROC) of demographic attribute prediction from frozen representations for the best ERM model. We train ERM models on MIMIC-CXR to predict four different binary tasks. ERM representations encode demographic attributes to a high degree. **b,** The fairness gap, as defined by the FPR gap for No Finding, and the FNR gap for all other tasks for the best ERM model. ERM models exhibit high fairness gaps, especially between age groups. **c,** The AUROC of demographic attribute prediction from frozen representations for the best ERM model on the ODIR dataset (ophthalmology), following the same experimental setup. **d,** The fairness gap for the best ERM model on the ODIR dataset (ophthalmology). **e,** The AUROC of demographic attribute prediction from frozen representations for the best ERM model on the ISIC dataset (dermatology), following the same experimental setup. **f,** The fairness gap for the best ERM model on the ISIC dataset (dermatology). **g,** The correlation between attribute prediction performance and fairness for all learned models. We exclude models with suboptimal performance, i.e., with an overall validation AUROC below 0.7. The attribute prediction AUROC shows a high correlation with the fairness gap (No Finding, age: R=0.82, p=4.7e-08; No Finding, sex & race: R=0.81, p=8.4e-09; Cardiomegaly, age: R=0.81, p=1.9e-07; Effusion, race: R=0.71, p=6.4e-06; Pneumothorax, sex: R=0.59, p=2.3e-03). Each bar and its error bar indicate the mean and standard deviation across 3 independent runs.

may impact model fairness. We note that a model encoding demographic information does not necessarily imply a fairness violation, as the model may not necessarily use this information for its prediction. For each task and attribute combination, we train different models with varying hyperparameters (see Sec. 8.2). We focus on the correlation between the degree of encoding of different attributes, and the fairness gaps as assessed by underdiagnosis. Fig. 8-3g shows that a stronger encoding of demographic information is significantly correlated with stronger model unfairness (No Finding, age: R=0.82, p=4.7e-08; No Finding, sex & race: R=0.81, p=8.4e-09; Cardiomegaly, age: R=0.81, p=1.9e-07; Effusion, race: R=0.71, p=6.4e-06; Pneumothorax, sex: R=0.59, p=2.3e-03). Such consistent observations indicate that models using demographic encodings as heuristic shortcuts also have larger performance disparities.

■ 8.3.2 Mitigating shortcuts creates locally optimal models that are fair and performant

We perform model evaluations first in the in-distribution (ID) setting, where ERM models trained and tested on data from the same source perform well. We compare ERM to state-of-the-art robustness methods that have been designed to effectively address fairness gaps while maintaining overall performance. As shown in Fig. 8-4a, ERM models exhibit large fairness gaps across age groups when predicting Cardiomegaly (i.e., models centered in the top right corner, FNR gap 20% between groups "80-100" and "18-40"). By applying debiasing robustness methods that correct demographic shortcuts, such as GroupDRO and DANN, the resulting models are able to close the FNR gap, while achieving similar AUROCs (e.g., the bottom right corner). Our results hold across different combinations of diseases and attributes (Fig. 8-4b).

To demonstrate the value of model debiasing, we further plot the set of ***locally optimal models*** - those on the Pareto front [316] that balance the performance-fairness tradeoff most optimally on ID data (Fig. 8-4a). Those models that lie on this front are "locally optimal", as they have the smallest fairness gap that can be achieved for a fixed performance constraint (e.g., AUROC > 0.8). In the ID setting, we find several existing algorithms that consistently achieve high ID fairness without losing overall performance for disease prediction (Fig. 8-4a, 8-4b).

Similar to our observations in radiology, we identify fairness gaps within subgroups

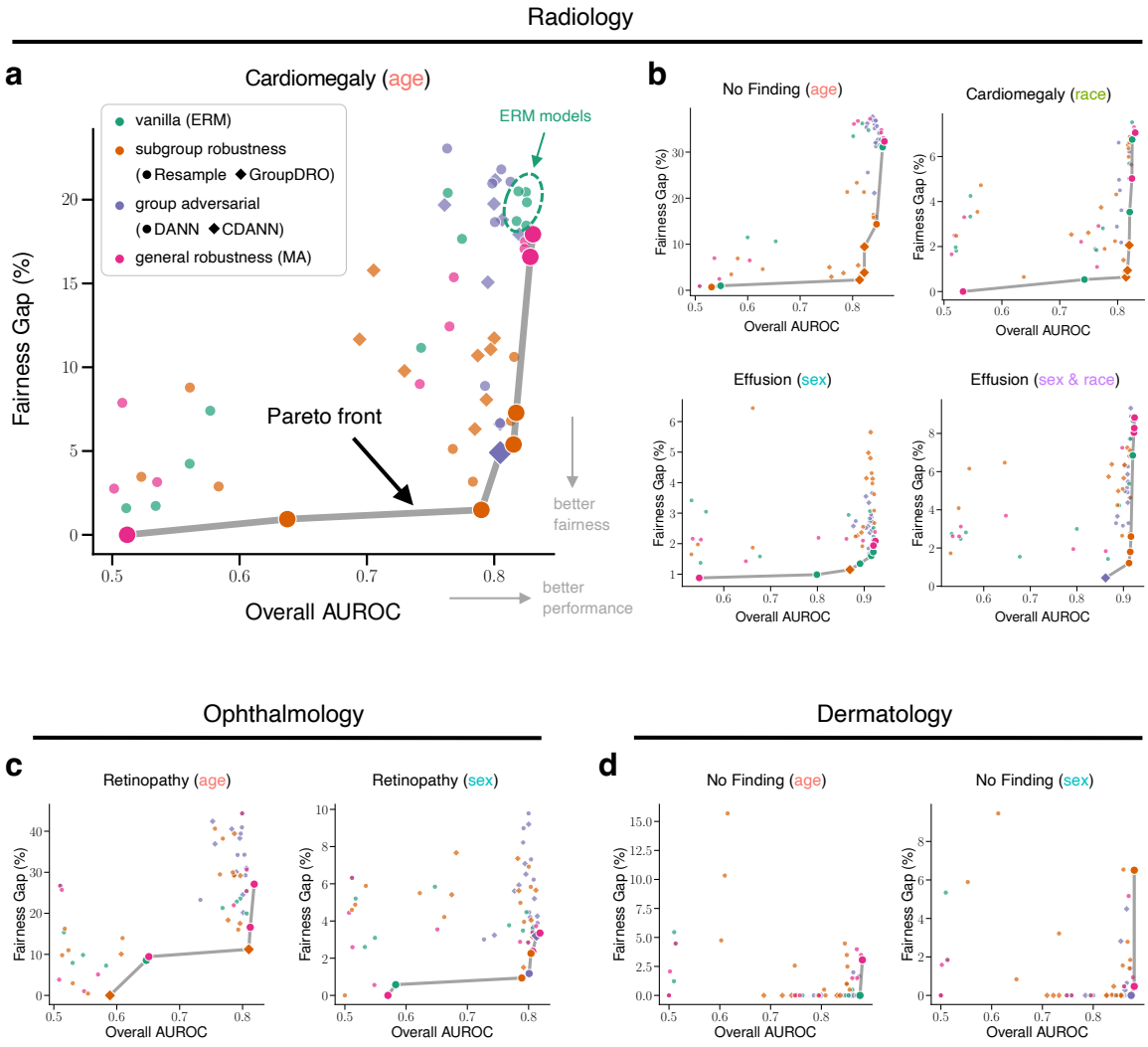


Figure 8-4: **Algorithms for removing demographic shortcuts mitigate in-distribution fairness gaps and maintain performance. a, b,** Trade-off between the fairness gap and overall AUROC for all trained models. Each plot represents a specific disease prediction task (e.g., Cardiomegaly) with a specific attribute (e.g., age). In each case, we plot the Pareto front, the best achievable fairness gap with a minimum constraint on the performance. **c, d,** Trade-off between the fairness gap and the overall AUROC on the ODIR dataset (ophthalmology) and ISIC dataset (dermatology).

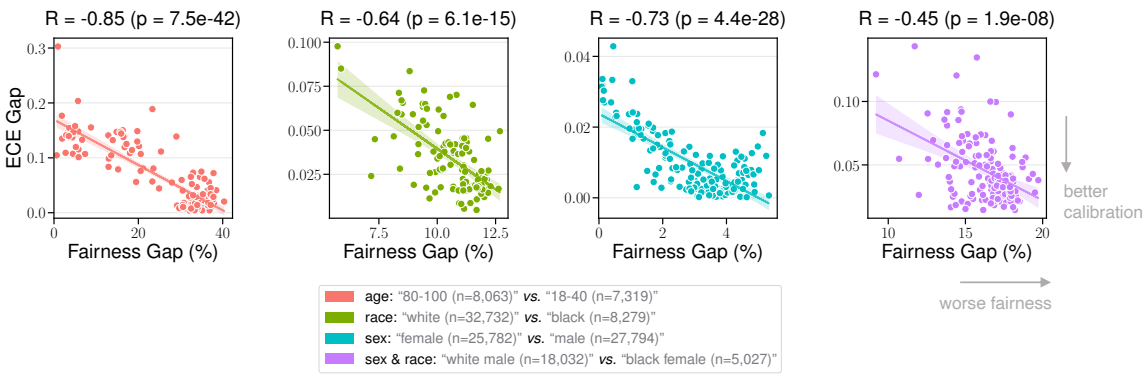


Figure 8-5: **The tradeoff between the fairness gap and the expected calibration error (ECE) gap.** For the No Finding task, we examine the trade-off between the fairness gap and the ECE gap (age: R=-0.85, p=7.5e-42; race: R=-0.64, p=6.1e-15; sex: R=-0.73, p=4.4e-28; sex & race: R=-0.45, p=1.9e-08).

based on age and sex in dermatology and ophthalmology, respectively (Fig. 8-3d, 8-3f). We further verify the Pareto front for both attributes, where similar observations hold that algorithms for fixing demographic shortcuts could improve in-distribution fairness while incurring minimal detriments to performance (Fig. 8-4c, 8-4d).

■ 8.3.3 Locally optimal models exhibit trade-offs in other metrics

We examine how locally optimal models that balance fairness and AUROC impact other metrics, as previous work has shown it is theoretical impossibility to balance fairness measured by probabilistic equalized odds and calibration by group [317, 318]. We find that optimizing fairness alone leads to worse results for other clinically meaningful metrics in some cases, indicating an inherent tradeoff between fairness and other metrics. First, for the "No Finding" prediction task, enforcing fair predictions across groups results in worse expected calibration error gap (ECE Gap, Fig. 8-5) between groups. Across different demographic attributes, we find a consistent statistically significant negative correlation between ECE Gap and Fairness Gap (age: R=-0.85, p=7.5e-42; race: R=-0.64, p=6.1e-15; sex: R=-0.73, p=4.4e-28; sex & race: R=-0.45, p=1.9e-08).

These findings stress that these models, though being locally optimal, exhibit worse results on other important and clinically relevant performance metrics. This uncovers the limitation of blindly optimizing fairness, emphasizing the necessity for more comprehensive evaluations to ensure the reliability of medical AI models.

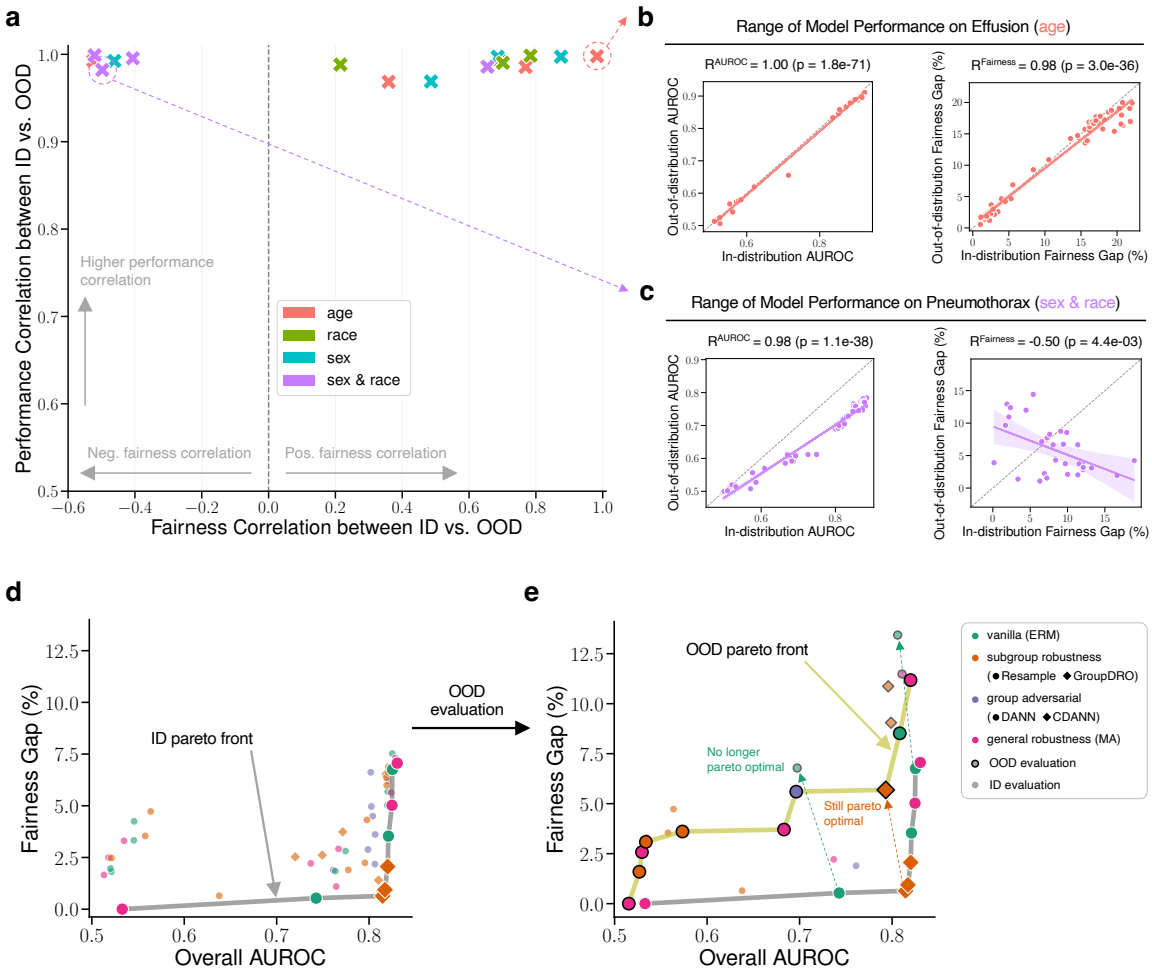


Figure 8-6: **The transfer of performance (overall AUROC) and fairness between the ID (MIMIC-CXR) and OOD datasets. a,** We plot the Pearson correlation coefficient of ID vs. OOD performance versus the Pearson correlation coefficient of ID vs. OOD fairness. Here, each point is derived from a grid of models trained on a particular combination of task and attribute. We find that there is a high correlation between ID and OOD performance in all cases, but the correlation between ID and OOD fairness is tenuous. **b, c,** We show how two particular points in the first plot are obtained; one where fairness transfers ("Effusion" with "age" as the attribute; R=0.98, p=3.0e-36), and one where it does not ("Pneumothorax" with "sex & race" as the attribute; R=-0.50, p=4.4e-3). **d, e,** We show the transformation of the ID Pareto front to the OOD Pareto front, for Cardiomegaly prediction using race as the attribute, finding that models that are Pareto optimal ID often do not maintain Pareto optimality OOD.

8.3.4 Locally optimal model fairness does not transfer under distribution shift

When deploying AI models in real settings, it is crucial to ensure that models can generalize to data from unseen institutions or environments. We directly test all trained models in the out-of-distribution (OOD) setting, where we report results on external test datasets that are unseen during model development. Fig. 8-6a illustrates that the correlation between ID and OOD performance is high across different settings, which has been observed in prior work [191]. However, we find that there is no consistent correlation between ID and OOD fairness. For example, Fig. 8-6b shows an instance where the correlation between ID fairness and OOD fairness is strongly positive ("Effusion" with "age" as the attribute; R=0.98, p=3.0e-36), while Fig. 8-6c shows an instance where the correlation between these metrics is actually significantly negative ("Pneumothorax" with "sex & race" as the attribute; R=-0.50, p=4.4e-03). Across 16 combinations of task and attribute, we find that 5 such settings exhibit this negative correlation, and 3 additional settings exhibit only a weak ($R < 0.5$) positive correlation. Thus, improving ID fairness may not lead to improvements in OOD fairness, highlighting the complex interplay between fairness and distribution shift [319].

In addition, we investigate whether models achieving ID Pareto optimality between fairness and performance will maintain in OOD settings. As shown for "Cardiomegaly" prediction using race as the attribute, models originally on the Pareto front ID (Fig. 8-6d) do not guarantee to maintain Pareto optimality when deployed in a different OOD setting (Fig. 8-6e).

8.3.5 Dissecting model fairness under distribution shift

To disentangle the OOD fairness gap, we present a way to decompose model fairness under distribution shift. Specifically, we decompose and attribute the change in fairness between ID and OOD to be the difference in performance change for each of the groups, i.e., the change in fairness is determined by how differently the distribution shift affects each group (details in Sec. 6.2).

In Fig. 8-7, we show an example of transferring a model to predict No Finding trained on CheXpert (ID) to the MIMIC-CXR data (OOD), while evaluating fairness across genders. We find that the model is fair with respect to the FPR gap in the ID setting (-0.1% gap, not significant), but has a significant FPR gap when deployed in the OOD setting

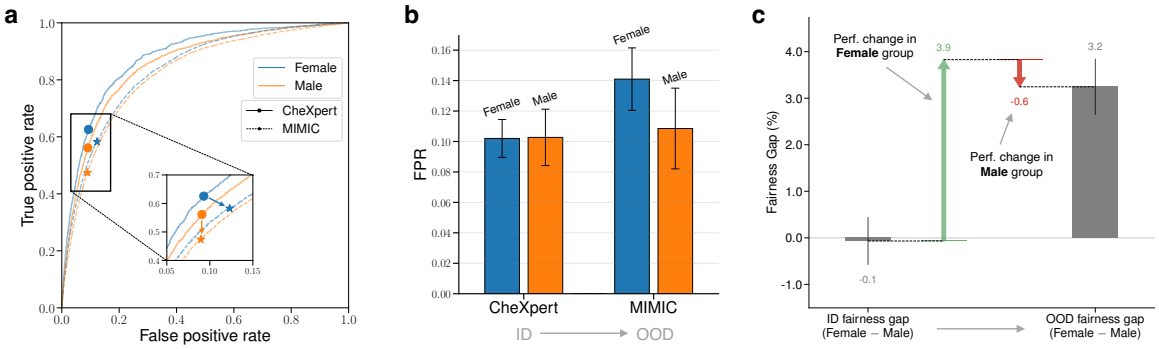


Figure 8-7: **Examining the gender biases of an ERM model for No Finding prediction, trained on CheXpert (ID) and deployed on MIMIC-CXR (OOD). a,** We plot the receiver operating characteristic (ROC) curves for each group and each dataset, marking the operating point of the model. **b,** This shift in the operating point results in a change in the FPR values of males and females on the OOD dataset, with both groups seeing increased underdiagnosis, but females are impacted more heavily. **c,** We decompose the OOD fairness gap as a function of the ID fairness gap, and the change in FPR for each of the groups, finding that the large increase in OOD fairness is primarily attributable to the increase in FPR for females. Each bar and its error bar indicate the mean and standard deviation across 3 independent runs.

(3.2%), with females being underdiagnosed at a higher rate (Fig. 8-7b).

We then segment this FPR gap by gender, and find that females experience an increase in FPR of 3.9%, while males experience an increase in FPR of 0.8% (Fig. 8-7c). In other words - the model becomes worse for both groups in an OOD setting, but to a much larger extent for female patients. This decomposition suggests that mitigation strategies which reduce the impact of the distribution shift on females could be effective in reducing the OOD fairness gap in this instance. We further extend this study to a larger set of tasks and protected attributes. Across all settings, the disparate impact of distribution shift on each group is a significant component, indicating that mitigating the impact of distribution shift is as important as mitigating ID fairness, if the goal is to achieve a fair model out-of-distribution.

■ 8.3.6 Globally optimal model selection for out-of-domain fairness

Fig. 8-6 shows that selecting a model based on in-distribution fairness may not lead to a model with optimal OOD fairness. Here, we examine alternate model selection criteria that may lead to better OOD fairness, when we only have access to ID data. Our goal is to find *"globally optimal"* models which maintain their performance and fairness in new domains. First, we subset our selection only to models that have satisfactory ID overall

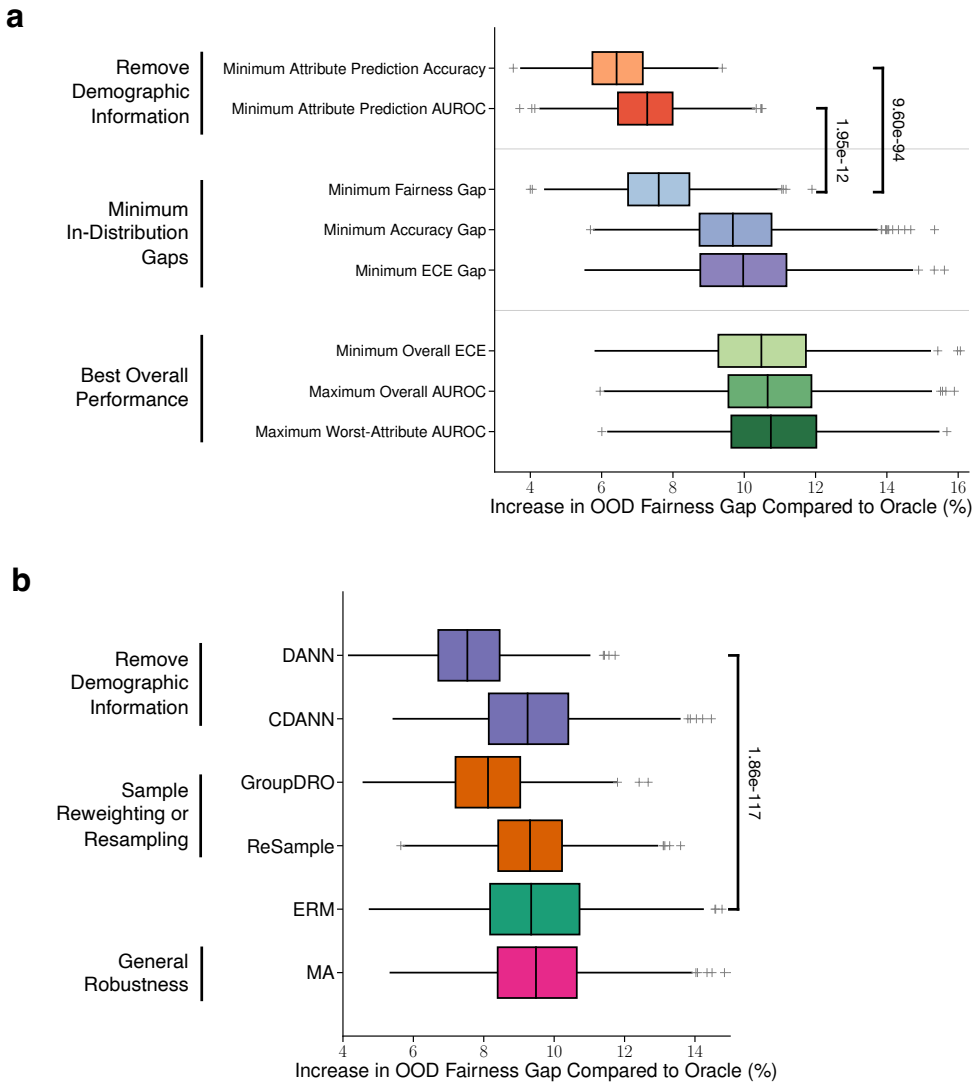


Figure 8-8: **OOD fairness of models with different model selection criteria and for different algorithms. a,** We vary the in-distribution model selection criteria, and compare the selected model against the oracle which chooses the model that is most fair OOD. We plot the increase in OOD fairness gap of the selected model over the oracle, averaged across 42 combinations of OOD dataset, task, and attribute. We find that selection criteria based on choosing models with minimum attribute encoding achieve better OOD fairness than naively selecting based on in-distribution fairness, or other aggregate performance metrics ("Minimum Attribute Prediction Accuracy" vs "Minimum Fairness Gap": p=9.60e-94, one-tailed Wilcoxon rank-sum test; "Minimum Attribute Prediction AUROC" vs "Minimum Fairness Gap": p=1.95e-12, one-tailed Wilcoxon rank-sum test). **b,** We select the model for each algorithm with the minimum in-distribution fairness gap. We evaluate its OOD fairness against the oracle on the same 42 settings. We find that removing demographic encoding (i.e., DANN) leads to the best OOD fairness ("DANN" vs "ERM": p=1.86e-117, one-tailed Wilcoxon rank-sum test). Error bars indicate 95% confidence intervals estimated using non-parametric bootstrap sampling (n=1,000) are shown.

performance (defined as those with overall validation AUROC no less than 5% of the best ERM model). As validated in Fig. 8-6, this set of models will also have satisfactory OOD performance.

Next, we propose eight candidate model selection criteria (Fig. 8-8a), corresponding to selecting the model from this set that minimizes or maximizes some in-distribution metric. We evaluate the selected model by its OOD fairness across five external datasets, each containing up to four attributes and up to four tasks, corresponding to a total of 42 settings. We compare the OOD fairness of the selected model to the OOD fairness of an "oracle", which observes samples from the OOD dataset and directly chooses the model with the smallest OOD fairness gap. For each setting, we compute the increase in fairness gap of each selection criteria relative to the oracle. In Fig. 8-8a, we report the mean across the 42 settings, as well as the 95% confidence interval computed from 1,000 bootstrap iterations. We find that, surprisingly, selecting the model with the minimum ID fairness gap may not be optimal. Instead, two other criteria based on selecting models where the embedding contains the least attribute information, lead to a lower average OOD fairness gap. For instance, we observe a significantly lower increase in OOD fairness gap by selecting models with the "Minimum Attribute Prediction Accuracy" as compared to "Minimum Fairness Gap" (p=9.60e-94, one-tailed Wilcoxon rank-sum test). The result echoes our finding in Fig. 8-3 that the encoding of demographic attributes is positively correlated with ID fairness.

Finally, we study the fairness of each algorithm in the OOD setting. We maintain the performance cutoff described above, and select the model for each algorithm with the lowest ID fairness gap. In Fig. 8-8b, we report the mean increase in OOD fairness gap relative to the oracle across the same 42 settings. We find that methods which remove demographic information from embeddings (specifically, DANN) lead to the lowest average OOD fairness gap ("DANN" vs "ERM": p=1.86e-117, one-tailed Wilcoxon rank-sum test). Our findings demonstrate that evaluating and removing demographic information encoded by the model in-distribution may be the key to "globally optimal" models which transfer both performance and fairness to external domains.

8.4 Discussion

We have demonstrated the interplays between the demographic encoding of attributes as "shortcuts" in medical imaging AI models, and how they change under distribution shifts. Importantly, we were able to validate our findings across global-scale datasets in radiology, and across multiple medical imaging modalities. The results show that algorithmic encoding of protected attributes leads to unfairness (Fig. 8-3), and mitigating shortcuts can reduce ID fairness gaps and maintain performance (Fig. 8-4). However, our results also show that there exists inherent tradeoff for clinically meaningful metrics beyond fairness (Fig. 8-5), and such fairness does not transfer under distribution shift (Fig. 8-6). We provide initial strategies to dissect and explain the model fairness under distribution shifts (Fig. 8-7). Our results further reveal actionable algorithm and model selection strategies for out-of-domain fairness (Fig. 8-8).

Our results have multiple implications. First, they offer a cautionary tale on the efficacy and consequences of eliminating demographic shortcuts in disease classification models. On the one hand, removing shortcuts addresses ID fairness which is a crucial consideration in fair clinical decision making [291]. On the other hand, the resulting trade-offs with other metrics and non-transferability to OOD settings raises the question about the long-term utility in removing such shortcuts. This is particularly complex in the healthcare setting, where the relationship between the demographics and the disease or outcome label are complex, variables can be mislabeled, and distribution shifts between domains are difficult to quantify [320].

Second, we frame demographic features as potential "shortcuts", which should not be utilized by the model to make disease predictions. However, some demographic variables could be a direct causal factor in some diseases (e.g., sex as a causal factor of breast cancer). In these cases, it would not be desirable to remove all demographic reliance, but instead match the reliance of the model on the demographic attribute to its true causal effect [321]. In the tasks we have examined here, demographic variables such as race may have an indirect effect on disease (e.g., through socioeconomic status) [322], which may vary across geographic location, or even time period. Whether demographic variables should serve as proxies for these causal factors is a decision that should rest with the model deployers [323, 320].

Third, we present a preliminary decomposition for diagnosing OOD model fairness changes, by expressing it as a function of the ID fairness gap, and the performance change of each group. We find that the disparate impacts of distribution shift on per-group performance is a significant contributor to lack of fairness in OOD settings. Our work suggests that, for practitioners trying to achieve fairness in models deployed in a different domain, mitigating ID fairness (e.g., through methods we have evaluated here) is at least as important as mitigating the impact of distribution shift for particular groups. However, building models robust to arbitrary domain shifts is, in general, a challenging task [133]. Having some knowledge or data about how the distributions may shift, or even the ability to actively collect data for particular groups, may be necessary [324]. Developing methods and deriving theoretical characterizations of fairness under distribution shift is an active area of research [319].

Fourth, the FDA, as the primary regulatory body for medical technologies [9], along with guidelines from a recent White House Executive Order on AI [325], does not require external validation of clinical AI models, relying instead on the assessment by the product creator. Our findings underscore the necessity for regular evaluation of model performance under distribution shift [142], challenging the popular opinion of a single fair model across different settings [326]. This questions the effectiveness of developer assurances on model fairness at the time of testing and highlights the need for regulatory bodies to consider real-world performance monitoring, including fairness degradation. Finally, when a model is deployed in any clinical environment, both its overall and per-group performance, as well as associated clinical outcomes, should be continuously monitored [327].

Finally, while we imply that smaller "fairness gaps" are better, enforcing these group fairness definitions can lead to worse utility and performance for all groups [317, 328], and other fairness definitions may be better suited to the clinical setting [329]. We note that these invariant notions of fairness could have drawbacks [330], as equalized odds are incompatible with calibration by group (Fig. 8-5), and enforcing equalized odds often lead to the "levelling down" effect in overall performance [328]. We present the Pareto curve showing the tradeoff between fairness and accuracy, allowing the practitioner to select a model that best fits their deployment scenario. In general, we encourage practitioners to choose a fairness definition that is best-suited to their use case, and carefully consider the performance-equality trade-off. The impact of minimizing algorithmic bias on real-world

health disparities, the ultimate objective, is complex [331], and there is no guarantee that deploying a fair model will lead to equitable outcomes. In addition, though we construct several models for clinical risk prediction in this chapter, we do not advocate for deployment of these models in real-world clinical settings without practitioners carefully testing models on their data and taking other considerations into account (e.g., privacy, regulation, interpretability) [197].

CHAPTER 9
Conclusion

In this dissertation, we present machine learning approaches and systems that extend healthcare capabilities beyond current clinical settings. By customizing machine learning models and algorithms, the presented technologies are able to close the healthcare gaps across time, location, and individuals, enabling new disease biomarker discovery and improving medical delivery and health equity. Specifically, the first part of the thesis introduces new machine learning algorithms to address inherent healthcare data challenges, including learning under *label scarcity*, tackling *data imbalance*, improving *domain generalization*, and maintaining fairness under *subpopulation shifts*. In the second part, we study the translation of these generic algorithms into practical systems that extend healthcare capabilities. These include the AI-enabled biomarker for early detection of Parkinson's disease, in-home touchless monitoring of sleep posture overnight, and practices for fair and ethical deployment of medical AI models in changing environments.

Our contributions span both machine learning and computational health. From a machine learning perspective, we introduce algorithms that account for the unique constraints posed by health data. Unlike the carefully curated datasets commonly used in other AI fields, imperfect health data present distinct challenges that necessitate robust algorithm design. Consequently, the algorithms developed not only enhance healthcare applications but are also versatile enough to be applied across other high-stakes domains. From a healthcare perspective, this dissertation adopts a data-driven computational approach that significantly extends healthcare capabilities. This approach holds the potential

to transform discovery, delivery, and equity in healthcare. While this dissertation presents only three implementations of such applications, they serve as driving examples for progressing towards the ultimate vision of a *generalist* medical AI that sees into the future, blends into our surroundings, and equalizes care for all.

The dissertation also has a broader impact on precision health and drug clinical trials. The AI-based sensing technologies could be deployed in the home to monitor sleep, vital signs, activities, and infer disease states in a passive and continuous manner. It could inform the caregiver of real-time changes in health status and help clinicians better understand disease progression. It could also be used in clinical trials to monitor medication response, improve safety, and speed up the drug development process. Notably, some of the technologies presented in this dissertation have already been deployed in the real world. In particular, our AI-based biomarker has been used in collaboration with University of Rochester Medical School to monitor sleep and gait in Parkinson's patients [16]. It has also been adapted to monitor COVID-19 patients remotely [24].

9.1 Future Directions

This thesis opens the door for several interesting future directions. Below, I outline directions that expand upon the presented research and move towards tackling open challenges in this field.

We begin with directions on the machine learning for healthcare side:

- *Interpretable Medical AI Models.* An exciting and crucial direction is enhancing the interpretability and human-compatibility of medical AI models. While AI models are powerful, they can often be fragile and opaque [332], which is particularly problematic in high-stakes fields such as medicine and healthcare. The biomarkers and systems developed here provide initial steps towards understanding model behaviors and decision-making processes (e.g., the attention layer and interpretation through EEG in Chapter 6). Future research should prioritize the development of models that are explainable, fostering trust among clinicians and enhancing their usability in clinical settings.

- *Combining Retrospective and Prospective Learning.* The dynamic and ever-changing nature of healthcare necessitates a dual approach in machine learning: leveraging insights from

historical data while constantly adapting to emerging trends. This dissertation has explored both retrospective analyses (e.g., Chapter 8) and prospective studies (e.g., Chapter 6 & 7). Future research could develop reliable offline and online learning frameworks that not only draw insights from complex historical health data but also adapt efficiently and proactively in real-time for effective decision-making.

- *Fairness under Shifts and Uncertainty.* In healthcare and other high-stakes applications, ensuring equity requires rethinking their behaviors under distribution shifts and uncertainties in evolving health and societal contexts. Building on this thesis in fairness under distribution shifts [7, 25, 26], future work could explore principled methods to quantify and mitigate biases under real-world shifts, uncertainties, and regulations. This involves addressing fairness on multiple fronts: individual, subgroup, and systemic levels, especially when algorithms are deployed in new environments or populations.

Next, we envision future directions for moving towards next-generation healthcare along multiple avenues:

- *Using AI to Understand Chronic Diseases.* This dissertation has demonstrated the potential of AI-enabled digital biomarkers for Parkinson's disease (Chapter 6). This modeling pipeline is an initial step toward a deeper understanding and management of chronic diseases. Future research could investigate phenotypes identification, discovering composite biomarkers, and measuring medication responses at a more personalized level. While this thesis has focused on Parkinson's, the framework could be expanded to include other neurological diseases such as ALS and Alzheimer's. Beyond neurological disorders, there is potential for this framework to be applied to a wider range of chronic conditions, including mental health issues and depression.

- *Closing the Sensing-Learning-Intervention Loop.* Another promising direction is to create a fully integrated "sensing-learning-intervention" ecosystem. This dissertation has highlighted the capabilities of AI-enabled devices for passive and remote health monitoring. By utilizing computational methods, these technologies can reveal insights outside clinics (Chapter 6 & 7). Taking this further, such a sensing-learning pipeline sets the stage for closing the loop by implementing AI-driven interventions that can modify behaviors and enhance health outcomes. We can then continually gather sensory data, establishing a self-sustaining loop that propels next-generation healthcare right in the home.

- *Human-Centered Medical AI.* Finally, incorporating a human-centric approach into medical AI systems is crucial for tackling unique challenges in health and medicine. Future research could focus on enhancing collaboration between humans and AI to create systems that are transparent, accountable, and capable of providing actionable insights while fostering trust among all stakeholders. This involves improving human-AI interactions using advanced tools, such as large language models (LLMs), to enhance patient-provider communications, support decision-making processes, and personalize interventions. Prioritizing the co-design of AI systems with clinicians, patients, and other stakeholders is crucial to ensure their ethical deployment.

APPENDIX A
Details and Results for SimPer

A.1 Dataset Details

In this section, we provide the detailed information of the six datasets we used in our experiments. Fig. A-1 shows examples of each dataset, and Table A-1 provides the statistics of each dataset.

RotatingDigits *(Synthetic Dataset)*. We create RotatingDigits, a synthetic periodic learning dataset of rotating MNIST digits [58], where samples are created with the original digits rotating on a plain background at rotational frequencies between 0.5Hz and 5Hz. The training set consists of $1,000$ rotating video clips (100 samples per digit number), each sample with a frame length of 150 and a sampling rate of 30Hz. The test set consists of $2,000$ rotating video clips (200 samples per digit number).

SCAMPS *(Human Physiology)*. The SCAMPS dataset [59] contains $2,800$ synthetic videos of avatars with realistic peripheral blood flow and breathing. The faces are synthesized using a blendshape-based rig with $7,667$ vertices and $7,414$ polygons and the identity basis is learned from a set of high-quality facial scans. These texture maps were sampled from 511 facial scans of subjects. The distribution of gender, age and ethnicity of the subjects who provided the facial scans can be found in [333]. Blood flow is simulated by adjusting properties of the physically-based shading material[1]. We randomly divide the whole dataset into training ($2,000$ samples), validation (400 samples), and test (400 samples) set.

[1] https://www.blender.org/

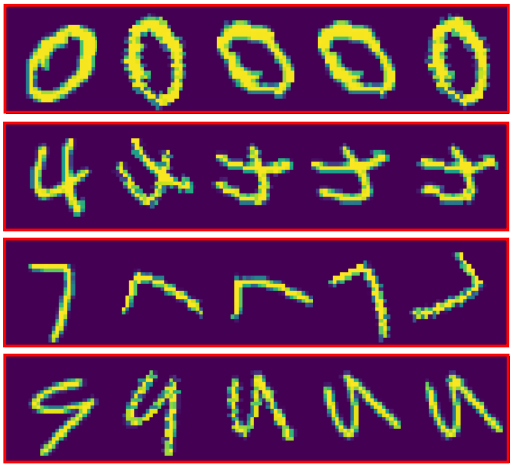
Each video clip has a frame length of 600 and a sampling rate of 30Hz.

UBFC *(Human Physiology)*. The UBFC dataset [60] contains a total of 42 videos from 42 subjects. The videos were recorded using a `Logitech C920 HD Pro` at 30Hz. A pulse oximeter was used to obtain the gold-standard PPG data (30Hz). The raw resolution is 640×480 and videos are recorded in a uncompressed 8-bit RGB format. We postprocess the videos by cropping the face region and resizing them to 36×36. We manually divide each video into non-overlapping chunks [45] with a window size 180 frames (6 seconds). The resulting number of training and test samples are 518 and 106, respectively.

PURE *(Human Physiology)*. The PURE dataset [61] includes 60 videos from 10 subjects (8 male, 2 female). The subjects were asked to seat in front of the camera at an average distance of 1.1 meters and lit from the front with ambient natural light through a window. Each subject was then instructed to perform six tasks with varying levels of head motion such as slow/fast translation between camera plane and head motion as well as small/medium head rotations. Gold-standard measurements were collected with a pulse oximeter at 60Hz. The raw video resolution is 640×480. We postprocess the videos by cropping the face region and resizing them to 36×36, and downsample the ground-truth PPG signal to 30Hz from 60Hz. We manually divide each video into non-overlapping chunks [45] with a window size 180 frames (6 seconds). The resulting number of training and test samples are $1,028$ and 226, respectively.

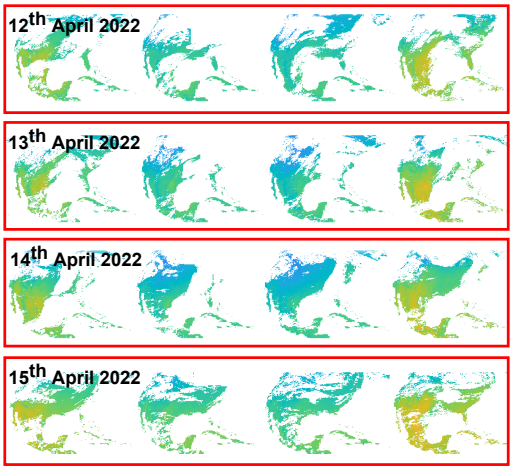
Countix *(Action Counting)*. The Countix dataset [43] is a subset of the Kinetics [62] dataset annotated with segments of repeated actions and corresponding counts. The creators crowdsourced the labels for repetition segments and counts for the selected classes. We further filter out videos that have a frame length shorter than 200, and make all videos have a fixed length of 200 frames. The resulting dataset has $1,712$ training samples, 457 validation samples, and 963 test samples, with a resolution of 96×96.

Land Surface Temperature (LST) *(Satellite Sensing)*. Land surface temperature is an indicator of the Earth surface energy budget and is widely required in applications of hydrology, meteorology and climatology. It is of fundamental importance to the net radiation budget at the Earth's surface and for monitoring the state of crops and vegetation, as well as an important indicator of both the greenhouse effect and the energy flux between the atmosphere and earth surface. We created a snapshot of data from the `NOAA GOES-16 Level 2`

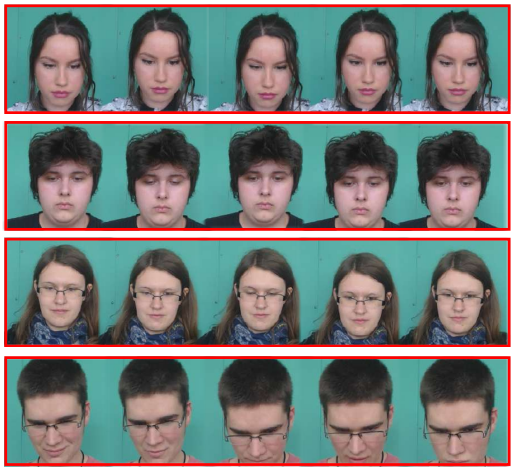


(a) RotatingDigits

(b) SCAMPS [59]



(c) Land Surface Temperature (LST)



(d) UBFC [60]

(e) PURE [61]

(f) Countix [43]

Figure A-1: **Examples of sequences from the datasets used in our experiments.**

Table A-1: **Detailed statistics of the datasets used in our experiments.**

	Targets	Sampling freq.	Frame length	# Training set	# Val. set	# Test set
RotatingDigits	Rotation frequency	30Hz	150	1,000	–	2,000
SCAMPS [59]	Heart rate	30Hz	600	2,000	400	400
UBFC [60]	Heart rate	30Hz	180	518	–	106
PURE [61]	Heart rate	30Hz	180	1,028	–	266
Countix [43]	Action counts	20∼30Hz	200	1,712	457	963
LST	Temperature	Hourly	100	276	–	92

`LST product` comprising of hourly land surfaces temperature outputs over the continental United States (CONUS). The LST measurements are sampled hourly over a 100 day period leading to $2,400$ LST maps at a resolution of $1,500 \times 2,500$. As the spatial resolution is high, we divide each map into four quarters (`North-West US`, `North-East US`, `South-West US`, and `South-East US`). We create each input sample using a window size of 100 frames with a step size of 24 (a day). The target signal is the temperature time series of the future 100 frames. The resulting dataset has 276 training samples and 92 test samples, with a spatial resolution of 100×100.

A.2 Experimental Settings

A.2.1 Competing Algorithms

We employ the following state-of-the-art SSL algorithms for comparisons.

SimCLR [36]. SimCLR learns feature representations by contrasting images with data augmentation. The positive pairs are constructed by sampling two images with different augmentations on one instance. The negative pairs are sampled from two different images.

MoCo v2 [37]. MoCo learns feature representations by building large dictionaries along with a contrastive loss. MoCo maintains the dictionary as a queue of data samples by enqueuing the current mini-batch and dequeuing the oldest mini-batch. The keys are encoded by a slowly progressing encoder with a momentum moving average and the query encoder.

BYOL [63]. BYOL leverages two neural networks to learn the feature representations: the online and target networks. The online network has an encoder, a projector, and a predictor

while the target network shares the same architecture but with a different set of weights. The online network is trained by the regression targets provided by the target network.

CVRL [39]. Contrastive Video Representation Learning (CVRL) is a self-supervised learning framework that learns spatial-temporal features representations from unlabelled videos. CVRL generates positive pairs by adding temporally consistent spatial augmentation on one videos clip and generate negative pairs by sampling two different video clips. The goal of contrastive loss is to minimize the embedding distance from the positive augmented video clips but maximize the distance from negative video clips.

■ A.2.2 Implementation Details

We describe the implementation details in this section. We first introduce parameters that are fixed to be the same across all methods, then detail the specific parameters for each dataset.

For all SSL methods, we follow the literature [36, 37] and apply the same standard data augmentations in contrastive learning. For temporal augmentations, we mainly employ `random_reverse` and `random_delay` (with shorter clip subsampling [39]). Unless specified, all augmentation hyper-parameters follow the original setup of each method.

RotatingDigits. On RotatingDigits, we adopt the network architecture as a simple 3D variant of the MNIST CNN used in [3, 133]. In the supervised setting, we train all models for 20 epochs using the Adam optimizer [334], with an initial learning rate of 10^{-3} and then decayed by 0.1 at the 12-th and 16-th epoch, respectively. We fix the batch size as 64 and use the checkpoint at the last epoch as the final model for evaluation. In the self-supervised setting, we train all models for 60 epochs, which ensures convergence for all tested algorithms. We again employ the Adam optimizer and decay the learning rating at the 40-th and 50-th epoch, respectively. Other training hyper-parameters remain unchanged.

SCAMPS. Similar to RotatingDigits, we employ the same 3D CNN architecture for all the SCAMPS experiments. In the supervised setting, we train all of the models for 30 epochs using the Adam optimizer, with an initial learning rate of 10^{-3} and then decayed by 0.1 at the 20-th and 25-th epoch, respectively. We fix the batch size as 32 and use the last checkpoint for final evaluation. In the self-supervised setting, we follow the same training regime in RotatingDigits as described in the previous section.

UBFC & PURE. Following [45, 48], we use the temporal shift convolution attention network (TS-CAN) as our backbone model. To adapt TS-CAN on `SimPer`, we remove the attention branch and make a variant of TS-CAN which only requires 3-channel as the input instead of 6-channel. In the supervised setting, we use the Adam optimizer, learning rate of 10^{-3} and train the network for a total of 10 epochs. On the inner-dataset evaluation (i.e., test and validation are the same), we use last epoch from the training of 80% for the dataset and evaluate the pre-trained model on the last 20% dataset. On the cross-dataset evaluation, we use 80% of the dataset for training and 20% for checkpoint selection then evaluate the pre-trained model on a different dataset. In the self-supervised setting, all other parameters remain unchanged except that we train for 60 epochs to ensure the SSL loss converges for all algorithms.

Countix. We use a ResNet-3D-18 [65, 64] architecture for all Countix experiments, which is widely used for video-based vision tasks. In the supervised setting, we train all models for 90 epochs using the Adam optimizer with an initial learning rate of 10^{-3} and then decayed by 0.1 at the 60-th and 80-th epoch. We fix the batch size as 32 for all experiments. In the self-supervised setting, we train all models for 200 epochs, and leave other parameters unchanged.

LST. Similar to Countix, we use the ResNet-3D-18 [65] network architecture for LST experiments. In the supervised setting, we train all models for 30 epochs using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 16. In the self-supervised setting, we train all models for 60 epochs while having other hyper-parameters the same for all methods.

■ A.2.3 Evaluation Metrics

We describe in detail all the evaluation metrics we used in our experiments.

MAE. The mean absolute error (MAE) is defined as $\frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y}_i|$, which represents the averaged absolute difference between the ground truth and predicted values over all samples.

MAPE. The mean absolute percentage error (MAPE) is defined as $\frac{1}{N}\sum_{i=1}^{N}|\frac{y_i-\widehat{y}_i}{y_i}|$, which assesses the averaged relative differences between the ground truth and predicted values over all samples.

GM. We use error Geometric Mean (**GM**) as another evaluation metric [2]. GM is defined as $(\prod_{i=1}^{N} e_i)^{\frac{1}{N}}$, where $e_i \triangleq |y_i - \widehat{y}_i|$ represents the L_1 error of each sample. GM aims to characterize the fairness (uniformity) of model predictions using the geometric mean instead of the arithmetic mean over the prediction errors.

Pearson correlation ρ. We employ Pearson correlation for performance evaluation on LST, where Pearson correlation evaluates the linear relationship between predictions and corresponding ground truth values.

A.3 Additional Results and Analysis

A.3.1 Data Efficiency w.r.t. Reduced Training Data

We provide quantitative results to verify the data efficiency of `SimPer` in the presence of reduced training data. Specifically, we use SCAMPS dataset, and vary the training dataset size from 100% to only 5%, and use it for both pre-training and fine-tuning. We show the final performance in Table A-2, where `SimPer` is able to achieve consistent performance gains compared to baselines when the dataset size varies. Furthermore, the gains are more significant when the dataset size is smaller (e.g., 5%), demonstrating that `SimPer` is particularly robust to reduced training data.

Table A-2: **Data efficiency w.r.t. reduced training data.** We vary the training dataset size of SCAMPS (size fixed for both pre-training and fine-tuning), and show the final fine-tuning performance of different methods.

Dataset size	100%		50%		20%		10%		5%	
Metrics	MAE↓	MAPE↓	MAE↓	MAPE↓	MAE↓	MAPE↓	MAE↓	MAPE↓	MAE↓	MAPE↓
SUPERVISED	3.61	5.33	3.85	5.60	4.57	7.16	7.13	10.08	12.24	15.42
SIMCLR [36]	4.96	6.92	6.55	9.39	6.01	9.25	7.63	10.19	13.75	15.72
CVRL [39]	5.52	7.34	3.66	5.64	4.86	7.77	7.08	9.45	14.11	15.91
SIMPER	**3.27**	**4.89**	**3.38**	**5.24**	**3.93**	**5.67**	**4.65**	**7.06**	**4.75**	**7.64**
GAINS VS. SUPERVISED	**+0.34**	**+0.44**	**+0.47**	**+0.36**	**+0.64**	**+1.49**	**+2.48**	**+3.02**	**+7.49**	**+7.78**

A.3.2 Amount of Labeled Data for Fine-tuning

We investigate the impact of the amount of labeled data for fine-tuning. Specifically, we use the whole training set of SCAMPS as the unlabeled dataset, and vary the labeled data

fraction for fine-tuning. As Table A-3 confirms, when the amount of labeled data is limited for fine-tuning, SimPer still substantially outperforms baselines by a large margin, achieving a 67% relative improvement in MAE even when the labeled data fraction is only 5%. The results again demonstrate that SimPer is data efficient in terms of the amount of labeled data available.

Table A-3: **Data efficiency w.r.t. amount of labeled data for fine-tuning.** We use all data from SCAMPS as unlabeled training set for self-supervised pre-training, and vary size of labeled data for fine-tuning.

Labeled data fraction	100%		50%		20%		10%		5%	
Metrics	MAE↓	MAPE↓	MAE↓	MAPE↓	MAE↓	MAPE↓	MAE↓	MAPE↓	MAE↓	MAPE↓
SUPERVISED	3.61	5.33	3.85	5.60	4.57	7.16	7.13	10.08	12.24	15.42
SIMCLR [36]	4.96	6.92	4.92	7.09	5.57	8.46	7.82	10.53	13.21	15.64
CVRL [39]	5.52	7.34	3.79	5.83	4.83	7.71	6.82	9.06	12.18	13.25
SIMPER	**3.27**	**4.89**	**3.32**	**5.13**	**3.58**	**5.44**	**3.98**	**5.81**	**4.02**	**6.27**
GAINS VS. SUPERVISED	**+0.34**	**+0.44**	**+0.53**	**+0.47**	**+0.99**	**+1.72**	**+3.15**	**+4.27**	**+8.22**	**+9.15**

A.3.3 Robustness to Spurious Correlations

We provide detailed quantitative results for the spurious correlations experiment in Section 2.3.5. Recall that SimCLR is easy to learn information that is spuriously correlated in the training data, and the learned representations do not generalize. Table A-4 further confirms the observation, where SimCLR achieves bad feature evaluation results with large MAE & MAPE errors.

In contrast, SimPer is able to learn the underlying frequency information even in the presence of strong spurious correlations, obtaining substantially smaller errors compared to SimCLR. The results demonstrate that SimPer is robust to spurious correlations, and can learn robust representations that generalize.

A.3.4 Ablation Studies for SimPer

In this section, we perform extensive ablation studies on SimPer to investigate the effect of different design choices as well as its hyper-parameter stability.

Table A-4: **Feature evaluation results on RotatingDigits with spurious correlations in training data.** Quantitative results in addition to Fig. 2-6 further verify that state-of-the-art SSL methods (e.g., SimCLR) are vulnerable to spurious correlations, and could easily learn information that is irrelevant to periodicity; In contrast, `SimPer` learns desirable periodic representations that are robust to spurious correlations.

	FFT		1-NN	
Metrics	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	MAPE$^{\downarrow}$
SimCLR [36]	3.06	125.48	1.49	80.28
SimPer	**0.36**	**15.04**	**0.78**	**27.03**
Gains	**+2.70**	**+110.44**	**+0.71**	**+53.25**

Range of Periodicity-Variant Frequency Augmentation

We study the effect of using different ranges of the variant speed augmentations in `SimPer`. We use the SCAMPS dataset, and vary the speed range during `SimPer` pre-training. As Table A-5 reports, using different speed ranges does not change the downstream performance by much, where all the results outperform the supervised baseline by a notable margin.

Table A-5: **Ablation study on the range of speed (frequency) augmentation.** Default settings used in the main experiments for `SimPer` are marked in gray.

Speed Range	$[0.5, 1.5]$	$[0.8, 1.8]$	$[0.5, 2]$	$[0.5, 3]$	Supervised
MAPE$^{\downarrow}$	4.97	4.92	4.89	4.98	5.33

Number of Periodicity-Variant Augmented Views

We study the effect of different number of periodicity-variant augmented views M on `SimPer`. We again employ the SCAMPS dataset, and vary the number of augmented views as $M \in \{3, 5, 10, 20\}$. Table A-6 shows the results, where we can observe a clear trend of decreased error rates when increasing M. Yet, when $M \geq 5$, the benefits of increasing M gradually diminish, indicating that a moderate M might be enough for the task. In the experiments of all tested datasets, to balance the efficiency while maintaining the contrastive ability, we set $M = 10$ by default.

Table A-6: **Ablation study on the number of periodicity-variant augmented views.** Default settings used in the main experiments for `SimPer` are marked in gray.

NUM. VIEWS	3	5	10	20	Supervised
MAPE$^{\downarrow}$	5.12	4.96	4.89	4.87	5.33

Table A-7: **Ablation study on the choices of different periodic similarity measures.** Default settings used in the main experiments for `SimPer` are marked in gray.

SIMILARITY METRICS	MXCorr	nPSD ($\cos(\cdot)$)	nPSD (L_2)	Supervised
MAPE$^{\downarrow}$	4.89	4.88	4.92	5.33

Choices of Different Similarity Metrics

We investigate the impact of different choices of periodic similarity measures introduced in Section 2.2.2. Specifically, we study three concrete instantiations of periodic similarity measures: **MXCorr**, **nPSD ($\cos(\cdot)$)**, and **nPSD (L_2)**. As Table A-7 shows, `SimPer` is robust to all aforementioned periodic similarity measures, achieving similar downstream performances. The results also demonstrate the effectiveness of the proposed similarity measures in periodic learning.

Effectiveness of the Generalized Contrastive Loss

We assess the effectiveness of the generalized contrastive loss, as compared to the classic InfoNCE contrastive loss. Table A-8 highlights the results over all six datasets, where consistent gains can be obtained when using the generalized contrastive loss in `SimPer` formulation.

Table A-8: **Ablation study on the effectiveness of using generalized contrastive loss in `SimPer`.** We show the feature evaluation results (FFT, MAE$^{\downarrow}$) with and without generalized contrastive loss across different datasets. Note that generalized contrastive loss with no continuity considered degenerates to InfoNCE [54].

	RotatingDigits	SCAMPS	UBFC	PURE	Countix	LST
`SimPer` (InfoNCE)	**0.23**	18.27	9.53	15.74	2.42	**4.84**
`SimPer` (Generalized)	**0.22**	**14.45**	**8.78**	**13.97**	**2.06**	**4.84**
Gains	**+0.01**	**+3.82**	**+0.75**	**+1.77**	**+0.36**	**+0.00**

Table A-9: **Ablation study on the input sequence lengths.** We show the fine-tune evaluation results (MAE$^{\downarrow}$) using different yet reasonable sequence lengths across various datasets.

	RotatingDigits			SCAMPS			LST		
# Frames	150	120	90	600	450	300	100	80	60
SUPERVISED	0.72	0.71	0.72	3.61	3.57	3.63	1.54	1.56	1.61
SIMPER	**0.20**	**0.19**	**0.20**	**3.27**	**3.11**	**3.12**	**1.47**	**1.47**	**1.48**
GAINS	**+0.52**	**+0.52**	**+0.52**	**+0.34**	**+0.46**	**+0.51**	**+0.07**	**+0.09**	**+0.13**

Choices of Different Input Sequence Lengths

Finally, we investigate the effect of different sequence lengths on the final performance in periodic learning. To make the observations more general and comprehensive, we choose three datasets from different domains (i.e., RotatingDigits, SCAMPS, and LST) to study the effect of sequence length. We fix all the experimental setups the same as in Appendix A.1 & A.2, and only vary the frame/sequence lengths with different yet reasonable choices for each dataset.

As highlighted from Table A-9, the results illustrate the following interesting observations:

- For "clean" periodic learning datasets with the periodic targets being the only dominating signal (i.e., RotatingDigits), using different frame lengths do not inherently change the final result.
- For dataset with relatively high SNR (i.e., LST), `SimPer` is also robust to different frame lengths. The supervised results however are worse with shorter clips, which could be attributed to the fact that less information is used in the input.
- Interestingly, for datasets where other periodic signals might exist (i.e., SCAMPS), using shorter (but with reasonable length) videos seems to slightly improve the performance of `SimPer`. We hypothesize that for a complex task such as video-based human physiological measurement, some videos may contain multiple periodic processes (e.g., PPG, breathing, blinking, etc.). A smaller frame length may not be enough to capture some of the "slow" periodic processes (e.g., breathing), thus the features learned by `SimPer` can become even more representative for PPG or heart beats estimation. Nevertheless, the differences between various choices are still small, indicating that `SimPer` is pretty robust to different frame lengths.

A.3.5 Compatibility with SOTA Supervised Learning Methods

As motivated, for each specific periodic learning application, supervised learning methods [43, 59, 335] have achieved remarkably good results via incorporating certain domain knowledge tailored for a specific task. Therefore, we provide additional results and comparisons using SOTA algorithms on each of the tested dataset. In the following, we show existing SOTA baselines and demonstrate that `SimPer` could further boost the performance when jointly applied.

Table A-10: **Compatibility of `SimPer` with SOTA supervised techniques across different datasets.** SOTA refers to RepNet [43] on Countix, and refers to EfficientPhys [335] on SCAMPS, UBFC & PURE. `SimPer` delivers robust performance and complements the performance of SOTA models.

	Countix		SCAMPS		UBFC		PURE	
Metrics	MAE$^{\downarrow}$	GM$^{\downarrow}$	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	MAPE$^{\downarrow}$
SOTA	1.03	0.41	2.42	4.10	4.14	3.79	2.87	2.89
SIMCLR + SOTA	1.06	0.43	2.56	4.17	4.31	4.02	2.94	3.25
SIMPER + SOTA	**0.72**	**0.22**	**1.96**	**3.45**	**3.27**	**3.06**	**2.29**	**2.21**
GAINS	**+0.29**	**+0.19**	**+0.46**	**+0.65**	**+0.87**	**+0.73**	**+0.58**	**+0.68**

Countix. In the video repetition counting domain, RepNet [43], a novel neural network architecture that composed of a ResNet-50 encoder and a Transformer based predictor, is proposed to achieve advanced results for repetitious counting in the wild. We verify the compatibility of `SimPer` with RepNet by changing the encoder on Countix to RepNet, and compare with the vanilla supervised training as well as SimCLR. To ensure a fair and comparable setting, we train RepNet from scratch instead of using ImageNet pre-trained ResNet-50 backbones as in the original paper [43].

SCAMPS, UBFC & PURE. In video-based human physiological measurement domain (i.e., SCAMPS, UBFC, and PURE), the main advances in the field have stemmed from better backbone architectures and network components [45, 335, 336]. In the main thesis, for SCAMPS, since it is a synthetic dataset, we employed a simple 3D ConvNet; as for real datasets UBFC and PURE, we used a more advanced backbone model [45]. To further demonstrate that `SimPer` can improve upon SOTA methods, we employ a recent architecture, called EfficientPhys [335], which is specialized for learning physiology from videos.

As confirmed in Table A-10, when jointly applied with SOTA models, `SimPer` can further boost the performance and consistently achieves the best results regardless of datasets and tasks. In contrast, SimCLR is not able to improve upon SOTA supervised learning techniques. The results indicate that `SimPer` is orthogonal to SOTA models for learning periodic targets.

Table A-11: **Comparisons between `SimPer` and additional SSL baselines on human physiological measurement datasets.** Compared to customized SSL algorithms in the specific domain, `SimPer` still delivers robust performance and consistently achieves the best results.

	SCAMPS		UBFC		PURE	
Metrics	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	MAPE$^{\downarrow}$
Without face saliency module:						
[336]	3.53	5.26	4.98	4.61	4.18	4.70
[337]	3.71	5.54	5.07	4.88	4.32	4.95
SIMPER	**3.27**	**4.89**	**4.24**	**3.97**	**3.89**	**4.01**
With face saliency module:						
[336]	3.51	5.15	4.88	4.29	4.03	4.28
[337]	3.61	5.40	5.02	4.86	4.07	4.33
SIMPER	**2.94**	**4.35**	**4.01**	**3.68**	**3.47**	**3.76**

Table A-12: **Comparisons between `SimPer` and additional SSL baselines on general periodic learning datasets other than human physiological measurement ones.** When extending to general periodic learning tasks, SSL baselines tailored for human physiological measurement [337, 336] no longer provide benefits, and sometimes perform even *worse* than the vanilla supervised learning. In contrast, `SimPer` consistently and substantially exhibits strengths in general periodic learning across all domains.

	RotatingDigits		Countix		LST	
Metrics	MAE$^{\downarrow}$	MAPE$^{\downarrow}$	MAE$^{\downarrow}$	GM$^{\downarrow}$	MAE$^{\downarrow}$	$\rho^{\uparrow}$
SUPERVISED	0.72	28.96	1.50	0.73	1.54	**0.96**
[336]	0.70	28.03	1.58	0.81	1.62	0.92
[337]	0.77	29.44	1.68	0.94	1.64	0.89
SIMPER	**0.20**	**14.33**	**1.33**	**0.59**	**1.47**	**0.96**

■ A.3.6 Comparisons to SSL Methods in Human Physiological Measurement

In video-based human physiological measurement domain, recent works [337, 336] have proposed to leverage contrastive SSL for better learned features and downstream perfor-

mance in the corresponding application (e.g., heart rate estimation). They studied specific SSL methods tailored for video-based human physiological measurement, and as a result, many of the proposed techniques therein only apply to that specific domain (e.g., the face detector, the saliency sampler, and the strong assumptions that are derived from the application context, cf. Table 1 in [336]). Nevertheless, it is possible to extend the SSL objectives therein to other general periodic learning domains. In this section, we provide additional experimental results and further discussions, which distinguish `SimPer` from these prior works.

Comparisons on the human physiological measurement task. We first compare `SimPer` against the aforementioned SSL methods [336, 337] on the human physiological measurement task. To provide a fair comparison, we fix all methods to use a simple 3D ConvNet backbone [133] on SCAMPS, and a TS-CAN backbone [45] on UBFC and PURE as stated in Appendix A.2. As Table A-11 demonstrates, `SimPer` outperforms these SSL baselines across all tested human physiology datasets by a notable margin. We break the results out to confirm that they hold regardless of whether we include the customized face saliency module [336] or not.

Comparisons on other periodic learning tasks. We further extend the comparisons to other general periodic learning tasks. We directly apply the SSL objectives in [336, 337] to other domains and datasets involving periodic learning, and show the corresponding results in Table A-12. The table clearly shows that the SSL objectives in the referenced papers do not provide a benefit in other periodic learning domains, and sometimes perform even *worse* than the vanilla supervised baseline. The above results further emphasize the significance of `SimPer`, which consistently and substantially exhibits strengths in general periodic learning across all domains.

■ A.3.7 Visualization of Learned Features

Since representations learned in periodic data naturally preserves the periodicity information, we can directly plot the learned 1-D features for visualization. Fig. A-2 shows the learned feature comparison between SimCLR, CVRL, and `SimPer`, together with the underlying periodic information (rotation angle & frequency) in RotatingDigits. As the figure verifies, `SimPer` consistently learns the periodic information with different frequency targets, delivering meaningful periodic representations that are robust and interpretable.

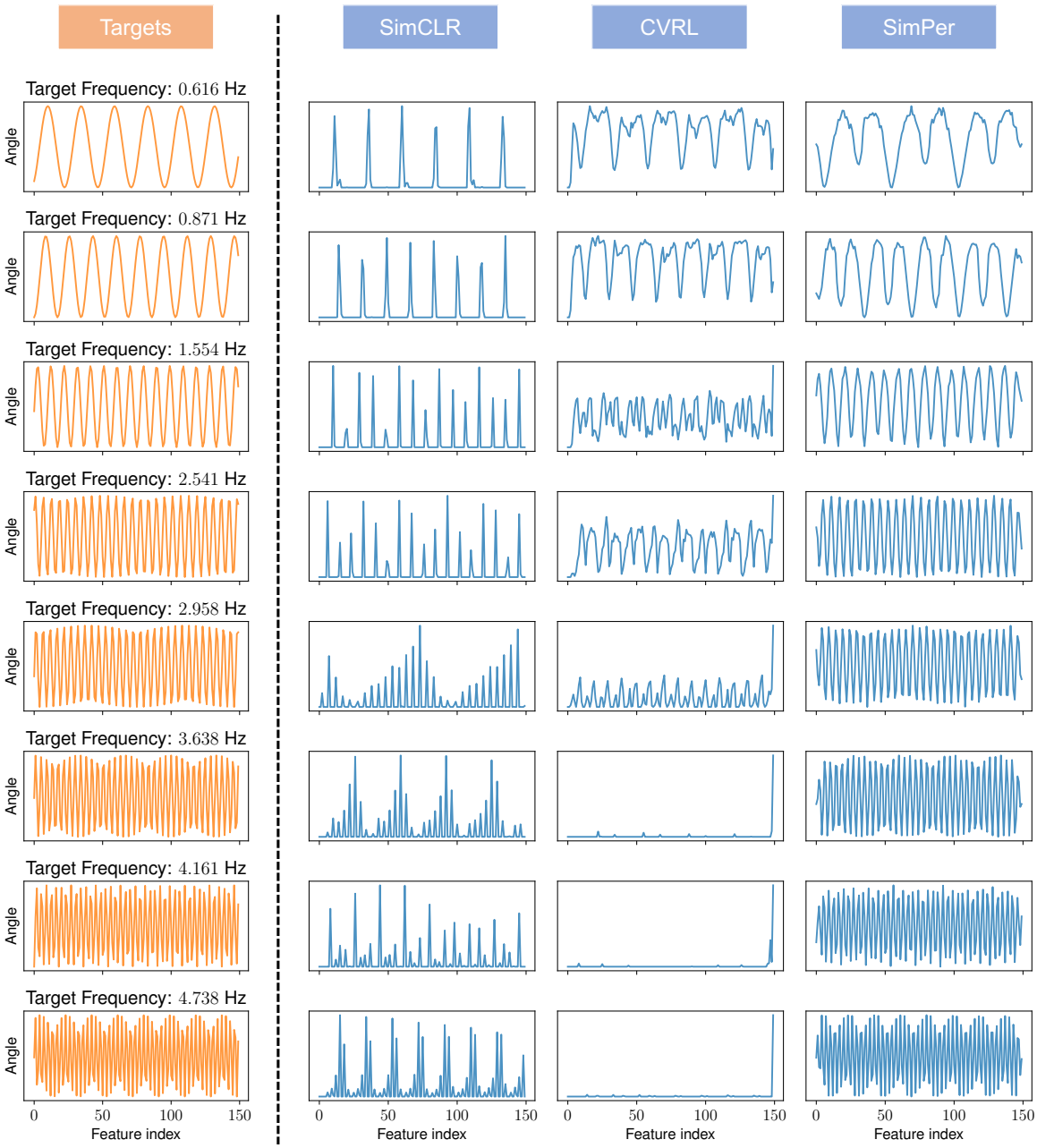


Figure A-2: **Visualization of learned periodic representations.** We directly plot the 1-D feature vector of data in the test set of RotatingDigits with different underlying target frequencies (left), via different self-supervised learning methods (right). Existing SSL solutions [39, 36] fail to learn meaningful periodic representations, whereas `SimPer` is able to capture the underlying periodicity information.

In contrast, existing SSL methods cannot capture the underlying periodicity, and fail to learn useful representations for periodic learning tasks.

APPENDIX B
Details and Results for Deep Imbalanced Regression

B.1 Details of DIR Datasets

In this section, we provide the detailed information of the five curated DIR datasets we used in our experiments. Table B-1 provides an overview of the five datasets.

B.1.1 IMDB-WIKI-DIR

The original IMDB-WIKI dataset [87] is a large-scale face image dataset for age estimation from single input image. The original version contains 523.0K face images and the corresponding ages, where 460.7K face images are collected from the IMDB website and 62.3K images from the Wikipedia website. We construct IMDB-WIKI-DIR by first filtering out unqualified images with low face scores [87], and then manually creating balanced validation and test set over the supported ages. Overall, the curated dataset has 191.5K images for training, and 11.0K images for validation and testing, respectively. We make the length of each bin to be 1 year, with a minimum age of 0 and a maximum age of 186. The number of images per bin varies between 1 and 7,149, exhibiting significant data imbalance.

As for the data pre-processing, the images are first resized to 224×224. During training, we follow the standard data augmentation scheme [64] to do zero-padding with 16 pixels on each side, and then random crop back to the original image size. We then randomly flip

Table B-1: **Overview of the five curated DIR datasets used in our experiments.**

Dataset	Target type	Target range	Bin size	Max bin density	Min bin density	# Training set	# Val. set	# Test set
IMDB-WIKI-DIR	Age	$0 \sim 186$	1	7,149	1	191,509	11,022	11,022
AgeDB-DIR	Age	$0 \sim 101$	1	353	1	12,208	2,140	2,140
STS-B-DIR	Text similarity score	$0 \sim 5$	0.1	428	1	5,249	1,000	1,000
NYUD2-DIR	Depth	$0.7 \sim 10$	0.1	1.46×10^8	1.13×10^6	50,688 (3.51×10^9)	–	654 (8.70×10^5)
SHHS-DIR	Health condition score	$0 \sim 100$	1	275	0	1,892	369	369

the images horizontally and normalize them into $[0, 1]$.

■ B.1.2 AgeDB-DIR

The original AgeDB dataset [90] is a manually collected in-the-wild age database with accurate and noise-free labels. Similar to IMDB-WIKI, the task is also to estimate age from visual appearance. The original dataset contains 16,488 images in total. We construct AgeDB-DIR in a similar manner as IMDB-WIKI-DIR, where the training set contains 12,208 images, with a minimum age of 0 and a maximum age of 101, and maximum bin density of 353 images and minimum bin density of 1. The validation set and test set are made balanced with 2,140 images. Similarly, the images in AgeDB are resized to 224×224, and go through the same data pre-processing schedule as in the IMDB-WIKI-DIR dataset.

■ B.1.3 STS-B-DIR

The original Semantic Textual Similarity Benchmark (STS-B) [91], also included in the GLUE benchmark [92], is a collection of sentence pairs drawn from news headlines, video and image captions, and natural language inference data. Each pair is human-annotated by multiple annotators with an averaged continuous similarity score from 0 to 5. The task is to predict these scores from the sentence pairs. From the original training set of 7.2K pairs, we create a training set with 5.2K pairs, and balanced validation set and test set of 1K pairs each for STS-B-DIR. We make the length of each bin to be 0.1, and the number of training pairs per bin varies between 1 and 428.

As for the data pre-processing, the sentences are first tokenized using NLTK toolkit [338] with a maximum length of 40. We then count the frequencies of all words (tokens) of all splits, build the word vocabulary based on the word frequency, and finally use the 300D GloVe word embeddings (840B Common Crawl version) [339] to embed words in the vocabulary into 300-dimensional vectors. Following [92], we use AllenNLP [340] open

source library to facilitate the data processing, as well as model training and evaluation.

B.1.4 NYUD2-DIR

We create NYUD2-DIR based on the NYU Depth Dataset V2 [93], which provides images and depth maps for different indoor scenes. Our task is to predict the depth maps from the RGB scene images. The depth maps have an upper bound of 10 meters and a lower bound of 0.7 meters. Following standard practices [94, 95], we use 50K images for training and 654 images for testing. We set the bin length to 0.1 meter and the number of pixels per bin varies between 1.13×10^6 and 1.46×10^8. Besides, we randomly select 9,357 test pixels (the minimum number of bin pixels in the test set) for each bin from 654 test images to make the test set balanced, with a total of 8.70×10^5 test pixels in the NYUD2-DIR test set, as indicated in Table B-1.

Following [94], for both training and evaluation phases, we first downsample images (both RGB and depth) from original size 640×480 to 320×240 using bilinear interpolation, then conduct center crop to obtain images of size 304×228, and finally normalize them into $[0, 1]$. Note that our pixel statistics are calculated and selected based on this resolution. For training, we further downsample the depth maps to 114×152 to fit the size of outputs. Additionally, we also employ the following data argumentation methods during training: (1) Flip: randomly flip both RGB and depth images horizontally with probability of 0.5; (2) Rotation: rotate both RGB and depth images by a random degree from -5 to 5; (3) Color Jitter: randomly scale the brightness, contrast, and saturation values of the RGB images by $c \in [0.6, 1.4]$.

B.1.5 SHHS-DIR

We create SHHS-DIR based on the SHHS dataset [96], which contains full-night Polysomnography (PSG) signals from 2,651 subjects. The signal length for each subject varies from 7,278 seconds to 45,448 seconds. Available PSG signals include Electroencephalography (EEG), Electrocardiography (ECG), and breathing signals (airflow, abdomen, and thorax). In the experiments, we consider all of these PSG signals as high-dimensional information, and use them as inputs. Specifically, we first preprocess both EEG and ECG signals to transform them from time domain to the frequency domain using the short-time Fourier transform (STFT), and get the dense EEG spectrograms $\mathbf{x}_e \in \mathbb{R}^{64 \times l_i}$ and ECG spectrograms

$\mathbf{x}_c \in \mathbb{R}^{22\times l_i}$, where $l_i \in [7278, 45448]$ is the signal length for the i-th subject. For the breathing signals, we use the original time series with a sampling rate of 10Hz, resulting in the high-dimensional input as $\mathbf{x}_b \in \mathbb{R}^{3\times 10l_i}$, where the three different breathing sources are concatenated as different channels.

The dataset also includes the 36-Item Short Form Health Survey (SF-36) [97] for each subject, where a General Health score is extracted. We use the score as the target value, and formulate the task as predicting the General Health score for different subjects from their PSG signals (i.e., $\mathbf{x}_e, \mathbf{x}_c, \mathbf{x}_b$). The training set of SHHS-DIR contains 1,892 samples (subjects), and the validation set and test set are made balanced over the health score with 369 samples each. We set the length of each bin to be 1, with a minimum score of 0 and a maximum score of 100. The number of samples per bin varies between 0 and 275, indicating the missing data issue in certain target bins.

B.2 Experimental Settings

B.2.1 Implementation Details

IMDB-WIKI-DIR & AgeDB-DIR. We use ResNet-50 model [64] for all IMDB-WIKI-DIR and AgeDB-DIR experiments. We train all models for 90 epochs using the Adam optimizer [334], with an initial learning rate of 10^{-3} and then decayed by 0.1 at the 60-th and 80-th epoch, respectively. We mainly employ the L_1 loss throughout the experiments, and fix the batch size as 256.

For both LDS and FDS, we use the Gaussian kernel for distribution smoothing, with the kernel size $l = 5$ and the standard deviation $\sigma = 2$. We study different choices of kernel types, training losses, and hyper-parameter values in Sec. B.4.1, B.4.2, and B.4.3. For the implementation of FDS, we simply use the feature variance instead of covariance for better computational efficiency. The momentum of FDS is fixed as 0.9. As for the baseline methods, we set $\beta = 0.2$ and $\gamma = 1$ for FOCAL-R. For RRT, in the second training stage, we employ an initial learning rate of 10^{-4} with total training epochs of 30. For SMOTER and SMOGN, we divide the target range based on a manually defined relevance method, under-sample majority regions, and over-sample minority regions by either interpolating with selected nearest neighbors [83] or also adding Gaussian noise perturbation [84]. We use pixel-wise Euclidean distance to define the image distance, which is further used to

determine nearest neighbors, and set Gaussian perturbation ratio as 0.1 for SMOGN.

STS-B-DIR. Following [92], we use 300D GloVe word embeddings (840B Common Crawl version) [339] and a two-layer, 1500D (per direction) BiLSTM with max pooling to encode the paired sentences into independent vectors u and v, and then pass $[u; v; |u - v|; uv]$ to a regressor. We train all models using the Adam optimizer with a fixed learning rate 10^{-4}. We validate the model every 10 epochs, use MSE as the validation metric, and stop training when performance does not improve, i.e., validation error does not decrease, after 10 validation checks. We employ the MSE loss throughout the experiments and fix the batch size as 128.

We use the same hyper-parameter settings for both LDS and FDS as in the IMDB-WIKI-DIR experiments. For the baselines, we employ MSE-based FOCAL-R and set $\beta = 20$ and $\gamma = 1$. For RRT, the hyper-parameter settings remain the same between the first and the second training stage. For SMOTER and SMOGN, we use the Euclidean distance between the word embeddings to measure the sentence distance and do interpolation or Gaussian noise argumentation based on the word embeddings. We set Gaussian perturbation ratio as 0.1 and the number of neighbors $k = 7$. For STS-B-DIR, we define *many-shot region* as bins with over 100 training samples, *medium-shot region* with 30$\sim$100 training samples, and *few-shot region* with under 30 training samples.

NYUD2-DIR. We use ResNet-50-based encoder-decoder architecture proposed by [94] for all NYUD2-DIR experiments, which consists of an encoder, a decoder, a multi-scale feature fusion module, and a refinement module. We train all models for 20 epochs using Adam optimizer with an initial learning rate of 10^{-4} and then decayed by 0.1 every 5 epochs. To better evaluate the performance of our methods, we simply use the MSE loss as the depth loss without adding the gradient and surface normal losses as in [94]. We fix the batch size as 32 for all experiments. We use the same hyper-parameter settings for both LDS and FDS as in the IMDB-WIKI-DIR experiments. For NYUD2-DIR, *many-shot region* is defined as bins with over 2.6×10^7 training pixels, *medium-shot region* as bins with $1.0 \times 10^7 \sim 2.6 \times 10^7$ training pixels, and *few-shot region* as bins with under 1.0×10^7 training pixels.

SHHS-DIR. Following [98], we use a CNN-RNN network architecture for SHHS-DIR experiments. The network first employs three encoders with the same architecture to encode the high-dimensional EEG $\mathbf{x}_e$, ECG $\mathbf{x}_c$, and breathing signals $\mathbf{x}_b$ into fixed-length vectors

(each with 256 dimensions). The encodings are then concatenated and sent to a 3-layer MLP regression network to produce the output value. Each of the encoder uses the ResNet block [64] with 1D convolution as the CNN components, and employs the simple recurrent units (SRU) [223] as the RNN components. We train all models for 80 epochs using the Adam optimizer with a learning rate of 10^{-3}, and remain all other hyper-parameters the same as [98]. We use the same hyper-parameter settings for both LDS and FDS, as well as other baseline methods as in the IMDB-WIKI-DIR experiments.

■ B.2.2 Evaluation Metrics

We describe in detail all the evaluation metrics we used in our experiments.

MAE. The mean absolute error (MAE) is defined as $\frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y}_i|$, which represents the averaged absolute difference between the ground truth and predicted values over all samples.

MSE & RMSE. The mean squared error (MSE) is defined as $\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2$, which represents the averaged squared difference between the ground truth and predicted values over all samples. The root mean squared error (RMSE) is computed by simply taking the square root of MSE.

GM. We propose another evaluation metric for regression, called error Geometric Mean (**GM**), and is defined as $(\prod_{i=1}^{N} e_i)^{\frac{1}{N}}$, where $e_i \triangleq |y_i - \widehat{y}_i|$ represents the L_1 error of each sample. GM aims to characterize the fairness (uniformity) of model predictions using the geometric mean instead of the arithmetic mean over the prediction errors.

Pearson correlation & Spearman correlation. Following the common evaluation practice as in the STS-B [91] and the GLUE benchmark [92], we employ Pearson correlation as well as Spearman correlation for performance evaluation on STS-B-DIR, where Pearson correlation evaluates the linear relationship between predictions and corresponding ground truth values, and Spearman correlation evaluates the monotonic rank-order relationship.

Mean $\log_{10}$ error & Threshold accuracy. For NYUD2-DIR, we further use several standard depth estimation evaluation metrics proposed by [341]: Mean $\log_{10}$ error ($\log_{10}$), which is expressed as $\frac{1}{N}\sum_{i=1}^{N}|\log_{10} d_i - \log_{10} g_i|$; Threshold accuracy ($\delta_i$), which is defined as the percentage of d_i such that $\max\left(\frac{d_i}{g_i}, \frac{g_i}{d_i}\right) = \delta_i < 1.25^i$ ($i = 1, 2, 3$). Here, g_i denotes the value of a pixel in the ground truth depth image, d_i represents the value of its corresponding

Table B-2: **Complete evaluation results on IMDB-WIKI-DIR.**

Metrics	MSE ↓				MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	138.06	108.70	366.09	964.92	8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
VANILLA + **LDS**	131.65	109.04	**298.98**	829.35	7.83	7.31	**12.43**	22.51	4.42	4.19	**7.00**	13.94
VANILLA + **FDS**	133.81	107.51	332.90	916.18	7.85	**7.18**	13.35	24.12	4.47	4.18	8.18	15.18
VANILLA + **LDS** + **FDS**	**129.35**	**106.52**	311.49	**811.82**	**7.78**	7.20	12.61	**22.19**	**4.37**	**4.12**	7.39	**12.61**
MIXUP [187]	141.11	109.13	389.95	1037.98	8.22	**7.29**	16.23	28.11	4.68	**4.22**	12.28	23.55
M-MIXUP [342]	137.45	**108.33**	363.72	957.53	8.22	7.39	15.24	26.70	4.80	4.39	10.85	21.86
SMOTER [83]	138.75	111.55	346.09	935.89	8.14	7.42	14.15	25.28	4.64	4.30	9.05	19.46
SMOGN [84]	136.09	109.15	339.09	944.20	8.03	7.30	14.02	25.93	4.63	4.30	8.74	20.12
SMOGN + **LDS**	137.31	111.79	333.15	823.07	8.02	7.39	13.71	23.22	4.63	4.39	8.71	15.80
SMOGN + **FDS**	137.82	109.42	340.65	847.96	8.03	7.35	14.06	23.44	4.65	4.33	8.87	16.00
SMOGN + **LDS** + **FDS**	**135.26**	110.91	**326.52**	**808.45**	**7.97**	7.38	**13.22**	**22.95**	**4.59**	4.39	**7.84**	**14.94**
FOCAL-R	136.98	106.87	368.60	1002.90	7.97	7.12	15.14	26.96	4.49	4.10	10.37	21.20
FOCAL-R + **LDS**	132.81	105.62	354.37	949.03	7.90	**7.10**	14.72	25.84	**4.47**	**4.09**	10.11	19.14
FOCAL-R + **FDS**	133.74	105.35	351.00	958.91	7.96	7.14	14.71	26.06	4.51	4.12	10.16	19.56
FOCAL-R + **LDS** + **FDS**	**132.58**	**105.33**	**338.65**	**944.92**	**7.88**	**7.10**	**14.08**	**25.75**	**4.47**	4.11	**9.32**	**18.67**
RRT	132.99	105.73	341.36	928.26	7.81	7.07	14.06	25.13	4.35	4.03	8.91	16.96
RRT + **LDS**	132.91	105.97	338.98	916.98	7.79	7.08	13.76	24.64	4.34	**4.02**	8.72	16.92
RRT + **FDS**	129.88	**104.63**	310.69	890.04	**7.65**	**7.02**	12.68	23.85	**4.31**	4.03	7.58	16.28
RRT + **LDS** + **FDS**	**129.14**	105.92	**306.69**	**880.13**	**7.65**	7.06	**12.41**	**23.51**	**4.31**	4.07	**7.17**	**15.44**
INV	139.48	116.72	305.19	869.50	8.17	7.64	12.46	22.83	4.70	4.51	6.94	13.78
SQINV	134.36	111.23	308.63	834.08	7.87	7.24	12.44	22.76	4.47	4.22	7.25	15.10
SQINV + **LDS**	131.65	109.04	**298.98**	829.35	7.83	7.31	**12.43**	22.51	4.42	4.19	7.00	13.94
SQINV + **FDS**	132.64	109.28	311.35	851.06	7.83	7.23	12.60	22.37	4.42	4.20	**6.93**	13.48
SQINV + **LDS** + **FDS**	**129.35**	**106.52**	311.49	**811.82**	**7.78**	**7.20**	12.61	**22.19**	**4.37**	**4.12**	7.39	**12.61**
OURS (BEST) VS. VANILLA	**+8.92**	**+4.07**	**+67.11**	**+156.47**	**+0.41**	**+0.21**	**+2.71**	**+4.14**	**+0.26**	**+0.15**	**+3.66**	**+7.85**

pixel in the predicted depth image, and N is the total number of evaluation pixels.

B.3 Additional Results

We provide complete evaluation results on the five DIR datasets, where more baselines and evaluation metrics are included in addition to the reported results in the main thesis.

B.3.1 Complete Results on IMDB-WIKI-DIR

We include more baseline methods for comparison on IMDB-WIKI-DIR. Specifically, the following two baselines are added for comparison in the group of *Synthetic samples* strategies:

- **Mixup** [187]: MIXUP trains a deep model using samples created by the convex combinations of pairs of inputs and corresponding labels. It has shown promising results on improving the generalization of deep models as a regularization technique.

Table B-3: **Complete evaluation results on AgeDB-DIR.**

Metrics	MSE ↓				MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	101.60	78.40	138.52	253.74	7.77	6.62	9.55	13.67	5.05	4.23	7.01	10.75
VANILLA + **LDS**	102.22	83.62	128.73	**204.64**	7.67	6.98	8.86	10.89	4.85	4.39	5.80	7.45
VANILLA + **FDS**	**98.55**	**75.06**	123.58	235.70	**7.55**	**6.50**	8.97	13.01	4.75	**4.03**	6.42	9.93
VANILLA + **LDS** + **FDS**	99.46	84.10	**112.20**	209.27	**7.55**	7.01	**8.24**	**10.79**	**4.72**	4.36	**5.45**	**6.79**
SMOTER [83]	114.34	93.35	129.89	244.57	8.16	7.39	8.65	12.28	5.21	4.65	5.69	8.49
SMOGN [84]	117.29	101.36	133.86	232.90	8.26	7.64	9.01	12.09	5.36	4.90	6.19	8.44
SMOGN + **LDS**	110.43	93.73	124.19	229.35	7.96	7.44	8.64	11.77	5.03	4.68	5.69	7.98
SMOGN + **FDS**	112.42	97.68	131.37	233.30	8.06	7.52	8.75	11.89	5.02	4.66	5.63	8.02
SMOGN + **LDS** + **FDS**	**108.41**	**91.58**	**120.28**	**218.59**	**7.90**	**7.32**	**8.51**	**11.19**	**4.98**	**4.64**	**5.41**	**7.35**
FOCAL-R	101.26	77.03	131.81	252.47	7.64	6.68	9.22	13.00	4.90	4.26	6.39	9.52
FOCAL-R + **LDS**	98.80	77.14	125.53	229.36	7.56	**6.67**	8.82	12.40	4.82	4.27	5.87	8.83
FOCAL-R + **FDS**	100.14	80.97	121.84	**221.15**	7.65	6.89	8.70	**11.92**	4.83	4.32	5.89	**8.04**
FOCAL-R + **LDS** + **FDS**	**96.70**	**76.11**	**115.86**	238.25	**7.47**	6.69	**8.30**	12.55	**4.71**	**4.25**	**5.36**	8.59
RRT	102.89	83.37	125.66	224.27	7.74	6.98	8.79	11.99	5.00	4.50	5.88	8.63
RRT + **LDS**	102.63	83.93	126.01	214.66	7.72	7.00	8.75	11.62	4.98	4.54	5.71	8.27
RRT + **FDS**	102.09	84.49	122.89	224.05	7.70	**6.95**	8.76	11.86	4.82	**4.32**	5.83	8.08
RRT + **LDS** + **FDS**	**101.74**	**83.12**	**121.08**	**210.78**	**7.66**	6.99	**8.60**	**11.32**	**4.80**	4.42	**5.53**	**6.99**
INV	110.24	91.93	130.68	211.92	7.97	7.31	8.81	11.62	5.05	4.64	5.75	8.20
SQINV	105.14	87.21	127.66	212.30	7.81	7.16	8.80	11.20	4.99	4.57	5.73	7.77
SQINV + **LDS**	102.22	**83.62**	128.73	204.64	7.67	**6.98**	8.86	10.89	4.85	4.39	5.80	7.45
SQINV + **FDS**	101.67	86.49	129.61	**167.75**	7.69	7.10	8.86	**9.98**	4.83	4.41	5.97	**6.29**
SQINV + **LDS** + **FDS**	**99.46**	84.10	**112.20**	209.27	**7.55**	7.01	**8.24**	10.79	**4.72**	**4.36**	**5.45**	6.79
OURS (BEST) VS. VANILLA	**+4.90**	**+3.34**	**+26.32**	**+85.99**	**+0.30**	**+0.12**	**+1.31**	**+3.69**	**+0.34**	**+0.20**	**+1.65**	**+4.46**

- **Manifold-Mixup** (M-MIXUP) [342]: M-MIXUP extends the idea of MIXUP from input space to the hidden representation space, where the linear interpolations are performed in (multiple) deep hidden layers.

We note that both MIXUP and M-MIXUP are not tailored for imbalanced regression problems, but share similarities with SMOTER and SMOGN as synthetic samples are constructed. The differences lie in the fact that MIXUP and M-MIXUP create virtual samples (either in input space or feature space) on the fly during network training, while SMOTER and SMOGN operate on a newly generated and fixed dataset for training. We set $\alpha = 0.2$ for MIXUP in implementation, and set $\alpha = 0.2$ as well and eligible layers $\mathcal{S} = \{0, 1, 2, 3\}$ for M-MIXUP. In addition, for INV which re-weights the loss based on the inverse frequency in the empirical label distribution, we further clip the maximum weight to be at most $200\times$ larger than the minimum weight to avoid extreme loss values.

We show the complete results in Table B-2. As the table illustrates, both MIXUP and M-MIXUP can improve the performance in the many-shot region, but lead to negligible improvements in the medium-shot and few-shot regions. In contrast, adding both FDS and LDS can substantially improve the results, especially for the underrepresented regions.

Table B-4: **Complete evaluation results on STS-B-DIR.**

Metrics	MSE ↓				MAE ↓				Pearson correlation (%) ↑				Spearman correlation (%) ↑			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	0.974	0.851	1.520	0.984	0.794	0.740	1.043	0.771	74.2	72.0	62.7	75.2	74.4	68.8	50.5	**75.0**
VANILLA + **LDS**	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	**55.6**	74.3
VANILLA + **FDS**	0.916	0.875	**1.027**	1.086	0.767	0.746	**0.840**	0.811	75.5	73.0	**67.0**	72.8	75.8	69.9	54.4	72.0
VANILLA + **LDS** + **FDS**	**0.907**	**0.802**	1.363	**0.942**	**0.766**	**0.718**	0.986	**0.755**	**76.0**	**74.0**	65.2	**76.6**	**76.4**	**70.7**	54.9	74.9
SMOTER [83]	1.046	0.924	1.542	1.154	0.834	0.782	1.052	0.861	72.6	69.3	65.3	70.6	72.6	65.6	**55.6**	69.1
SMOGN [84]	0.990	0.896	1.327	1.175	0.798	0.755	0.967	0.848	73.2	70.4	65.5	69.2	73.2	67.0	55.1	67.0
SMOGN + **LDS**	0.962	0.880	1.242	1.155	0.787	0.748	0.944	0.837	74.0	71.5	65.2	69.8	74.3	68.5	53.6	67.1
SMOGN + **FDS**	0.987	0.945	**1.101**	1.153	0.796	0.776	**0.864**	0.838	73.0	69.6	**68.5**	69.9	72.9	66.0	54.3	68.0
SMOGN + **LDS** + **FDS**	**0.950**	**0.851**	1.327	**1.095**	**0.785**	**0.738**	0.987	**0.799**	**74.6**	**72.1**	65.9	**71.7**	**75.0**	**68.9**	54.4	**70.3**
FOCAL-R	0.951	0.843	1.425	0.957	0.790	0.739	1.028	0.759	74.6	72.3	61.8	76.4	75.0	69.4	51.9	75.5
FOCAL-R + **LDS**	0.930	**0.807**	1.449	0.993	0.781	**0.723**	1.031	0.801	**75.7**	**73.9**	62.4	75.4	**76.2**	**71.2**	50.7	74.7
FOCAL-R + **FDS**	**0.920**	0.855	**1.169**	1.008	**0.775**	0.743	**0.903**	0.804	75.1	72.6	**66.4**	74.7	75.4	69.4	**52.7**	75.4
FOCAL-R + **LDS** + **FDS**	0.940	0.849	1.358	**0.916**	0.785	0.737	0.984	**0.732**	74.9	72.2	66.3	**77.3**	75.1	69.2	52.5	**76.4**
RRT	0.964	0.842	1.503	0.978	0.793	0.739	1.044	0.768	74.5	72.4	62.3	75.4	74.7	69.2	51.3	**74.7**
RRT + **LDS**	0.916	0.817	1.344	0.945	0.772	0.727	0.980	**0.756**	75.7	73.5	64.1	76.6	76.1	70.4	53.2	74.2
RRT + **FDS**	0.929	0.857	**1.209**	1.025	0.769	0.736	**0.905**	0.795	74.9	72.1	**67.2**	74.0	75.0	69.1	52.8	74.6
RRT + **LDS** + **FDS**	**0.903**	**0.806**	1.323	**0.936**	**0.764**	**0.719**	0.965	0.760	**76.0**	**73.8**	65.2	**76.7**	**76.4**	**70.8**	**54.7**	**74.7**
INV	1.005	0.894	1.482	1.046	0.805	0.761	1.016	0.780	72.8	70.3	62.5	73.2	73.1	67.2	54.1	71.4
INV + **LDS**	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	**55.6**	74.3
INV + **FDS**	0.927	0.851	**1.225**	1.012	0.771	0.740	**0.914**	0.756	75.0	72.4	**66.6**	74.2	75.2	69.2	55.2	74.8
INV + **LDS** + **FDS**	**0.907**	**0.802**	1.363	**0.942**	**0.766**	**0.718**	0.986	**0.755**	**76.0**	**74.0**	65.2	**76.6**	**76.4**	**70.7**	54.9	**74.9**
OURS (BEST) VS. VANILLA	**+.071**	**+.049**	**+.419**	**+.068**	**+.030**	**+.022**	**+.203**	**+.039**	**+1.8**	**+2.0**	**+5.8**	**+2.1**	**+2.0**	**+2.4**	**+5.1**	**+1.4**

Table B-5: **Complete evaluation results on NYUD2-DIR.**

Metrics	RMSE ↓				$\log_{10}$ ↓				δ_1 ↑				δ_2 ↑				δ_3 ↑			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	1.477	0.591	0.952	2.123	0.086	0.066	0.082	0.107	0.677	0.777	0.693	0.570	0.899	0.956	0.906	0.840	0.969	0.990	0.975	0.946
VANILLA + **LDS**	1.387	0.671	0.913	1.954	0.086	0.079	0.079	0.097	0.672	0.701	0.706	0.630	0.907	0.932	0.929	0.875	0.976	0.984	0.982	0.964
VANILLA + **FDS**	1.442	**0.615**	0.940	2.059	0.084	**0.069**	0.080	0.101	0.681	**0.760**	0.695	0.596	0.903	**0.952**	0.918	0.849	0.975	**0.989**	0.976	0.960
VANILLA + **LDS** + **FDS**	**1.338**	0.670	**0.851**	**1.880**	**0.080**	0.074	**0.070**	**0.090**	**0.705**	0.730	**0.764**	**0.655**	**0.916**	0.939	**0.941**	**0.884**	**0.979**	0.984	**0.983**	**0.971**
OURS (BEST) VS. VANILLA	**+.139**	**-.024**	**+.101**	**+.243**	**+.006**	**-.003**	**+.012**	**+.017**	**+.028**	**-.017**	**+.071**	**+.085**	**+.017**	**-.004**	**+.035**	**+.044**	**+.010**	**-.001**	**+.008**	**+.025**

Finally, FDS and LDS lead to remarkable improvements when compared to the VANILLA model across all evaluation metrics.

■ B.3.2 Complete Results on AgeDB-DIR

We provide complete evaluation results for AgeDB-DIR in Table B-3. Similar to IMDB-WIKI-DIR, within each group of techniques, adding either LDS, FDS, or both can lead to performance gains, while LDS + FDS often achieves the best results. Overall, for different groups of strategies, both FDS and LDS consistently boost the performance, where the larger gains come from the medium-shot and few-shot regions.

Table B-6: **Complete evaluation results on SHHS-DIR.**

Metrics	MSE ↓				MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	369.18	269.37	311.45	417.31	15.36	12.47	13.98	16.94	10.63	8.04	9.59	12.20
VANILLA + **LDS**	309.19	220.87	252.53	394.91	14.14	11.66	12.77	16.05	9.26	7.64	8.18	11.32
VANILLA + **FDS**	303.82	214.63	267.08	386.75	13.84	11.13	12.72	15.95	8.89	**6.93**	8.05	11.19
VANILLA + **LDS** + **FDS**	**292.18**	**211.89**	**247.48**	**346.01**	**13.76**	**11.12**	**12.18**	**15.07**	**8.70**	6.94	**7.60**	**10.18**
FOCAL-R	345.44	219.75	309.01	430.26	14.67	11.70	13.69	17.06	9.98	7.93	8.85	11.95
FOCAL-R + **LDS**	317.39	242.18	270.04	411.73	14.49	12.01	12.43	16.57	9.98	7.89	8.59	11.40
FOCAL-R + **FDS**	310.94	**185.16**	303.90	391.22	14.18	**11.06**	13.56	15.99	9.45	**6.95**	8.81	11.13
FOCAL-R + **LDS** + **FDS**	**297.85**	193.42	**259.33**	**375.16**	**14.02**	11.08	**12.24**	**15.49**	**9.32**	7.18	**8.10**	**10.39**
RRT	354.75	274.01	308.83	408.47	14.78	12.43	14.01	16.48	10.12	8.05	9.71	11.96
RRT + **LDS**	344.18	245.39	304.32	402.56	14.56	12.08	13.44	16.45	9.89	7.85	9.18	11.82
RRT + **FDS**	328.66	239.83	298.71	397.25	14.36	11.97	13.33	16.08	9.74	7.54	9.20	11.31
RRT + **LDS** + **FDS**	**313.58**	**238.07**	**276.50**	**380.64**	**14.33**	**11.96**	**12.47**	**15.92**	**9.63**	**7.35**	**8.74**	**11.17**
INV	322.17	231.68	293.43	387.48	14.39	11.84	13.12	16.02	9.34	7.73	8.49	11.20
INV + **LDS**	309.19	220.87	252.53	394.91	14.14	11.66	12.77	16.05	9.26	7.64	8.18	11.32
INV + **FDS**	307.95	219.36	247.55	361.29	13.91	**11.12**	12.29	15.53	8.94	**6.91**	7.79	10.65
INV + **LDS** + **FDS**	**292.18**	**211.89**	**247.48**	**346.01**	**13.76**	**11.12**	**12.18**	**15.07**	**8.70**	6.94	**7.60**	**10.18**
OURS (BEST) VS. VANILLA	+77.00	+84.21	+63.97	+71.30	+1.60	+1.41	+1.80	+1.87	+1.93	+1.13	+1.99	+2.02

■ B.3.3 Complete Results on STS-B-DIR

We present complete results on STS-B-DIR in Table B-4, where more metrics, such as MAE and Spearman correlation are added for further evaluation. In summary, across all the metrics used, by adding LDS and FDS we can substantially improve the results, particularly for the medium-shot and few-shot regions. The advantage is even more profound under *Pearson correlation*, which is commonly used for this task.

■ B.3.4 Complete Results on NYUD2-DIR

Table B-5 shows the complete evaluation results on NYUD2-DIR. As described before, we further add common metrics for depth estimation evaluation, including $\log_{10}$, δ_1, δ_2, and δ_3. The table reveals the following results. First, either FDS or LDS alone can improve the overall depth regression results, where LDS is more effective for improving performance in the few-shot region. Furthermore, when combined together, LDS & FDS can alleviate the overfitting phenomenon to many-shot regions of the vanilla model, and generalize better to all regions.

■ B.3.5 Complete Results on SHHS-DIR

We report the complete results on SHHS-DIR in Table B-6. The results again confirm the effectiveness of both LDS and FDS beyond the success on typical image data and text data,

Table B-7: **Ablation study of different kernel types for LDS & FDS on IMDB-WIKI-DIR.**

Metrics	MSE ↓				MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	138.06	108.70	366.09	964.92	8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
LDS:												
GAUSSIAN KERNEL	131.65	109.04	298.98	834.08	7.83	7.31	12.43	22.51	4.42	4.19	7.00	13.94
TRIANGULAR KERNEL	133.77	110.24	309.70	850.74	7.89	7.30	12.72	22.80	4.50	4.24	7.75	14.91
LAPLACIAN KERNEL	132.87	109.27	312.10	829.83	7.87	7.29	12.68	22.38	4.50	4.26	7.29	13.71
FDS:												
GAUSSIAN KERNEL	133.81	107.51	332.90	916.18	7.85	7.18	13.35	24.12	4.47	4.18	8.18	15.18
TRIANGULAR KERNEL	134.09	110.49	301.18	927.99	7.97	7.41	12.20	23.99	4.64	4.41	7.06	14.28
LAPLACIAN KERNEL	133.00	104.26	352.95	968.62	8.05	7.25	14.78	26.16	4.71	4.33	10.19	19.09

Table B-8: **Ablation study of different kernel types for LDS & FDS on STS-B-DIR.**

Metrics	MSE ↓				MAE ↓				Pearson correlation (%) ↑				Spearman correlation (%) ↑			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA	0.974	0.851	1.520	0.984	0.794	0.740	1.043	0.771	74.2	72.0	62.7	75.2	74.4	68.8	50.5	75.0
LDS:																
GAUSSIAN KERNEL	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	55.6	74.3
TRIANGULAR KERNEL	0.938	0.870	1.193	1.039	0.786	0.754	0.929	0.784	74.8	72.4	64.1	74.0	75.2	69.3	54.1	73.9
LAPLACIAN KERNEL	0.938	0.829	1.413	0.962	0.782	0.731	1.014	0.773	75.7	73.0	65.8	76.5	76.0	70.0	52.3	75.2
FDS:																
GAUSSIAN KERNEL	0.916	0.875	1.027	1.086	0.767	0.746	0.840	0.811	75.5	73.0	67.0	72.8	75.8	69.9	54.4	72.0
TRIANGULAR KERNEL	0.935	0.863	1.239	0.966	0.762	0.725	0.912	0.788	74.6	72.4	64.8	75.9	74.4	69.1	48.4	75.4
LAPLACIAN KERNEL	0.925	0.843	1.247	1.020	0.771	0.733	0.929	0.800	75.0	72.6	64.7	74.2	75.4	70.1	53.5	73.5

as superior performance is demonstrated when applied for real-world imbalanced regression tasks with healthcare data as inputs (i.e., PSG signals). We verify that by combining LDS and FDS, the highest performance gains are established over all tested regions.

B.4 Further Analysis and Ablation Studies

B.4.1 Kernel Type for LDS & FDS

We study the effects of different kernel types for LDS and FDS when applying distribution smoothing, in addition to the default setting where Gaussian kernels are employed. We select three different kernel types, i.e., Gaussian, Laplacian, and Triangular kernel, and evaluate their effects on both LDS and FDS. We remain other hyper-parameters unchanged as in Sec. B.2.1, and report results on IMDB-WIKI-DIR in Table B-7 and results on STS-B-DIR in Table B-8. In general, as both tables indicate, all kernel types can lead to notable

Table B-9: **Ablation study of different loss functions used during training for LDS & FDS on STS-B-DIR.**

Metrics	MSE ↓				MAE ↓				Pearson correlation (%) ↑				Spearman correlation (%) ↑			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
LDS:																
L1	0.893	0.808	1.241	0.964	0.765	0.727	0.938	0.758	76.3	73.9	66.0	75.9	76.7	71.1	54.5	75.6
MSE	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	55.6	74.3
HUBER LOSS	0.902	0.811	1.276	0.978	0.761	0.718	0.954	0.751	76.1	74.2	64.7	75.5	76.5	71.6	52.9	74.3
FDS:																
L1	0.918	0.860	1.105	1.082	0.762	0.733	0.859	0.833	75.5	73.7	65.3	72.3	75.6	70.9	52.1	71.5
MSE	0.916	0.875	1.027	1.086	0.767	0.746	0.840	0.811	75.5	73.0	67.0	72.8	75.8	69.9	54.4	72.0
HUBER LOSS	0.920	0.867	1.097	1.052	0.765	0.741	0.858	0.800	75.3	72.9	66.6	73.6	75.3	69.7	52.3	73.6

Table B-10: **Hyper-parameter study on kernel size l and standard deviation σ for LDS & FDS on IMDB-WIKI-DIR.**

Metrics		MSE ↓				MAE ↓				GM ↓			
Shot		All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA		138.06	108.70	366.09	964.92	8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
l	σ												
LDS:													
5	1	132.08	108.53	309.03	843.53	7.80	7.22	12.61	22.33	4.42	4.19	7.16	12.54
9	1	135.04	112.32	307.90	803.15	7.97	7.39	12.74	22.19	4.55	4.30	7.53	14.11
15	1	134.06	110.49	308.83	864.30	7.84	7.28	12.35	22.81	4.44	4.22	6.95	14.22
5	2	131.65	109.04	298.98	834.08	7.83	7.31	12.43	22.51	4.42	4.19	7.00	13.94
9	2	136.78	112.41	322.65	850.47	8.02	7.41	13.00	23.23	4.55	4.29	7.55	15.65
15	2	135.66	111.68	319.20	833.02	7.98	7.40	12.74	22.27	4.60	4.37	7.30	12.92
5	3	137.56	113.50	322.47	831.38	8.07	7.47	13.06	22.85	4.63	4.36	7.87	15.11
9	3	138.91	114.89	319.40	863.16	8.18	7.57	13.19	23.33	4.71	4.44	8.09	15.17
15	3	138.86	114.25	326.97	856.27	8.18	7.54	13.53	23.17	4.77	4.47	8.52	15.25
FDS:													
5	1	133.63	104.80	354.24	972.54	7.87	7.06	14.71	25.96	4.42	4.04	9.95	18.47
9	1	134.34	105.97	356.54	919.16	7.95	7.18	14.58	24.80	4.54	4.20	9.56	15.13
15	1	136.32	107.47	355.84	948.71	7.97	7.23	14.81	25.59	4.60	4.23	9.99	17.60
5	2	133.81	107.51	332.90	916.18	7.85	7.18	13.35	24.12	4.47	4.18	8.18	15.18
9	2	133.99	105.01	357.31	963.79	7.94	7.11	14.95	25.97	4.48	4.09	10.49	18.19
15	2	136.61	107.93	361.08	973.56	7.98	7.23	14.68	25.21	4.61	4.24	10.14	17.91
5	3	136.81	107.76	359.08	953.16	7.98	7.18	14.85	24.94	4.53	4.15	10.27	17.33
9	3	133.48	104.14	359.80	972.29	7.94	7.09	15.04	25.87	4.48	4.09	10.40	16.85
15	3	132.55	103.08	360.39	970.43	8.03	7.22	14.86	25.40	4.67	4.33	10.04	13.86

gains compared to the vanilla model. Moreover, Gaussian kernel often delivers the best results among all kernel types, which is consistent for both LDS and FDS.

■ B.4.2 Training Loss for LDS & FDS

In the main thesis, we fix the training loss function used for each dataset (e.g., MSE loss is used for experiments on STS-B-DIR). In this section, we investigate the influence of different training loss functions on LDS & FDS. We select three common losses used for regression tasks, i.e., L_1 loss, MSE loss, and the Huber loss (also referred to as smoothed L_1 loss).

Table B-11: **Hyper-parameter study on kernel size l and standard deviation σ for LDS & FDS on STS-B-DIR.**

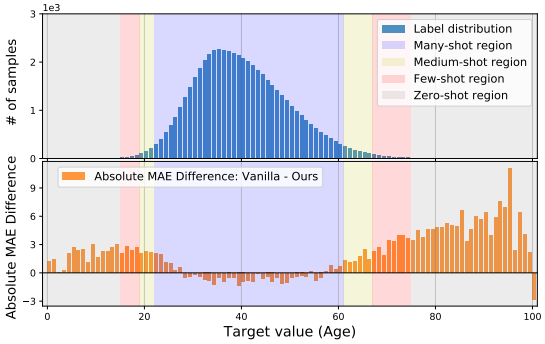
Metrics		MSE ↓				MAE ↓				Pearson correlation (%) ↑				Spearman correlation (%) ↑			
Shot		All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
VANILLA		0.974	0.851	1.520	0.984	0.794	0.740	1.043	0.771	74.2	72.0	62.7	75.2	74.4	68.8	50.5	75.0
l	σ																
LDS:																	
5	1	0.942	0.825	1.431	1.023	0.781	0.726	1.016	0.809	75.1	73.2	61.8	74.5	75.3	70.2	52.2	72.5
9	1	0.931	0.840	1.323	0.962	0.785	0.744	0.972	0.773	75.0	72.7	63.3	75.8	75.6	70.1	53.6	74.8
15	1	0.941	0.833	1.413	0.953	0.781	0.728	1.014	0.776	75.0	72.8	62.6	76.3	75.5	70.2	52.0	74.6
5	2	0.914	0.819	1.319	0.955	0.773	0.729	0.970	0.772	75.6	73.4	63.8	76.2	76.1	70.4	55.6	74.3
9	2	0.926	0.823	1.379	0.944	0.782	0.733	1.003	0.764	75.5	73.4	63.6	76.8	76.0	70.5	53.5	76.2
15	2	0.949	0.831	1.452	1.005	0.788	0.735	1.023	0.782	74.9	72.9	63.0	74.7	75.4	70.1	52.5	73.6
5	3	0.928	0.845	1.250	1.041	0.775	0.733	0.951	0.798	75.1	73.3	63.2	73.8	75.3	70.4	51.4	72.6
9	3	0.939	0.816	1.462	1.000	0.786	0.732	1.030	0.783	75.3	73.5	62.6	74.7	75.9	70.9	53.0	73.7
15	3	0.927	0.824	1.348	1.010	0.774	0.726	0.982	0.780	75.2	73.4	62.2	74.6	75.7	70.7	53.0	72.3
FDS:																	
5	1	0.943	0.869	1.217	1.066	0.776	0.742	0.914	0.799	74.4	71.7	65.6	72.5	74.2	68.4	51.1	71.2
9	1	0.927	0.851	1.193	1.096	0.770	0.736	0.896	0.822	74.9	72.8	65.8	71.6	74.8	69.7	52.3	68.3
15	1	0.926	0.854	1.202	1.029	0.776	0.743	0.914	0.800	74.9	72.6	66.1	74.0	75.1	69.8	49.5	73.6
5	2	0.916	0.875	1.027	1.086	0.767	0.746	0.840	0.811	75.5	73.0	67.0	72.8	75.8	69.9	54.4	72.0
9	2	0.933	0.888	1.068	1.081	0.776	0.752	0.855	0.839	74.8	72.0	67.9	72.2	74.9	68.9	53.3	72.0
15	2	0.944	0.890	1.125	1.078	0.783	0.761	0.864	0.822	74.4	71.8	65.8	72.2	74.5	68.9	53.1	70.9
5	3	0.924	0.860	1.190	0.964	0.771	0.740	0.897	0.790	75.0	72.7	64.4	76.1	75.1	69.4	53.8	76.5
9	3	0.932	0.878	1.149	0.982	0.770	0.746	0.876	0.780	74.8	72.5	63.8	75.3	74.8	69.3	50.2	75.6
15	3	0.956	0.915	1.110	1.016	0.784	0.767	0.855	0.803	74.4	72.1	63.7	75.5	74.3	68.7	50.0	74.6

We show the results on STS-B-DIR in Table B-9, where similar results are obtained for all the losses, with no significant performance differences observed between loss functions, indicating that FDS & LDS are robust to different loss functions.

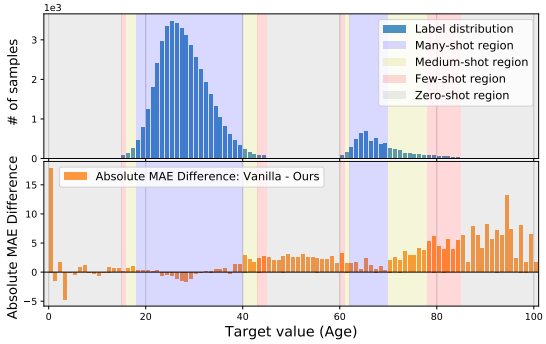
■ B.4.3 Hyper-parameters for LDS & FDS

In this section, we study the effects of different hyper-parameters on both LDS and FDS. As we mainly employ the Gaussian kernel for distribution smoothing, we extensively study different choices of the kernel size l and the standard deviation σ. Specifically, we conduct controlled experiments on IMDB-WIKI-DIR and STS-B-DIR, where we vary the choices of these hyper-parameters as $l \in \{5, 9, 15\}$ and $\sigma \in \{1, 2, 3\}$, and leave other training hyper-parameters unchanged.

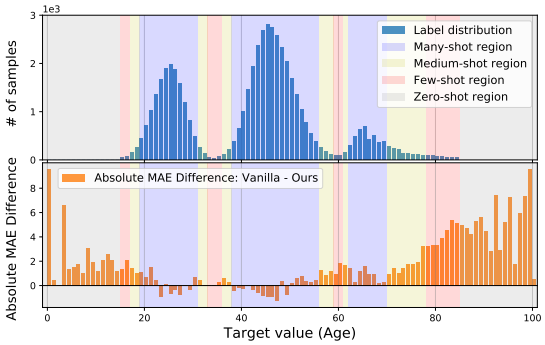
IMDB-WIKI-DIR. We first report the results on IMDB-WIKI-DIR in Table B-10. The table reveals the following observations. First, both LDS and FDS are robust to different hyper-parameters within the given range, where similar performance gains are obtained across different choices of $\{l, \sigma\}$. Specifically, for LDS, the relative MAE improvements in the few-shot regions range from 11.4% to 15.7%, where a smaller σ usually leads to slightly better results over all regions. As for FDS, similar conclusion can be made, while a smaller l often



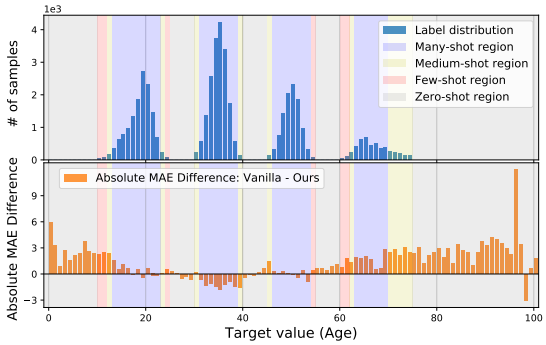
(a) Curated label distribution, with 1 Gaussian peak



(b) Curated label distribution, with 2 Gaussian peaks



(c) Curated label distribution, with 3 Gaussian peaks



(d) Curated label distribution, with 4 Gaussian peaks

Figure B-1: **The absolute MAE gains of LDS + FDS over the vanilla model under different skewed label distributions.** We curate different imbalanced label distributions on IMDB-WIKI-DIR using different number of skewed Gaussians over the target space. We confirm that LDS and FDS are robust to distribution change, and can consistently bring improvements under different imbalanced label distributions.

obtains slightly higher improvements. Interestingly, we can also observe that LDS leads to larger gains w.r.t. the performance in medium-shot and few-shot regions, while with minor degradation in many-shot regions. In contrast, FDS equally boosts all the regions, with slightly smaller improvements in medium-shot and few-shot regions compared to LDS. Finally, for both LDS and FDS, setting $l = 5$ and $\sigma = 2$ exhibits the best results.

STS-B-DIR. Further, we show the results of different hyper-parameters on STS-B-DIR in Table B-11. Similar to the results on IMDB-WIKI-DIR, we observe that both LDS and FDS are robust to the hyper-parameter changes, where the performance gaps between $\{l, \sigma\}$ pairs become smaller. In summary, the overall MSE gains range from 3.3% to 6.2% compared to the vanilla model, with $l = 5$ and $\sigma = 2$ exhibiting the best results for both LDS and FDS.

Table B-12: **Ablation study on different skewed label distributions on IMDB-WIKI-DIR.**

Metrics	MAE ↓							GM ↓						
Shot	All	Many	Med.	Few	Zero	Interp.	Extrap.	All	Many	Med.	Few	Zero	Interp.	Extrap.
1 peak:														
VANILLA	11.20	6.05	11.43	14.76	22.67	–	22.67	7.02	**3.84**	8.67	12.26	21.07	–	21.07
VANILLA + **LDS**	10.09	6.26	9.91	12.12	19.37	–	19.37	6.14	3.92	6.50	8.30	16.35	–	16.35
VANILLA + **FDS**	11.04	**5.97**	11.19	14.54	22.35	–	22.35	6.96	**3.84**	8.54	12.08	20.71	–	20.71
VANILLA + **LDS** + **FDS**	**10.00**	6.28	**9.66**	**11.83**	**19.21**	–	**19.21**	**6.09**	3.96	**6.26**	**8.14**	**15.89**	–	**15.89**
2 peaks:														
VANILLA	11.72	6.83	11.78	15.35	16.86	16.13	18.19	7.44	3.61	8.06	12.94	15.21	14.41	16.74
VANILLA + **LDS**	10.54	6.72	9.65	12.60	15.30	14.14	17.38	6.50	3.65	**5.65**	9.30	13.20	12.13	15.36
VANILLA + **FDS**	11.40	6.69	11.02	14.85	16.61	15.83	18.01	7.18	**3.50**	7.49	12.73	14.86	14.02	16.48
VANILLA + **LDS** + **FDS**	**10.27**	**6.61**	**9.46**	**11.96**	**14.89**	**13.71**	**17.02**	**6.33**	3.54	5.68	**8.80**	**12.83**	**11.71**	**15.13**
3 peaks:														
VANILLA	9.83	7.01	9.81	11.93	20.11	–	20.11	6.04	3.93	6.94	9.84	17.77	–	17.77
VANILLA + **LDS**	9.08	**6.77**	8.82	10.48	18.43	–	18.43	**5.35**	**3.78**	5.63	7.49	15.46	–	15.46
VANILLA + **FDS**	9.65	6.88	9.58	11.75	19.80	–	19.80	5.86	3.83	6.68	9.48	17.43	–	17.43
VANILLA + **LDS** + **FDS**	**8.96**	6.88	**8.62**	**10.08**	**17.76**	–	**17.76**	5.38	3.90	**5.61**	**7.36**	**14.65**	–	**14.65**
4 peaks:														
VANILLA	9.49	7.23	9.73	10.85	12.16	8.23	18.78	5.68	3.45	6.95	8.20	9.43	6.89	16.02
VANILLA + **LDS**	8.80	**6.98**	8.26	10.07	11.26	8.31	**16.22**	5.10	**3.33**	**5.07**	7.08	8.47	6.66	**12.74**
VANILLA + **FDS**	9.28	7.11	9.16	10.88	11.95	8.30	18.11	5.49	3.36	6.35	8.15	9.21	6.82	15.30
VANILLA + **LDS** + **FDS**	**8.76**	7.07	**8.23**	**9.54**	**11.13**	**8.05**	16.32	**5.05**	3.36	**5.07**	**6.56**	**8.30**	**6.34**	13.10

■ B.4.4 Robustness to Diverse Skewed Label Distributions

We analyze the effects of different skewed label distributions on our techniques for DIR tasks. We curate different imbalanced label distributions for IMDB-WIKI-DIR by combining different number of skewed Gaussians over the target space. Precisely, as shown in Fig. B-1, we create new training sets with $\{1,2,3,4\}$ disjoint skewed Gaussian distributions over the label space, with potential missing data in certain target regions, and evaluate the robustness of LDS and FDS to the distribution change.

We verify in Table B-12 that even under different imbalanced label distributions, LDS and FDS consistently bring improvements compared to the vanilla model. Substantial improvements are established not only on regions that have data, but more prominent on those without data, i.e., zero-shot regions that require target interpolation or extrapolation. We further visualize the absolute MAE gains of our methods over the vanilla model for the curated skewed distributions in Fig. B-1. Our methods provide a comprehensive treatment to the many, medium, few, as well as zero-shot regions, where remarkable performance gains are achieved across all skewed distributions, confirming the robustness of LDS and FDS under distribution change.

Table B-13: **Additional study of performance on different test set label distributions on IMDB-WIKI-DIR.**

Metrics	MSE ↓				MAE ↓				GM ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
Balanced:												
VANILLA	138.06	108.70	366.09	964.92	8.06	7.23	15.12	26.33	4.57	4.17	10.59	20.46
VANILLA + **LDS** + **FDS**	**129.35**	**106.52**	**311.49**	**811.82**	**7.78**	**7.20**	**12.61**	**22.19**	**4.37**	**4.12**	**7.39**	**12.61**
Same as training set:												
VANILLA	**68.44**	**62.10**	320.52	1350.01	**5.84**	**5.72**	15.11	30.54	**3.44**	**3.40**	11.76	24.06
VANILLA + **LDS** + **FDS**	69.86	63.43	**161.97**	**1067.89**	5.90	5.77	**9.94**	**25.17**	3.48	3.44	**7.03**	**15.95**

■ B.4.5 Additional Study on Test Set Label Distributions

We define the evaluation of DIR as generalizing to a testset that is balanced over the entire target range, which is also aligned with the evaluation in the class imbalance setting [69]. In this section, we further investigate the performance under different test set label distributions. Specifically, we consider the test set to have exactly the same label distribution as the training set, i.e., the test set also exhibits skewed label distribution (see IMDB-WIKI-DIR in Fig. 3-6). We show the results in Table B-13. As the table indicates, in the balanced testset case, using LDS and FDS can consistently improve the performance of all the regions, demonstrating that our approaches provide a comprehensive and unbiased treatment to all the target values, achieving substantial improvements. Moreover, when the testset has the same label distribution as the training set, we observe that adding LDS and FDS leads to minor degradation in the many-shot region, but drastically boosts the performance in medium-shot and few-shot regions. Note that when testset also exhibits skewed label distribution, the overall performance is dominated by the many-shot region, which can result in biased and undesired evaluation for DIR tasks.

■ B.4.6 Further Comparisons to Imbalanced Classification Methods

We provide additional study on comparisons to imbalanced classification methods. For DIR tasks that are appropriate (e.g., limited target value ranges), imbalanced classification methods can also be plugged in by discretizing the continuous label space. To gain more insights on the intrinsic difference between imbalanced classification and imbalanced regression problems, we directly apply existing imbalanced classification schemes on several appropriate DIR datasets, and show empirical comparisons with imbalanced regression

Table B-14: **Additional study on comparisons to imbalanced classification methods across several appropriate DIR datasets.**

Dataset	IMDB-WIKI-DIR (subsampled)				STS-B-DIR				NYUD2-DIR			
Metric	MAE ↓				MSE ↓				RMSE ↓			
Shot	All	Many	Med.	Few	All	Many	Med.	Few	All	Many	Med.	Few
Imbalanced Classification:												
CLS-VANILLA	15.94	15.64	18.95	30.21	1.926	1.906	2.022	1.907	1.576	0.596	1.011	2.275
CB [72]	22.41	22.32	22.05	32.90	2.159	2.194	2.028	2.107	1.664	0.592	1.044	2.415
CRT [82]	15.65	15.33	17.52	29.54	1.891	1.906	1.930	1.650	1.488	0.659	1.032	2.107
Imbalanced Regression:												
REG-VANILLA	14.64	13.98	17.47	30.29	0.974	0.851	1.520	0.984	1.477	**0.591**	0.952	2.123
LDS	14.03	13.72	15.93	26.71	0.914	0.819	1.319	0.955	1.387	0.671	0.913	1.954
FDS	13.97	13.55	16.42	24.64	0.916	0.875	**1.027**	1.086	1.442	0.615	0.940	2.059
LDS + FDS	**13.32**	**13.14**	**15.06**	**23.87**	**0.907**	**0.802**	1.363	**0.942**	**1.338**	0.670	**0.851**	**1.880**

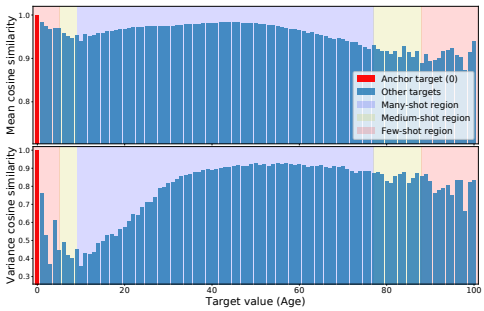
approaches. Specifically, we select the subsampled IMDB-WIKI-DIR (see Fig. 3-2), STS-B-DIR, and NYUD2-DIR for comparison. We compare with CB [72] and CRT [82], which are the state-of-the-art methods for imbalanced classification. We also denote the vanilla classification method as CLS-VANILLA. For fair comparison, the classes are set to the same bins used in LDS and FDS. Table B-14 confirms that LDS and FDS outperform imbalanced classification schemes by a large margin across all DIR datasets, where the errors for few-shot regions can be reduced by up to 50% to 60%. Interestingly, the results also show that imbalanced classification schemes often perform *worse* than even the vanilla regression model (i.e., REG-VANILLA), which confirms that regression requires different approaches for data imbalance than simply applying classification methods.

We note that imbalanced classification methods could fail on regression problems for several reasons. First, they ignore the similarity between data samples that are close w.r.t. the continuous target; Treating different target values as distinct classes is unlikely to yield the best results because it does not take advantage of the similarity between nearby targets. Moreover, classification methods cannot extrapolate or interpolate in the continuous label space, therefore unable to deal with missing data in certain target regions.

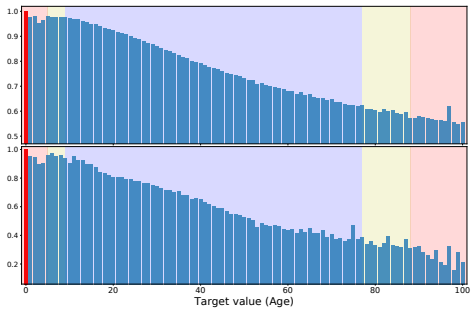
■ B.4.7 Complete Visualization for Feature Statistics Similarity

We provide additional results for understanding FDS, i.e., how FDS influences the feature statistics. In Fig. B-2, we plot the similarity of the feature statistics for different anchor ages in $\{0, 30, 60, 90\}$, using models trained without and with FDS. As the figure indicates, for

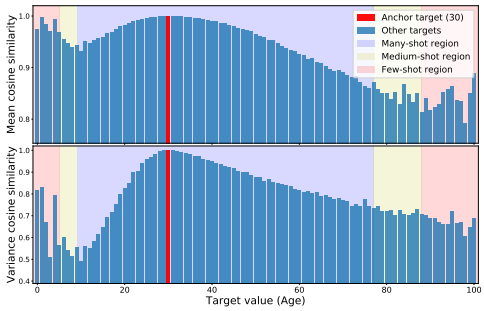
the vanilla model (i.e., Fig. B-2(a), B-2(c), B-2(e), and B-2(g)), there exists unexpected high similarities between the anchor ages and the regions that have very few data samples. For example, in Fig. B-2(a) where the anchor age is 0, the highest similarity is obtained with age range between 40 and 80, rather than its nearby ages. Moreover, for anchor ages that lie in the many-shot regions (e.g., Fig. B-2(c), B-2(e), and B-2(g)), they also exhibit unjustified feature statistics similarity with samples from age range 0 to 6, which is due to data imbalance. In contrast, by adding FDS (i.e., Fig. B-2(b), B-2(d), B-2(f), and B-2(h)), the statistics are better calibrated for all anchor ages, leading to a high similarity only in the neighborhood, and a gradually decreasing similarity score as target value becomes smaller or larger.



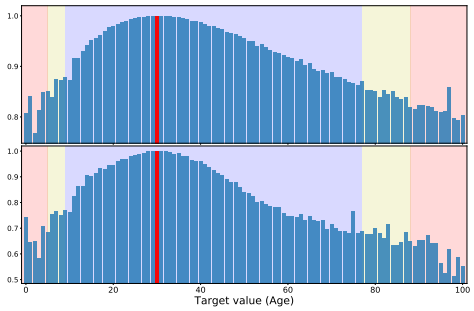
(a) Feature similarity for age 0, without FDS



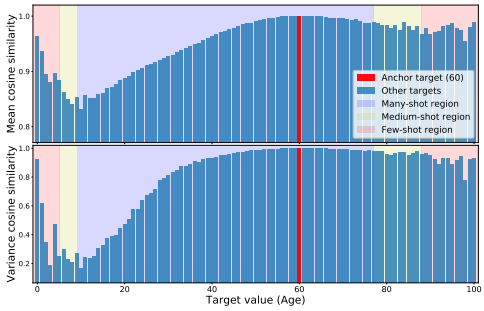
(b) Feature similarity for age 0, with FDS



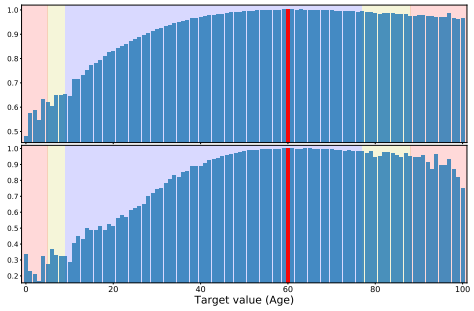
(c) Feature similarity for age 30, without FDS



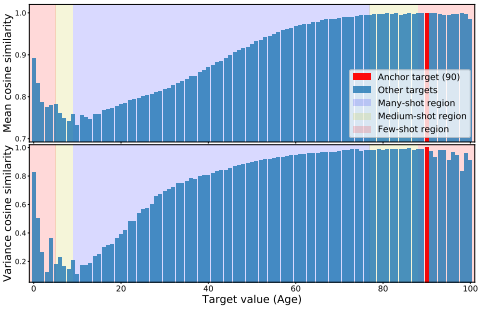
(d) Feature similarity for age 30, with FDS



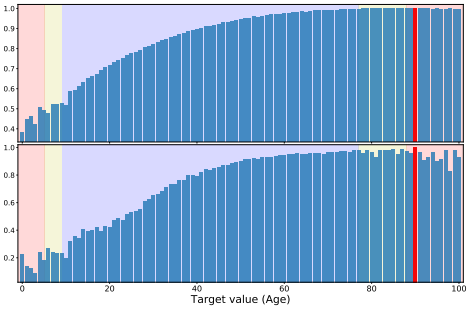
(e) Feature similarity for age 60, without FDS



(f) Feature similarity for age 60, with FDS



(g) Feature similarity for age 90, without FDS



(h) Feature similarity for age 90, with FDS

Figure B-2: **Analysis on how FDS works. First column:** Feature statistics similarity for anchor ages $\{0, 30, 60, 90\}$, using model trained without FDS. **Second column:** Feature statistics similarity for anchor ages $\{0, 30, 60, 90\}$, using model trained with FDS. We show that using FDS, the statistics are better calibrated for all anchor ages, leading to a high similarity only in the neighborhood, and a gradually decreasing similarity score as target value becomes smaller or larger.

APPENDIX C
Details and Results for Multi-Domain Long-Tailed Recognition

C.1 Theoretical Analysis and Complete Proofs

In this section, we explain the details of Theorem 4 in the main thesis, and also formally describe Theorem 6. We start with giving additional definitions and providing a useful lemma and its proof, which invoked through the proof of the theorems. We then formally prove the arguments in Theorem 4 and 6.

C.1.1 Additional Definition, Lemma, and Theorem

Definition 5 ($(\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma})$ Calibrated Transferability Statistics)**.** *The transferability graph can be further described by the following three components:*

$$\widetilde{\alpha} = \mathbb{E}_c \mathbb{E}_d \mathbb{E}_{d' \neq d} \left[\lambda_{d,c}^{d',c} \cdot \text{trans}\big((d,c),(d',c)\big) \right],$$
$$\widetilde{\beta} = \mathbb{E}_d \mathbb{E}_c \mathbb{E}_{c' \neq c} \left[\lambda_{d,c}^{d,c'} \cdot \text{trans}\big((d,c),(d,c')\big) \right],$$
$$\widetilde{\gamma} = \mathbb{E}_d \mathbb{E}_{d' \neq d} \mathbb{E}_c \mathbb{E}_{c' \neq c} \left[\lambda_{d,c}^{d',c'} \cdot \text{trans}\big((d,c),(d',c')\big) \right],$$

where $\lambda_{d,c}^{d',c'} = \left(\frac{N_{d',c'}}{N_{d,c}}\right)^{\nu}$ *denotes the distance calibration coefficient.*

Lemma 5.1. *Let $\eta, \pi > 0$ and $\varphi : \mathbb{R} \to \mathbb{R}$, $\varphi(x) = \log(\eta + \pi \exp(x))$. Given a finite sequence $x_1, x_2, \ldots, x_M \in \mathbb{R}$, it holds that*

$$\frac{1}{M}\sum_{i=1}^{M}\varphi(x_i) \geq \varphi\left(\frac{1}{M}\sum_{i=1}^{M}x_i\right).$$

Proof. Note that φ is smooth and thus twice differentiable for all $x \in \mathbb{R}$. We obtain the second derivative of φ as

$$\varphi''(x) = \frac{\eta\pi\exp(x)}{(\eta + \pi\exp(x))^2} > 0, \quad \forall x \in \mathbb{R}.$$

Therefore, φ is convex. Thus, by Jensen's inequality, we obtain that $\frac{1}{M}\sum_{i=1}^{M}\varphi(x_i) \geq \varphi\left(\frac{1}{M}\sum_{i=1}^{M}x_i\right)$, which completes the proof. □

Theorem 6 ($\widetilde{\mathcal{L}}_{\texttt{BoDA}}$ as an Upper Bound)**.** *Given a multi-domain long-tailed dataset $\mathcal{S}$ with domain label space $\mathcal{D}$ and class label space $\mathcal{C}$ satisfying $|\mathcal{D}| > 1$ and $|\mathcal{C}| > 1$, let $\mathcal{Z}$ be the representation set of all training samples. It holds that*

$$\widetilde{\mathcal{L}}_{\texttt{BoDA}}(\mathcal{Z}, \{\boldsymbol{\mu}\}) \geq N \log\left(|\mathcal{D}| - 1 + |\mathcal{D}|(|\mathcal{C}| - 1)\exp\left(\frac{|\mathcal{C}||\mathcal{D}|}{N}\cdot\widetilde{\alpha} - \frac{|\mathcal{C}|}{N}\cdot\widetilde{\beta} - \frac{|\mathcal{C}|(|\mathcal{D}|-1)}{N}\cdot\widetilde{\gamma}\right)\right), \tag{C.1}$$

where $(\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma})$ are the calibrated transferability statistics for $\mathcal{S}$ defined in Definition 5.

■ C.1.2 Proof of Theorem 4

Recall that $\mathcal{M} = \mathcal{D} \times \mathcal{C} := \{(d, c) : d \in \mathcal{D}, c \in \mathcal{C}\}$ is the set of all domain-class pairs. $\mathcal{L}_{\texttt{BoDA}}$ is given by

$$\begin{aligned}\mathcal{L}_{\texttt{BoDA}}(\mathcal{Z}, \{\boldsymbol{\mu}\}) &= \sum_{\mathbf{z}_i \in \mathcal{Z}} \frac{-1}{|\mathcal{D}| - 1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \log \frac{\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c') \in \mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)} \\ &= \sum_{\mathbf{z}_i \in \mathcal{Z}} \ell_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\}),\end{aligned}$$

where $\ell_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\})$ is the *sample-wise* BoDA loss. We rewrite $\ell_{\texttt{BoDA}}$ in the following format

$$\begin{aligned}\ell_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\}) &= -\frac{1}{|\mathcal{D}|-1}\sum_{d\in\mathcal{D}\setminus\{d_i\}}\log\frac{\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}}\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)} \\ &= \log\left(\frac{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}}\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)}{\prod_{d\in\mathcal{D}\setminus\{d_i\}}\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)^{\frac{1}{|\mathcal{D}|-1}}}\right) \\ &= \log\left(\frac{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}}\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)}{\exp\left(-\frac{1}{|\mathcal{D}|-1}\sum_{d\in\mathcal{D}\setminus\{d_i\}}\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}\right). \end{aligned} \tag{C.2}$$

We will first focus on the term in the numerator of Eqn. (C.2). We can rewrite the sum into two terms

$$\begin{aligned}&\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}}\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right) \\ &= \underbrace{\sum_{d'\in\mathcal{D}\setminus\{d_i\}}\sum_{c'\in\{c_i\}}\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)}_{T_1} + \underbrace{\sum_{d'\in\mathcal{D}}\sum_{c'\in\mathcal{C}\setminus\{c_i\}}\exp\left(-\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)}_{T_2}.\end{aligned}$$

Since the exponential function $\exp(\cdot)$ is convex, we apply Jensen's inequality on both T_1 and T_2:

$$\begin{aligned}T_1 &\geq (|\mathcal{D}|-1)\exp\left(-\frac{1}{|\mathcal{D}|-1}\sum_{d'\in\mathcal{D}\setminus\{d_i\}}\sum_{c'\in\{c_i\}}\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right) \\ &= (|\mathcal{D}|-1)\exp\left(-\frac{1}{|\mathcal{D}|-1}\sum_{d'\in\mathcal{D}\setminus\{d_i\}}\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c_i})\right), \\ T_2 &\geq |\mathcal{D}|(|\mathcal{C}|-1)\exp\left(-\frac{1}{|\mathcal{D}|(|\mathcal{C}|-1)}\sum_{d'\in\mathcal{D}}\sum_{c'\in\mathcal{C}\setminus\{c_i\}}\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right).\end{aligned}$$

Thus, by using $\exp(x)/\exp(y) = \exp(x-y)$ and rearranging terms, we bound $\ell_{\texttt{BoDA}}$ by

$$\begin{aligned}&\ell_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\}) \\ &\geq \log\left(|\mathcal{D}|-1+|\mathcal{D}|(|\mathcal{C}|-1)\exp\left(\underbrace{\frac{1}{|\mathcal{D}|-1}\sum_{d'\in\mathcal{D}\setminus\{d_i\}}\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c_i}) - \frac{1}{|\mathcal{D}|(|\mathcal{C}|-1)}\sum_{d'\in\mathcal{D}}\sum_{c'\in\mathcal{C}\setminus\{c_i\}}\widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})}_{T(\mathbf{z}_i,\{\boldsymbol{\mu}\})}\right)\right).\end{aligned}$$

Leveraging Lemma 5.1, by setting $\eta = |\mathcal{D}| - 1$, $\pi = |\mathcal{D}|(|\mathcal{C}| - 1)$, and $x_i = T(\mathbf{z}_i, \{\boldsymbol{\mu}\})$, we further bound $\mathcal{L}_{\texttt{BoDA}}(\mathcal{Z}, \{\boldsymbol{\mu}\})$ by

$$\begin{aligned}
\mathcal{L}_{\texttt{BoDA}}(\mathcal{Z}, \{\boldsymbol{\mu}\}) &= \sum_{\mathbf{z}_i \in \mathcal{Z}} \ell_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\}) \\
&\geq \sum_{\mathbf{z}_i \in \mathcal{Z}} \log\left(|\mathcal{D}| - 1 + |\mathcal{D}|(|\mathcal{C}| - 1)\exp\left(T(\mathbf{z}_i, \{\boldsymbol{\mu}\})\right)\right) \\
&\geq |\mathcal{Z}| \log\left(|\mathcal{D}| - 1 + |\mathcal{D}|(|\mathcal{C}| - 1)\exp\left(\frac{1}{|\mathcal{Z}|}\sum_{\mathbf{z}_i \in \mathcal{Z}} T(\mathbf{z}_i, \{\boldsymbol{\mu}\})\right)\right). \qquad \text{(C.3)}
\end{aligned}$$

Note that the argument of the $\exp(\cdot)$ in Eqn. (C.3) can be expanded and further rearranged as

$$\begin{aligned}
\frac{1}{|\mathcal{Z}|}\sum_{\mathbf{z}_i \in \mathcal{Z}} T(\mathbf{z}_i, \{\boldsymbol{\mu}\}) = {} & \frac{1}{|\mathcal{Z}|}\sum_{\mathbf{z}_i \in \mathcal{Z}} \frac{1}{|\mathcal{D}| - 1} \sum_{d' \in \mathcal{D} \setminus \{d_i\}} \widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d', c_i}) - \\
& \frac{1}{|\mathcal{Z}|}\sum_{\mathbf{z}_i \in \mathcal{Z}} \frac{1}{|\mathcal{D}|(|\mathcal{C}| - 1)} \sum_{d' \in \mathcal{D}} \sum_{c' \in \mathcal{C} \setminus \{c_i\}} \widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d', c'}) \\
= {} & \underbrace{\frac{1}{|\mathcal{Z}|}\frac{1}{|\mathcal{D}| - 1}\sum_{\mathbf{z}_i \in \mathcal{Z}} \sum_{d' \in \mathcal{D} \setminus \{d_i\}} \widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d', c_i})}_{T_\alpha} - \\
& \underbrace{\frac{1}{|\mathcal{Z}|}\frac{1}{|\mathcal{D}|(|\mathcal{C}| - 1)}\sum_{\mathbf{z}_i \in \mathcal{Z}} \sum_{c' \in \mathcal{C} \setminus \{c_i\}} \widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d_i, c'})}_{T_\beta} - \\
& \underbrace{\frac{1}{|\mathcal{Z}|}\frac{1}{|\mathcal{D}|(|\mathcal{C}| - 1)}\sum_{\mathbf{z}_i \in \mathcal{Z}} \sum_{d' \in \mathcal{D} \setminus \{d_i\}} \sum_{c' \in \mathcal{C} \setminus \{c_i\}} \widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d', c'})}_{T_\gamma}. \qquad \text{(C.4)}
\end{aligned}$$

Recall that each $\mathbf{z}_i \in \mathcal{Z}$ belongs to a domain-class pair (d_i, c_i), and $\mathcal{Z}_{d,c}$ denotes the representation set of $\mathcal{S}_{d,c}$ with size $N_{d,c}$. For simplicity, we remove the subscript i in the follow-

ing derivation. We can further rewrite $T_\alpha, T_\beta, T_\gamma$ as

$$\begin{aligned}
T_\alpha &= \frac{1}{|\mathcal{Z}|}\frac{1}{|\mathcal{D}|-1}\sum_{c\in\mathcal{C}}\sum_{d\in\mathcal{D}}\sum_{d'\in\mathcal{D}\setminus\{d\}}\sum_{\mathbf{z}\in\mathcal{Z}_{d,c}}\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c}) \\
&= \frac{1}{|\mathcal{Z}|}\frac{1}{|\mathcal{D}|-1}|\mathcal{C}||\mathcal{D}|(|\mathcal{D}|-1)\mathbb{E}_c\mathbb{E}_d\mathbb{E}_{d'\neq d}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\big[\underbrace{N_{d,c}\cdot\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c})}_{\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d',c})}\big] \\
&= \frac{|\mathcal{C}||\mathcal{D}|}{|\mathcal{Z}|}\underbrace{\mathbb{E}_c\mathbb{E}_d\mathbb{E}_{d'\neq d}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\big[\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d',c})\big]}_{\alpha}, \qquad \text{(C.5)} \\
T_\beta &= \frac{1}{|\mathcal{Z}|}\frac{1}{|\mathcal{D}|(|\mathcal{C}|-1)}\sum_{c\in\mathcal{C}}\sum_{d\in\mathcal{D}}\sum_{c'\in\mathcal{C}\setminus\{c\}}\sum_{\mathbf{z}\in\mathcal{Z}_{d,c}}\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d,c'}) \\
&= \frac{1}{|\mathcal{Z}|}\frac{1}{|\mathcal{D}|(|\mathcal{C}|-1)}|\mathcal{C}||\mathcal{D}|(|\mathcal{C}|-1)\mathbb{E}_d\mathbb{E}_c\mathbb{E}_{c'\neq c}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\big[\underbrace{N_{d,c}\cdot\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d,c'})}_{\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d,c'})}\big] \\
&= \frac{|\mathcal{C}|}{|\mathcal{Z}|}\underbrace{\mathbb{E}_d\mathbb{E}_c\mathbb{E}_{c'\neq c}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\big[\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d,c'})\big]}_{\beta}, \qquad \text{(C.6)} \\
T_\gamma &= \frac{1}{|\mathcal{Z}|}\frac{1}{|\mathcal{D}|(|\mathcal{C}|-1)}\sum_{c\in\mathcal{C}}\sum_{d\in\mathcal{D}}\sum_{d'\in\mathcal{D}\setminus\{d\}}\sum_{c'\in\mathcal{C}\setminus\{c\}}\sum_{\mathbf{z}\in\mathcal{Z}_{d,c}}\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c'}) \\
&= \frac{1}{|\mathcal{Z}|}\frac{|\mathcal{C}||\mathcal{D}|(|\mathcal{D}|-1)(|\mathcal{C}|-1)}{|\mathcal{D}|(|\mathcal{C}|-1)}\mathbb{E}_d\mathbb{E}_{d'\neq d}\mathbb{E}_c\mathbb{E}_{c'\neq c}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\big[\underbrace{N_{d,c}\cdot\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c'})}_{\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d',c'})}\big] \\
&= \frac{|\mathcal{C}|(|\mathcal{D}|-1)}{|\mathcal{Z}|}\underbrace{\mathbb{E}_d\mathbb{E}_{d'\neq d}\mathbb{E}_c\mathbb{E}_{c'\neq c}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\big[\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d',c'})\big]}_{\gamma}, \qquad \text{(C.7)}
\end{aligned}$$

where (α,β,γ) are the transferability statistics for $\mathcal{S}$ as in Definition 3. Finally, replace $|\mathcal{Z}| = N$ and combine Eqn. (C.3), (C.4), (C.5), (C.6), and (C.7), we have

$$\mathcal{L}_{\texttt{BoDA}}(\mathcal{Z},\{\boldsymbol{\mu}\}) \geq N\log\left(|\mathcal{D}|-1+|\mathcal{D}|(|\mathcal{C}|-1)\exp\left(\frac{|\mathcal{C}||\mathcal{D}|}{N}\cdot\alpha-\frac{|\mathcal{C}|}{N}\cdot\beta-\frac{|\mathcal{C}|(|\mathcal{D}|-1)}{N}\cdot\gamma\right)\right).$$

This completes the proof.

■ C.1.3 Proof of Theorem 6

We first define a notion of *calibrated distance* $\widehat{\mathsf{d}}$. Let $\mathbf{z}\in\mathcal{Z}_{d,c}$, we have

$$\widehat{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c'}) \triangleq \lambda_{d,c}^{d',c'}\cdot\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c'}) = \left(\frac{N_{d',c'}}{N_{d,c}}\right)^{\nu}\cdot\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c'}).$$

From Theorem 4, by substituting $\widetilde{\mathsf{d}}$ with $\widehat{\mathsf{d}}$, it holds that

$$\begin{aligned}\widetilde{\mathcal{L}}_{\texttt{BoDA}}(\mathcal{Z},\{\boldsymbol{\mu}\}) &= \mathcal{L}_{\texttt{BoDA}}(\mathcal{Z},\{\boldsymbol{\mu}\})\Big|_{\widetilde{\mathsf{d}}\to\widehat{\mathsf{d}}} \\ &\geq N\log\left(|\mathcal{D}|-1+|\mathcal{D}|(|\mathcal{C}|-1)\exp\left(T'_\alpha-T'_\beta-T'_\gamma\right)\right),\end{aligned} \tag{C.8}$$

where T'_α, T'_β, and T'_γ can be expressed as

$$\begin{aligned}T'_\alpha &= \frac{|\mathcal{C}||\mathcal{D}|}{N}\mathbb{E}_c\mathbb{E}_d\mathbb{E}_{d'\neq d}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\left[N_{d,c}\cdot\widehat{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c})\right] \\ &= \frac{|\mathcal{C}||\mathcal{D}|}{N}\mathbb{E}_c\mathbb{E}_d\mathbb{E}_{d'\neq d}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\Big[\lambda_{d,c}^{d',c}\cdot\underbrace{N_{d,c}\cdot\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c})}_{\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d',c})}\Big] \\ &= \frac{|\mathcal{C}||\mathcal{D}|}{N}\underbrace{\mathbb{E}_c\mathbb{E}_d\mathbb{E}_{d'\neq d}\left[\lambda_{d,c}^{d',c}\cdot\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\left[\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d',c})\right]\right]}_{\widetilde{\alpha}},\end{aligned} \tag{C.9}$$

$$\begin{aligned}T'_\beta &= \frac{|\mathcal{C}|}{N}\mathbb{E}_d\mathbb{E}_c\mathbb{E}_{c'\neq c}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\left[N_{d,c}\cdot\widehat{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d,c'})\right] \\ &= \frac{|\mathcal{C}|}{N}\mathbb{E}_d\mathbb{E}_c\mathbb{E}_{c'\neq c}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\Big[\lambda_{d,c}^{d,c'}\cdot\underbrace{N_{d,c}\cdot\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d,c'})}_{\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d,c'})}\Big] \\ &= \frac{|\mathcal{C}|}{N}\underbrace{\mathbb{E}_d\mathbb{E}_c\mathbb{E}_{c'\neq c}\left[\lambda_{d,c}^{d,c'}\cdot\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\left[\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d,c'})\right]\right]}_{\widetilde{\beta}},\end{aligned} \tag{C.10}$$

$$\begin{aligned}T'_\gamma &= \frac{|\mathcal{C}|(|\mathcal{D}|-1)}{N}\mathbb{E}_d\mathbb{E}_{d'\neq d}\mathbb{E}_c\mathbb{E}_{c'\neq c}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\left[N_{d,c}\cdot\widehat{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c'})\right] \\ &= \frac{|\mathcal{C}|(|\mathcal{D}|-1)}{N}\mathbb{E}_d\mathbb{E}_{d'\neq d}\mathbb{E}_c\mathbb{E}_{c'\neq c}\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\Big[\lambda_{d,c}^{d',c'}\cdot\underbrace{N_{d,c}\cdot\widetilde{\mathsf{d}}(\mathbf{z},\boldsymbol{\mu}_{d',c'})}_{\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d',c'})}\Big] \\ &= \frac{|\mathcal{C}|(|\mathcal{D}|-1)}{N}\underbrace{\mathbb{E}_d\mathbb{E}_{d'\neq d}\mathbb{E}_c\mathbb{E}_{c'\neq c}\left[\lambda_{d,c}^{d',c'}\cdot\mathbb{E}_{\mathbf{z}\in\mathcal{Z}_{d,c}}\left[\mathsf{d}(\mathbf{z},\boldsymbol{\mu}_{d',c'})\right]\right]}_{\widetilde{\gamma}},\end{aligned} \tag{C.11}$$

where $(\widetilde{\alpha},\widetilde{\beta},\widetilde{\gamma})$ are formally defined in Definition 5. Combine Eqn. (C.8), (C.9), (C.10), and (C.11), we have

$$\widetilde{\mathcal{L}}_{\texttt{BoDA}}(\mathcal{Z},\{\boldsymbol{\mu}\}) \geq N\log\left(|\mathcal{D}|-1+|\mathcal{D}|(|\mathcal{C}|-1)\exp\left(\frac{|\mathcal{C}||\mathcal{D}|}{N}\cdot\widetilde{\alpha}-\frac{|\mathcal{C}|}{N}\cdot\widetilde{\beta}-\frac{|\mathcal{C}|(|\mathcal{D}|-1)}{N}\cdot\widetilde{\gamma}\right)\right),$$

which completes the proof.

C.2 Additional Discussions, Properties, and Interpretations

C.2.1 Unified Interpretation for Single- and Multi-Domain Imbalance

In the main thesis we show that, in the multi-domain setting, label imbalance implicitly brings *label divergence* across domains, which brings additional challenges and potentially harms MDLT performance. Here we provide a unified viewpoint from the *label divergence* perspective to explain single- and multi-domain data imbalance.

To elaborate, in single domain imbalanced learning, we essentially cope with the divergence between the imbalanced training label distribution and the uniform test label distribution:

$$\operatorname{div}(p(y) \,\|\, \mathcal{U}),$$

where $\operatorname{div}(\cdot\|\cdot)$ indicates certain divergence measure. In contrast, when extending to the multi-domain scenario, given $|\mathcal{D}|$ domains with (different) imbalanced label distributions, the target divergence becomes

$$\underbrace{\sum_{d} \operatorname{div}\big(p_d(y) \,\|\, \mathcal{U}\big)}_{\text{imbalanced training}} + \text{const} \cdot \underbrace{\sum_{d \neq d'} \operatorname{div}\big(p_d(y) \,\|\, p_{d'}(y)\big)}_{\text{divergence across domains}},$$

where one not only needs to tackle the imbalanced training data for each domain $d \in \mathcal{D}$ in order to generalize to the balanced test set, but also takes into consideration the *label divergence* across domains.

Such interpretation echoes our `BoDA` objective: We design the `DA` loss for cross-domain distribution alignment to tackle the latter term, and further adapt it to `BoDA` via balanced distance to address the former term.

C.2.2 A Probabilistic Perspective of $\mathcal{L}_{\texttt{DA}}$ Derivation

Recall $\mathcal{M} = \mathcal{D} \times \mathcal{C}$ the set of all (d, c) pairs. Let $(\mathbf{x}_i, c_i, d_i)$ denote a sample with feature $\mathbf{z}_i$. Following the metric learning setting [124], we model the likelihood of $\boldsymbol{\mu}_{d,c}$ given $\mathbf{z}_i$ to decay exponentially with respect to their distance in the representation space. Such modeling can be viewed as performing a random walk with transition probability inversely related to distance [343]. For domain-class pairs that share the same class label but differ-

ent domain labels with $\mathbf{x}_i$ (i.e., $(d, c_i), d \neq d_i$), the normalized likelihood of $\boldsymbol{\mu}_{d,c_i}$ given $\mathbf{z}_i$ can be written as

$$\mathbb{P}((d, c_i)|\mathbf{z}_i) = \frac{\exp\left(-\mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)},$$

where the denominator is a sum over all domain-class pairs except (d_i, c_i). As motivated, we want to concentrate all $\mathbf{z}_i$ from the same class across different domains (i.e., smaller α), while separating $\mathbf{z}_i$ from different classes within and across domains (i.e., larger β, γ). Therefore, the positive domain-class pairs with $\mathbf{x}_i$ are those share the same class labels but different domain labels. As a result, we define the per-sample loss as the average negative log-likelihood over all positive domain-class pairs:

$$\ell_{\texttt{DA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\}) = -\frac{1}{|\mathcal{D}| - 1} \sum_{d\in\mathcal{D}\setminus\{d_i\}} \log \frac{\exp\left(-\mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)}.$$

Given a set of all training samples with representation set as $\mathcal{Z}$, the total loss can then be derived as

$$\mathcal{L}_{\texttt{DA}}(\mathcal{Z}, \{\boldsymbol{\mu}\}) = \sum_{\mathbf{z}_i\in\mathcal{Z}} \frac{-1}{|\mathcal{D}| - 1} \sum_{d\in\mathcal{D}\setminus\{d_i\}} \log \frac{\exp\left(-\mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)}.$$

■ C.2.3 Intrinsic Hardness-Aware Property of BoDA

Below, we demonstrate an additional property of BoDA: the intrinsic *hardness-aware* property. Specifically, we analyze the gradients of BoDA loss with respect to positive (d, c) pairs and different negative (d, c) pairs. We observe that the gradient contributions from *hard* positives/negatives are larger than that from the *easy* ones, indicating that BoDA automatically concentrates on the *hard* (d, c) pairs, where penalties are given according to their hardness.

Recall that the sample-wise calibrated BoDA loss $\widetilde{\ell}_{\texttt{BoDA}}$ can be written as

$$
\begin{aligned}
&\widetilde{\ell}_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\}) \\
&= -\frac{1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \log \frac{\exp\left(-\lambda_{d_i,c_i}^{d,c_i} \widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\lambda_{d_i,c_i}^{d',c'} \widetilde{\mathsf{d}}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)} \\
&= -\frac{1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \log \frac{\exp\left(-\frac{\lambda_{d_i,c_i}^{d,c_i}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\frac{\lambda_{d_i,c_i}^{d',c'}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)},
\end{aligned}
\tag{C.12}
$$

where $\mathbf{z}_i \in \mathcal{Z}_{d_i,c_i}$. For convenience, we further define the probability of $\mathbf{z}_i$ being recognized as belonging to $\boldsymbol{\mu}_{d,c}$ as

$$
P_{d,c}^i \triangleq \frac{\exp\left(-\frac{\lambda_{d_i,c_i}^{d,c}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\frac{\lambda_{d_i,c_i}^{d',c'}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)}, \quad (d,c) \in \mathcal{M} \setminus \{(d_i,c_i)\}.
$$

Note that the essential goal of Eqn. (C.12) is to align (minimize) *positive* distances $\mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})$ and to separate (maximize) *negative* distances $\mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})$. Therefore, we analyze the gradients with respect to positive distance and different negative distances to explore the properties of $\widetilde{\ell}_{\texttt{BoDA}}$. Specifically, we have

$$
\begin{aligned}
&\frac{\partial \widetilde{\ell}_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\})}{\partial \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})} \\
&= \frac{-1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \frac{\partial}{\partial \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})} \left\{ -\frac{\lambda_{d_i,c_i}^{d,c_i}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i}) - \log \sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\frac{\lambda_{d_i,c_i}^{d',c'}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right) \right\} \\
&= \frac{1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \frac{\lambda_{d_i,c_i}^{d,c_i}}{N_{d_i,c_i}} \left(1 - \frac{\exp\left(-\frac{\lambda_{d_i,c_i}^{d,c_i}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})\right)}{\sum_{(d',c')\in\mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\frac{\lambda_{d_i,c_i}^{d',c'}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})\right)} \right) \\
&= \frac{1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \frac{N_{d,c_i}^{\nu}}{N_{d_i,c_i}^{(1+\nu)}} \left(1 - P_{d,c_i}^i\right) \\
&\propto \sum_{d \in \mathcal{D}\setminus\{d_i\}} N_{d,c_i}^{\nu} \left(1 - P_{d,c_i}^i\right),
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\partial \widetilde{\ell}_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\})}{\partial \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})} \\
&= \frac{-1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \frac{\partial}{\partial \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})} \left\{ -\frac{\lambda_{d_i,c_i}^{d,c_i}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i}) - \log \sum_{(d',c') \in \mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\frac{\lambda_{d_i,c_i}^{d',c'}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'}) \right) \right\} \\
&= -\frac{1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \frac{\lambda_{d_i,c_i}^{d',c'}}{N_{d_i,c_i}} \frac{\exp\left(-\frac{\lambda_{d_i,c_i}^{d',c'}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i}) \right)}{\sum_{(d',c') \in \mathcal{M}\setminus\{(d_i,c_i)\}} \exp\left(-\frac{\lambda_{d_i,c_i}^{d',c'}}{N_{d_i,c_i}} \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'}) \right)} \\
&= -\frac{1}{|\mathcal{D}|-1} \sum_{d \in \mathcal{D}\setminus\{d_i\}} \frac{N_{d',c'}^{\nu}}{N_{d_i,c_i}^{(1+\nu)}} P_{d',c'}^{i} \\
&\propto -N_{d',c'}^{\nu} P_{d',c'}^{i}.
\end{aligned}
$$

Combine the above results, we have

$$
\text{positive:} \quad \frac{\partial \widetilde{\ell}_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\})}{\partial \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d,c_i})} \propto \sum_{d \in \mathcal{D}\setminus\{d_i\}} N_{d,c_i}^{\nu} \left(1 - P_{d,c_i}^{i}\right), \tag{C.13}
$$

$$
\text{negative:} \quad \frac{\partial \widetilde{\ell}_{\texttt{BoDA}}(\mathbf{z}_i, \{\boldsymbol{\mu}\})}{\partial \mathsf{d}(\mathbf{z}_i, \boldsymbol{\mu}_{d',c'})} \propto -N_{d',c'}^{\nu} P_{d',c'}^{i}. \tag{C.14}
$$

Interpretation. Eqn. (C.13) and (C.14) illustrate several interesting and important properties of BoDA:

- *Intrinsic hard positive and negative mining*. For positive pairs, we observe that the gradient magnitudes are proportional to $(1 - P_{d,c_i}^{i})$, where for an easy (d, c_i) pair, $P_{d,c_i}^{i} \approx 1$ and $(1 - P_{d,c_i}^{i}) \approx 0$, and for a hard (d, c_i) pair, $P_{d,c_i}^{i} \approx 0$ and $(1 - P_{d,c_i}^{i}) \approx 1$, indicating that the gradient contributions from *hard* positives are larger than *easy* ones. Similarly, for negative pairs, the gradient magnitudes are proportional to $P_{d',c'}^{i}$, where an easy (d', c') pair has $P_{d',c'}^{i} \approx 0$ and a hard (d, c_i) pair induces $P_{d',c'}^{i} \approx 1$, showing that the gradient contribution is large for hard negatives and small for easy negatives. Therefore, BoDA is a hardness-aware loss with intrinsic hard positive/negative mining property.

- *Scaling gradients according to the number of samples of each* (d, c). Furthermore, as we have shown in Fig. 4-5, when data are imbalanced across different (d, c) pairs, minority pairs with smaller number of samples would induce worse $\boldsymbol{\mu}_{d,c}$ estimates. We further observe that the gradients for both positive and negative pairs are proportional to their number of samples (i.e., N_{d,c_i}^{ν} and $N_{d',c'}^{\nu}$). This suggests that BoDA automatically adjusts the gradient scale for each (d, c) according to how accurate the estimation of $\boldsymbol{\mu}_{d,c}$ is.

Table C-1: **Detailed statistics of the curated MDLT datasets used in our experiments.** For the synthetic `Digits-MLT` dataset, we manually vary the minimum (d, c) size to simulate different degrees of imbalance.

Dataset	# Domains	# Classes	Max (d, c) size	Min (d, c) size	# Training set	# Val. set	# Test set
`Digits-MLT`	2	10	1,000	$10 \sim 1{,}000$	$20{,}000 \sim 4{,}956$	16,000	16,000
`VLCS-MLT`	4	5	1,454	0	9,872	285	572
`PACS-MLT`	4	7	741	5	7,891	700	1,400
`OfficeHome-MLT`	4	65	84	0	11,688	1,300	2,600
`TerraInc-MLT`	4	10	4,455	0	23,269	353	708
`DomainNet-MLT`	6	345	778	0	468,574	39,240	78,761

The appealing property highlights that `BoDA` also implicitly calibrates the gradient scale, emphasizing gradients from majority pairs (which are more reliable) while suppressing gradients from minority pairs (which are less reliable). Such behavior is essential for better statistics transfer as we demonstrated in the main thesis.

■ C.3 Details of MDLT Datasets

In this section, we provide the detailed information of the curated MDLT datasets we used in our experiments. Table C-1 provides an overview of the datasets. Table C-2 provides the image examples across domains for each MDLT dataset.

`Digits-MLT`. We construct `Digits-MLT` by combining two digit datasets: (1) MNIST-M [113], a variant of the original MNIST handwritten digit classification dataset [344] with colorful background, and (2) SVHN [123]. The original MNIST-M dataset contains 60,000 training samples and 10,000 testing examples, and the original SVHN dataset contains 73,257 images for training and 26,032 images for testing. Both datasets have examples of dimension $(3, 32, 32)$ and 10 classes. We create `Digits-MLT` with controllable degrees of data imbalance, where we keep the maximum number of samples each (d, c) to be 1,000, and manually vary the imbalance degree to adjust the number of samples for minority (d, c). For validation and test set, we use the original test set of the two datasets, but keep the number of samples each (d, c) to be 800.

`VLCS-MLT`. The original `VLCS` dataset [128] is an object recognition dataset that comprises photographic domains $d \in \{$ Caltech101, LabelMe, SUN09, VOC2007 $\}$, with scenes captured from urban to rural. The dataset contains 5 classes with 10,729 examples of dimen-

Table C-2: **Overview of images from different domains in all MDLT datasets.** For each dataset, we pick a single class and show illustrative images from each domain.

Dataset	Domains					
Digits-MLT	MNIST-M 	SVHN 				
VLCS-MLT	Caltech101 	LabelMe 	SUN09 	VOC2007 		
PACS-MLT	Art 	Cartoon 	Photo 	Sketch 		
OfficeHome-MLT	Art 	Clipart 	Product 	Photo 		
TerraInc-MLT	L100 	L38 	L43 	L46 		
	(camera trap location)					
DomainNet-MLT	Clipart 	Infographic 	Painting 	QuickDraw 	Photo 	Sketch 

sion $(3, 224, 224)$. To construct VLCS-MLT, for each (d, c) we split out a validation set of size 15 and a test set of size 30, and leave the rest for training.

PACS-MLT. The original PACS dataset [129] is an object recognition dataset that comprises four domains $d \in \{$ art, cartoons, photos, sketches $\}$ with image style changes. It contains 7 classes with 9,991 examples of dimension $(3, 224, 224)$. We construct PACS-MLT in a similar manner as VLCS-MLT, where we split out a validation set of size 25 and a test set of size 50 for each (d, c), and leave the rest for training.

OfficeHome-MLT. The original OfficeHome dataset [130] includes domains $d \in \{$ art, clipart, product, real $\}$, containing 15,588 examples of dimension $(3, 224, 224)$ and 65 classes. We make OfficeHome-MLT by splitting out a validation set of size 5 and a test set of size 10 for each (d, c), leaving the rest for training.

TerraInc-MLT. TerraInc-MLT is constructed from TerraIncognita dataset [131], a species classification dataset that contains photographs of wild animals taken by camera traps at

locations $d \in \{\text{L100}, \text{L38}, \text{L43}, \text{L46}\}$. The dataset contains 10 classes with 24,788 examples of dimension $(3, 224, 224)$. For each (d, c), we split out a validation set of size 10 and a test set of size 20, and use all remaining samples for training.

`DomainNet-MLT`**.** We construct `DomainNet-MLT` using `DomainNet` dataset [132], a large-scale multi-domain dataset for object recognition that consists of six domains $d \in \{$ clipart, infograph, painting, quickdraw, real, sketch $\}$, 345 classes, and 586,575 examples of size $(3, 224, 224)$. To construct `DomainNet-MLT`, for each (d, c) we split out a validation set of size 20 and a test set of size 40, and leave the rest for training.

C.4 Experimental Settings

C.4.1 Implementation Details

For the synthetic `Digits-MLT` dataset, we fix the network architecture as a small MNIST CNN [133] for all algorithms, and use no data augmentation. For all other MDLT datasets, following [133], we use the pretrained ResNet-50 model [64] as the backbone network for all algorithms, and use the same data augmentation protocol as [133]: random crop and resize to 224×224 pixels, random horizontal flips, random color jitter, grayscaling the image with 10% probability, and normalization using the ImageNet channel statistics. We train all models using the Adam optimizer [345] for 5,000 steps on all MDLT datasets except `DomainNet-MLT`, on which we train longer for 15,000 steps to ensure convergence. We fix a batch size of 64 per domain for `Digits-MLT` experiments, a batch size of 32 per domain for `DomainNet-MLT` experiments, and a batch size of 24 per domain for experiments on all other datasets.

For all MDLT datasets except `OfficeHome-MLT` and `TerraInc-MLT`, we define *many-shot* (d, c) pairs as with over 100 training samples, *medium-shot* as with 20$\sim$100 training samples, and *few-shot* as with under 20 training samples. For `OfficeHome-MLT`, we define *many-shot* as (d, c) pairs with over 60 training samples, *medium-shot* as with 20$\sim$60 training samples, and *few-shot* as with under 20 training samples. For `TerraInc-MLT`, we define *many-shot* as (d, c) pairs with over 100 training samples, *medium-shot* as with 25$\sim$100 training samples, and *few-shot* as with under 25 training samples.

■ C.4.2 Competing Algorithms

We compare BoDA to a large number of algorithms that span different learning strategies. We group them according to their categories, and provide detailed descriptions for each algorithm below.

- *Vanilla:* The empirical risk minimization (**ERM**) [127] minimizes the sum of errors across all domains and samples.
- *Group robust optimization:* Group distributionally robust optimization (**GroupDRO**) [134] performs ERM while increasing the importance of domains with larger errors.
- *Cross-domain data augmentation:* Inter-domain mixup (**Mixup**) [135] performs ERM on linear interpolations of examples from random pairs of domains and their labels. Style-agnostic network (**SagNet**) [136] disentangles style encodings from image content by randomizing and augmenting styles.
- *Meta-learning:* Meta-learning for domain generalization (**MLDG**) [116] leverages meta-learning to learn how to generalize across domains.
- *Domain-invariant representation learning:* Invariant risk minimization (**IRM**) [66] learns a feature representation such that the optimal linear classifier on top of that representation matches across domains. Domain adversarial neural networks (**DANN**) [113] employ an adversarial network to match feature distributions. Class-conditional DANN (**CDANN**) [112] builds upon DANN but further matches the conditional distributions across domains for all labels. Deep correlation alignment (**CORAL**) [111] matches the mean and covariance of feature distributions. Maximum mean discrepancy (**MMD**) [137] matches the MMD [346] of feature distributions.
- *Transfer learning:* Marginal transfer learning (**MTL**) [138] estimates a mean embedding per domain, passed as a second argument to the classifier.
- *Multi-task learning:* Gradient matching for domain generalization (**Fish**) [139] maximizes the inner product between gradients from different domains through a multi-task objective.
- *Imbalanced learning:* Focal loss (**Focal**) [99] reduces the relative loss for well-classified samples and focuses on difficult samples. Class-balanced loss (**CBLoss**) [72] proposes re-weighting by the inverse effective number of samples. The LDAM loss (**LDAM**) [71] employs a modified marginal loss that favors minority samples more. Balanced-Softmax

(**BSoftmax**) [100] extends Softmax to an unbiased estimation that considers the number of samples of each class. Self-supervised pre-training (**SSP**) [22] uses self-supervised learning as a first-stage pre-training to alleviate the network dependence on imbalanced labels. Classifier re-training (**CRT**) [82] decomposes the representation and classifier learning into two stages, where it fine-tunes the classifier using class-balanced sampling with representation fixed in the second stage.

C.4.3 Hyperparameters Search Protocol

For a fair evaluation across different algorithms, following the training protocol in [133], for each algorithm we conduct a random search of 20 trials over a joint distribution of its all hyperparameters. We then use the validation set to select the best hyperparameters for each algorithm, fix them and rerun the experiments under 3 different random seeds to report the final average accuracy (and standard deviation). Such process ensures the comparison is best-versus-best, and the hyperparameters are optimized for all algorithms.

We detail the hyperparameter choices for each algorithm in Table C-3.

C.4.4 Settings for DG Experiments

For DG experiments, we strictly follow the training protocols described in [133]. Across all benchmark DG datasets, we keep the same hyperparameter search space for `BoDA` as in Table C-3. We fix all other training parameters unchanged so that the results of `BoDA` are directly comparable to the results in [133].

For model selection, we use the *training-domain validation set* protocol in [133] with $80\% - 20\%$ training-validation split, and the average out-domain test performance is reported across all runs for each domain.

C.5 Complete Results for MDLT

We provide complete evaluation results on the five MDLT datasets. In addition to the reported results in the main thesis, for each dataset we also include the accuracy on each domain together with the averaged and the worst accuracy.

Table C-3: **Hyperparameters search space for all experiments.**

Condition	Parameter	Default value	Random distribution
General:			
ResNet	learning rate	0.00005	$10^{\text{Uniform}(-5,-3.5)}$
	dropout	0	$\text{RandomChoice}([0, 0.1, 0.5])$
	generator learning rate	0.00005	$10^{\text{Uniform}(-5,-3.5)}$
	discriminator learning rate	0.00005	$10^{\text{Uniform}(-5,-3.5)}$
not ResNet	learning rate	0.001	$10^{\text{Uniform}(-4.5,-3.5)}$
	generator learning rate	0.001	$10^{\text{Uniform}(-4.5,-2.5)}$
	discriminator learning rate	0.001	$10^{\text{Uniform}(-4.5,-2.5)}$
`Digits-MLT`	weight decay	0	0
	generator weight decay	0	0
not `Digits-MLT`	weight decay	0	$10^{\text{Uniform}(-6,-2)}$
	generator weight decay	0	$10^{\text{Uniform}(-6,-2)}$
Algorithm-specific:			
IRM	lambda	100	$10^{\text{Uniform}(-1,5)}$
	iterations of penalty annealing	500	$10^{\text{Uniform}(0,4)}$
GroupDRO	eta	0.01	$10^{\text{Uniform}(-3,-1)}$
Mixup	alpha	0.2	$10^{\text{Uniform}(0,4)}$
MLDG	beta	1	$10^{\text{Uniform}(-1,1)}$
CORAL, MMD	gamma	1	$10^{\text{Uniform}(-1,1)}$
DANN, CDANN	lambda	1.0	$10^{\text{Uniform}(-2,2)}$
	discriminator weight decay	0	$10^{\text{Uniform}(-6,-2)}$
	discriminator steps	1	$2^{\text{Uniform}(0,3)}$
	gradient penalty	0	$10^{\text{Uniform}(-2,1)}$
	adam β_1	0.5	$\text{RandomChoice}([0, 0.5])$
MTL	ema	0.99	$\text{RandomChoice}([.5, .9, .99, 1])$
SagNet	adversary weight	0.1	$10^{\text{Uniform}(-2,1)}$
Fish	meta learning rate	0.5	$\text{RandomChoice}([.05, .1, .5])$
Focal	gamma	1	$0.5 * 10^{\text{Uniform}(0,1)}$
CBLoss	beta	0.9999	$1 - 10^{\text{Uniform}(-5,-2)}$
LDAM	max_m	0.5	$10^{\text{Uniform}(-1,-0.1)}$
	scale	30	$\text{RandomChoice}([10, 30])$
BoDA	nu	1	$10^{\text{Uniform}(-0.5,0)}$
	`BoDA` loss weight	0.1	$10^{\text{Uniform}(-2,-0.5)}$

■ C.5.1 VLCS-MLT

Table C-4: **Complete evaluation results on VLCS-MLT.**

	Accuracy (by domain)						Accuracy (by shot)			
Algorithm	**C**	**L**	**S**	**V**	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM	99.3 ±0.3	53.6 ±1.1	65.9 ±1.2	86.4 ±0.7	76.3 ±0.4	53.6 ±1.1	84.6 ±0.5	76.6 ±0.4	–	32.9 ±0.4
IRM	99.1 ±0.4	52.3 ±0.7	68.8 ±1.4	86.0 ±0.3	76.5 ±0.2	52.3 ±0.7	85.3 ±0.6	75.5 ±1.0	–	33.5 ±1.0
GroupDRO	98.7 ±0.3	54.1 ±1.3	67.5 ±1.5	86.7 ±0.3	76.7 ±0.4	54.1 ±1.3	85.3 ±0.9	76.2 ±1.0	–	34.5 ±2.0
Mixup	99.3 ±0.3	52.7 ±1.3	66.1 ±0.0	85.3 ±1.1	75.9 ±0.1	52.7 ±1.3	84.4 ±0.2	77.1 ±0.6	–	29.2 ±1.4
MLDG	99.3 ±0.3	53.6 ±0.5	68.3 ±0.4	86.4 ±0.5	76.9 ±0.2	53.6 ±0.5	84.9 ±0.3	77.5 ±1.0	–	34.4 ±0.9
CORAL	99.3 ±0.3	51.6 ±0.7	67.5 ±1.8	85.3 ±0.9	75.9 ±0.5	51.6 ±0.7	84.3 ±0.6	75.5 ±0.5	–	34.5 ±0.8
MMD	99.6 ±0.2	53.4 ±0.3	65.6 ±0.8	86.7 ±1.1	76.3 ±0.6	53.4 ±0.3	84.5 ±0.8	77.1 ±0.5	–	32.7 ±0.3
DANN	99.6 ±0.2	54.1 ±0.3	69.9 ±0.2	86.7 ±0.0	77.5 ±0.1	54.1 ±0.3	85.9 ±0.5	76.0 ±0.4	–	38.0 ±2.3
CDANN	99.6 ±0.4	53.6 ±0.4	67.5 ±0.6	85.8 ±0.8	76.6 ±0.4	53.6 ±0.4	84.4 ±0.7	77.3 ±0.8	–	35.0 ±0.8
MTL	99.1 ±0.2	52.9 ±0.5	66.7 ±0.4	86.7 ±0.6	76.3 ±0.3	52.9 ±0.5	84.8 ±0.9	76.2 ±0.6	–	33.3 ±1.4
SagNet	99.6 ±0.4	52.3 ±0.2	67.2 ±0.2	86.2 ±1.0	76.3 ±0.2	52.3 ±0.2	85.3 ±0.3	75.1 ±0.2	–	32.9 ±0.3
Fish	98.7 ±0.3	54.3 ±0.4	69.4 ±0.8	87.6 ±0.4	77.5 ±0.3	54.3 ±0.4	86.2 ±0.5	76.0 ±0.4	–	35.6 ±2.2
Focal	99.1 ±0.4	52.3 ±0.2	66.1 ±0.8	84.9 ±0.2	75.6 ±0.4	52.3 ±0.2	84.0 ±0.2	75.5 ±0.6	–	32.7 ±0.9
CBLoss	99.1 ±0.2	52.5 ±0.5	68.5 ±1.0	87.1 ±1.0	76.8 ±0.3	52.5 ±0.5	84.8 ±0.7	77.5 ±1.4	–	33.2 ±1.6
LDAM	98.9 ±0.2	52.9 ±0.2	69.4 ±1.4	**88.0** ±1.3	77.5 ±0.1	52.9 ±0.2	**86.5** ±0.4	75.5 ±0.5	–	35.2 ±0.6
BSoftmax	99.3 ±0.3	52.9 ±0.9	68.0 ±0.2	86.7 ±0.8	76.7 ±0.5	52.9 ±0.9	84.4 ±0.9	78.2 ±0.6	–	34.3 ±0.9
SSP	99.1 ±0.2	52.3 ±1.0	68.0 ±0.2	85.1 ±0.4	76.1 ±0.3	52.3 ±1.0	83.8 ±0.3	76.0 ±1.2	–	37.1 ±0.7
CRT	99.6 ±0.3	51.4 ±0.3	66.9 ±0.8	86.9 ±0.4	76.3 ±0.2	51.4 ±0.3	84.5 ±0.1	77.3 ±0.0	–	31.7 ±1.0
BoDA$_r$	99.3 ±0.3	51.4 ±0.3	70.2 ±0.4	86.7 ±0.3	76.9 ±0.5	51.4 ±0.3	85.3 ±0.3	77.3 ±0.2	–	33.3 ±0.5
BoDA-M$_r$	**100.0** ±0.0	53.4 ±0.3	68.5 ±0.4	**88.0** ±0.8	77.5 ±0.3	53.4 ±0.3	85.8 ±0.2	77.3 ±0.2	–	35.7 ±0.7
BoDA$_{r,c}$	99.3 ±0.3	53.4 ±0.3	68.5 ±0.4	**88.0** ±0.4	77.3 ±0.2	53.4 ±0.3	85.3 ±0.3	78.0 ±0.2	–	38.6 ±0.7
BoDA-M$_{r,c}$	**100.0** ±0.0	**55.4** ±0.5	**72.6** ±0.3	84.7 ±0.5	**78.2** ±0.4	**55.4** ±0.5	85.3 ±0.3	**79.3** ±0.6	–	**43.3** ±1.1
BoDA *vs.* ERM	**+0.7**	**+1.8**	**+6.7**	**+1.6**	**+1.9**	**+1.8**	**+0.7**	**+2.7**	–	**+10.4**

C.5.2 PACS-MLT

Table C-5: **Complete evaluation results on PACS-MLT.**

	Accuracy (by domain)						Accuracy (by shot)			
Algorithm	**A**	**C**	**P**	**S**	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM	96.8 ±0.1	97.0 ±0.3	98.9 ±0.3	95.8 ±0.2	97.1 ±0.1	95.8 ±0.2	97.1 ±0.0	97.0 ±0.0	98.0 ±0.9	–
IRM	96.8 ±0.1	96.3 ±0.7	98.7 ±0.2	95.2 ±0.4	96.7 ±0.2	95.2 ±0.4	96.8 ±0.2	96.7 ±0.7	94.7 ±1.4	–
GroupDRO	96.9 ±0.2	97.0 ±0.4	99.0 ±0.1	95.3 ±0.4	97.0 ±0.1	95.3 ±0.4	97.3 ±0.1	95.3 ±1.2	94.7 ±3.6	–
Mixup	96.5 ±0.3	96.9 ±0.7	98.5 ±0.2	95.1 ±0.2	96.7 ±0.2	95.1 ±0.2	97.0 ±0.1	96.7 ±0.3	91.3 ±2.7	–
MLDG	96.6 ±0.2	97.2 ±0.3	98.5 ±0.1	94.1 ±0.3	96.6 ±0.1	94.1 ±0.3	96.8 ±0.1	96.3 ±0.7	92.7 ±0.5	–
CORAL	96.9 ±0.4	97.0 ±0.5	98.3 ±0.3	94.3 ±0.7	96.6 ±0.5	94.3 ±0.7	96.6 ±0.5	97.0 ±0.8	94.7 ±0.5	–
MMD	96.8 ±0.2	97.1 ±0.4	97.4 ±0.3	96.3 ±0.3	96.9 ±0.1	96.2 ±0.2	96.9 ±0.2	97.0 ±0.0	96.7 ±0.5	–
DANN	95.7 ±0.3	97.2 ±0.4	98.9 ±0.1	94.3 ±0.1	96.5 ±0.0	94.3 ±0.1	96.5 ±0.1	98.0 ±0.0	94.7 ±2.4	–
CDANN	95.5 ±0.5	96.7 ±0.2	97.2 ±0.3	94.9 ±0.5	96.1 ±0.1	94.5 ±0.2	96.1 ±0.1	96.3 ±0.5	94.0 ±0.9	–
MTL	96.3 ±0.4	97.9 ±0.3	98.2 ±0.3	94.6 ±0.7	96.7 ±0.2	94.5 ±0.6	96.8 ±0.1	95.3 ±1.7	97.3 ±1.1	–
SagNet	97.0 ±0.2	97.8 ±0.4	98.9 ±0.1	95.2 ±0.3	**97.2** ±0.1	95.2 ±0.3	**97.4** ±0.1	96.7 ±0.5	95.3 ±0.5	–
Fish	95.5 ±0.2	97.9 ±0.4	98.2 ±0.3	95.9 ±0.5	96.9 ±0.2	95.2 ±0.2	97.0 ±0.1	97.0 ±0.5	94.7 ±1.1	–
Focal	96.6 ±0.4	96.6 ±0.8	98.1 ±0.2	94.6 ±0.7	96.5 ±0.2	94.6 ±0.7	96.6 ±0.1	95.0 ±1.7	96.7 ±0.5	–
CBLoss	**97.3** ±0.1	97.4 ±0.5	97.8 ±0.6	95.1 ±0.4	96.9 ±0.1	95.1 ±0.4	96.8 ±0.2	97.0 ±1.2	**100.0** ±0.0	–
LDAM	96.9 ±0.1	96.6 ±0.6	97.9 ±0.1	94.7 ±0.2	96.5 ±0.2	94.7 ±0.2	96.6 ±0.1	95.7 ±1.4	96.0 ±0.0	–
BSoftmax	96.0 ±0.5	96.9 ±0.6	98.8 ±0.6	95.9 ±0.1	96.9 ±0.3	95.6 ±0.3	96.6 ±0.4	**98.7** ±0.7	99.3 ±0.5	–
SSP	96.2 ±0.5	96.8 ±0.2	98.9 ±0.1	95.7 ±0.3	96.9 ±0.2	95.4 ±0.4	96.7 ±0.2	98.3 ±0.5	98.0 ±0.9	–
CRT	95.3 ±0.2	96.7 ±0.1	98.5 ±0.1	94.9 ±0.1	96.3 ±0.1	94.9 ±0.1	96.3 ±0.1	97.3 ±0.3	94.0 ±0.9	–
BoDA$_r$	96.9 ±0.4	97.4 ±0.2	98.6 ±0.2	95.1 ±0.4	97.0 ±0.1	95.1 ±0.4	97.0 ±0.1	96.3 ±0.5	98.0 ±0.9	–
BoDA-M$_r$	96.6 ±0.2	**98.0** ±0.2	99.1 ±0.2	94.9 ±0.1	97.1 ±0.1	94.9 ±0.1	97.3 ±0.1	96.3 ±0.5	96.0 ±0.0	–
BoDA$_{r,c}$	96.3 ±0.1	97.4 ±0.5	**99.4** ±0.3	95.7 ±0.3	**97.2** ±0.1	95.7 ±0.3	**97.4** ±0.1	97.0 ±0.0	94.7 ±1.1	–
BoDA-M$_{r,c}$	96.3 ±0.4	97.7 ±0.2	98.1 ±0.4	**96.4** ±0.2	97.1 ±0.2	**96.3** ±0.1	97.1 ±0.0	97.0 ±0.8	96.0 ±0.0	–
BoDA *vs.* ERM	-0.5	+0.7	+0.5	+0.6	+0.1	+0.5	+0.3	+0.0	-2.0	–

C.5.3 OfficeHome-MLT

Table C-6: **Complete evaluation results on OfficeHome-MLT.**

	Accuracy (by domain)						Accuracy (by shot)			
Algorithm	**A**	**C**	**P**	**R**	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM	71.3 ±0.1	78.4 ±0.2	89.6 ±0.3	83.3 ±0.2	80.7 ±0.0	71.3 ±0.1	87.8 ±0.2	81.0 ±0.2	63.1 ±0.1	63.3 ±7.2
IRM	70.7 ±0.2	78.5 ±0.8	89.4 ±0.5	83.8 ±0.6	80.6 ±0.4	70.7 ±0.2	87.6 ±0.4	81.5 ±0.4	61.1 ±0.9	56.7 ±1.4
GroupDRO	68.7 ±0.9	79.0 ±0.2	89.4 ±0.4	83.3 ±0.5	80.1 ±0.3	68.7 ±0.9	88.1 ±0.2	80.8 ±0.4	59.8 ±1.2	51.7 ±3.6
Mixup	72.3 ±0.6	79.1 ±0.4	89.7 ±0.1	83.9 ±0.2	81.2 ±0.2	72.3 ±0.6	87.9 ±0.4	81.8 ±0.1	64.1 ±0.4	60.0 ±4.1
MLDG	70.2 ±0.6	78.2 ±0.5	89.4 ±0.4	83.7 ±0.3	80.4 ±0.2	70.2 ±0.6	87.1 ±0.1	81.3 ±0.3	61.3 ±1.0	61.7 ±1.4
CORAL	**72.7** ±0.6	80.9 ±0.3	89.9 ±0.2	84.2 ±0.4	81.9 ±0.1	**72.7** ±0.6	87.9 ±0.1	83.0 ±0.1	63.5 ±0.7	65.0 ±2.4
MMD	67.7 ±0.8	77.8 ±0.2	87.4 ±0.5	80.6 ±0.4	78.4 ±0.4	67.7 ±0.8	85.2 ±0.2	79.4 ±0.7	58.8 ±0.4	56.7 ±3.6
DANN	70.2 ±0.9	77.3 ±0.3	87.3 ±0.5	82.1 ±0.4	79.2 ±0.2	70.2 ±0.9	86.2 ±0.1	80.0 ±0.1	60.3 ±1.1	61.7 ±5.9
CDANN	69.4 ±0.3	77.2 ±0.3	87.7 ±0.2	81.5 ±0.3	79.0 ±0.2	69.4 ±0.3	86.4 ±0.6	79.8 ±0.1	58.9 ±0.8	50.0 ±4.7
MTL	69.8 ±0.6	77.6 ±0.3	87.9 ±0.1	82.4 ±0.3	79.5 ±0.2	69.8 ±0.6	87.3 ±0.3	79.8 ±0.2	61.1 ±0.2	51.7 ±2.7
SagNet	70.5 ±0.5	79.6 ±0.5	89.3 ±0.4	83.9 ±0.1	80.9 ±0.1	70.5 ±0.5	87.8 ±0.4	81.9 ±0.1	61.2 ±0.9	56.7 ±3.6
Fish	71.3 ±0.7	79.1 ±0.1	**90.2** ±0.6	84.7 ±0.4	81.3 ±0.3	71.3 ±0.7	**88.2** ±0.2	81.9 ±0.3	63.2 ±0.8	61.7 ±1.4
Focal	67.6 ±0.4	76.6 ±0.8	87.1 ±0.5	80.2 ±0.3	77.9 ±0.0	67.6 ±0.4	86.5 ±0.3	78.3 ±0.1	57.4 ±0.3	46.7 ±3.6
CBLoss	69.5 ±0.7	78.7 ±0.3	88.9 ±0.4	82.2 ±0.1	79.8 ±0.2	69.5 ±0.7	86.6 ±0.4	80.6 ±0.2	61.1 ±1.4	65.0 ±2.4
LDAM	69.9 ±0.5	78.9 ±0.4	89.4 ±0.3	83.0 ±0.4	80.3 ±0.2	69.9 ±0.5	87.1 ±0.2	81.3 ±0.3	61.1 ±0.2	51.7 ±2.7
BSoftmax	70.9 ±0.5	78.7 ±0.2	89.0 ±0.8	83.0 ±0.3	80.4 ±0.2	70.9 ±0.5	86.7 ±0.5	81.3 ±0.3	62.4 ±1.0	60.0 ±4.1
SSP	71.1 ±0.3	79.6 ±0.8	89.4 ±0.3	84.2 ±0.2	81.1 ±0.3	71.1 ±0.3	87.3 ±0.6	82.3 ±0.3	61.6 ±0.7	63.3 ±1.4
CRT	72.5 ±0.2	79.6 ±0.2	88.9 ±0.1	83.6 ±0.2	81.2 ±0.0	72.5 ±0.2	87.7 ±0.1	81.8 ±0.1	64.0 ±0.1	65.0 ±2.4
BoDA$_r$	71.8 ±0.1	80.3 ±0.3	89.1 ±0.4	84.6 ±0.2	81.5 ±0.1	71.8 ±0.1	87.7 ±0.2	82.3 ±0.1	**64.2** ±0.3	63.3 ±1.4
BoDA-M$_r$	71.6 ±0.2	80.5 ±0.3	89.2 ±0.2	85.7 ±0.4	81.9 ±0.2	71.6 ±0.2	87.3 ±0.3	83.4 ±0.2	62.3 ±0.3	65.0 ±2.4
BoDA$_{r,c}$	72.3 ±0.3	80.8 ±0.2	89.4 ±0.4	**86.3** ±0.3	82.3 ±0.1	72.3 ±0.3	87.1 ±0.2	**83.9** ±0.3	63.2 ±0.2	65.0 ±2.4
BoDA-M$_{r,c}$	72.3 ±0.3	**81.5** ±0.4	89.5 ±0.3	85.8 ±0.2	**82.4** ±0.2	72.3 ±0.3	87.7 ±0.1	**83.9** ±0.6	**64.2** ±0.3	**66.7** ±2.7
BoDA *vs.* ERM	**+1.0**	**+3.1**	**–0.1**	**+3.0**	**+1.7**	**+1.0**	**–0.1**	**+2.9**	**+1.1**	**+3.4**

■ C.5.4 TerraInc-MLT

Table C-7: **Complete evaluation results on TerraInc-MLT.**

	Accuracy (by domain)						Accuracy (by shot)			
Algorithm	**L100**	**L38**	**L43**	**L46**	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM	80.3 ±1.3	71.2 ±0.7	82.2 ±0.3	67.4 ±0.3	75.3 ±0.3	67.4 ±0.3	85.6 ±0.8	69.6 ±3.2	66.1 ±2.4	14.4 ±2.8
IRM	78.2 ±0.9	69.6 ±2.0	81.1 ±0.7	64.3 ±1.3	73.3 ±0.7	64.3 ±1.3	83.5 ±0.6	70.0 ±1.8	58.3 ±3.4	20.1 ±1.4
GroupDRO	68.3 ±1.0	68.8 ±1.3	82.6 ±0.2	68.1 ±0.8	72.0 ±0.4	66.6 ±0.2	84.7 ±1.1	64.6 ±4.7	38.9 ±1.2	13.5 ±1.1
Mixup	75.4 ±1.4	70.2 ±1.3	78.3 ±0.6	60.4 ±1.1	71.1 ±0.7	60.4 ±1.1	83.2 ±0.7	60.0 ±0.6	56.1 ±3.0	12.2 ±2.1
MLDG	82.3 ±0.9	73.5 ±2.0	83.8 ±1.4	66.9 ±0.5	76.6 ±0.2	66.9 ±0.5	86.1 ±0.6	73.8 ±3.9	70.6 ±3.7	18.8 ±2.4
CORAL	81.6 ±1.0	72.0 ±0.6	84.2 ±0.2	67.8 ±0.9	76.4 ±0.5	67.8 ±0.9	86.3 ±0.3	77.5 ±3.1	66.1 ±2.0	11.0 ±1.4
MMD	78.9 ±0.6	68.8 ±1.0	81.9 ±0.9	63.7 ±1.1	73.3 ±0.4	63.7 ±1.1	84.0 ±0.4	67.9 ±2.7	60.6 ±1.6	13.6 ±2.6
DANN	74.1 ±0.8	63.1 ±1.9	75.9 ±0.2	61.5 ±0.9	68.7 ±0.9	61.1 ±1.0	79.6 ±1.2	62.5 ±8.1	48.9 ±2.8	13.3 ±1.1
CDANN	73.0 ±1.3	67.8 ±2.0	75.0 ±0.6	65.2 ±1.1	70.3 ±0.5	63.9 ±1.0	83.5 ±0.8	50.0 ±4.2	43.9 ±4.7	20.4 ±3.1
MTL	79.4 ±0.8	70.8 ±0.6	81.9 ±0.8	67.8 ±1.4	75.0 ±0.7	67.7 ±1.4	85.2 ±0.7	73.8 ±1.6	61.1 ±2.8	12.4 ±4.0
SagNet	79.4 ±1.8	71.2 ±0.7	83.4 ±2.4	66.5 ±2.1	75.1 ±1.6	66.5 ±2.1	85.5 ±0.9	77.1 ±5.0	57.8 ±4.3	13.0 ±3.4
Fish	80.1 ±1.9	70.2 ±0.2	84.4 ±0.9	66.3 ±0.5	75.3 ±0.5	66.3 ±0.5	85.8 ±0.2	73.3 ±3.9	61.1 ±3.0	13.7 ±3.3
Focal	80.9 ±0.7	71.6 ±1.6	84.4 ±1.3	66.1 ±1.7	75.7 ±0.4	65.3 ±1.1	85.7 ±0.3	76.2 ±3.9	68.9 ±3.2	12.6 ±1.9
CBLoss	84.9 ±0.6	78.0 ±1.2	80.7 ±0.3	68.3 ±2.0	78.0 ±0.4	68.3 ±2.0	85.0 ±0.1	89.2 ±1.2	83.9 ±2.5	9.3 ±3.9
LDAM	83.0 ±0.9	70.6 ±0.6	81.3 ±1.1	64.1 ±1.4	74.7 ±0.9	64.1 ±1.4	85.1 ±0.6	70.8 ±3.5	67.8 ±1.2	11.1 ±2.4
BSoftmax	83.5 ±2.1	75.5 ±0.4	82.1 ±0.7	65.6 ±1.3	76.7 ±1.0	65.6 ±1.3	83.4 ±0.8	90.8 ±0.9	78.3 ±3.9	12.6 ±2.4
SSP	82.6 ±1.3	80.7 ±1.8	83.2 ±0.6	67.3 ±0.4	78.5 ±0.7	67.3 ±0.4	85.5 ±1.0	87.8 ±0.9	82.6 ±1.2	13.2 ±2.8
CRT	89.0 ±0.1	81.8 ±0.3	85.8 ±0.3	70.0 ±0.4	81.6 ±0.1	70.0 ±0.4	**89.7** ±0.2	90.4 ±0.3	83.9 ±0.5	12.9 ±0.0
BoDA$_r$	86.7 ±0.7	74.1 ±1.1	85.2 ±0.7	68.5 ±0.3	78.6 ±0.4	68.5 ±0.3	86.4 ±0.1	85.0 ±1.0	80.0 ±0.9	13.7 ±2.1
BoDA-M$_r$	87.8 ±0.9	76.5 ±0.9	82.2 ±0.3	71.3 ±0.4	79.4 ±0.6	71.3 ±0.4	88.4 ±0.3	76.2 ±2.7	88.3 ±1.6	14.4 ±1.4
BoDA$_{r,c}$	88.3 ±0.6	**82.9** ±0.5	**89.3** ±0.9	68.5 ±0.6	82.3 ±0.3	68.5 ±0.6	89.2 ±0.2	**92.5** ±0.9	88.3 ±1.2	21.3 ±0.7
BoDA-M$_{r,c}$	**90.4** ±0.3	81.2 ±0.7	85.8 ±0.4	**74.6** ±0.7	**83.0** ±0.4	**74.6** ±0.7	89.2 ±0.2	91.2 ±0.6	**91.7** ±2.0	**21.7** ±1.4
BoDA *vs.* ERM	**+10.1**	**+11.7**	**+7.1**	**+7.2**	**+7.7**	**+7.2**	**+3.6**	**+22.9**	**+25.6**	**+7.3**

■ C.5.5 DomainNet-MLT

Table C-8: **Complete evaluation results on** `DomainNet-MLT`**.**

	Accuracy (by domain)								Accuracy (by shot)			
Algorithm	**clip**	**info**	**paint**	**quick**	**real**	**sketch**	**Average**	**Worst**	**Many**	**Medium**	**Few**	**Zero**
ERM	68.6 ±0.1	29.4 ±0.3	57.1 ±0.2	62.8 ±0.3	72.1 ±0.2	61.7 ±0.2	58.6 ±0.2	29.4 ±0.3	66.0 ±0.1	56.1 ±0.1	35.9 ±0.5	27.6 ±0.3
IRM	66.7 ±0.2	27.6 ±0.1	56.0 ±0.2	60.1 ±0.1	72.0 ±0.0	60.2 ±0.2	57.1 ±0.1	27.6 ±0.1	64.7 ±0.1	54.3 ±0.3	33.5 ±0.3	25.8 ±0.3
GroupDRO	60.1 ±0.2	25.9 ±0.2	50.3 ±0.1	63.9 ±0.2	64.9 ±0.2	56.7 ±0.3	53.6 ±0.1	25.9 ±0.2	61.8 ±0.1	49.1 ±0.3	30.7 ±0.7	22.0 ±0.1
Mixup	67.6 ±0.2	28.7 ±0.0	56.4 ±0.2	60.0 ±0.4	72.1 ±0.1	60.9 ±0.1	57.6 ±0.1	28.7 ±0.0	64.9 ±0.2	54.5 ±0.1	35.6 ±0.2	27.3 ±0.3
MLDG	68.0 ±0.2	28.7 ±0.1	57.2 ±0.1	61.6 ±0.2	73.3 ±0.1	61.9 ±0.2	58.5 ±0.0	28.7 ±0.1	66.0 ±0.1	55.7 ±0.1	35.3 ±0.2	26.9 ±0.3
CORAL	69.1 ±0.3	30.1 ±0.4	57.8 ±0.2	63.4 ±0.2	72.8 ±0.2	63.3 ±0.3	59.4 ±0.1	30.1 ±0.4	66.4 ±0.1	57.1 ±0.0	37.7 ±0.6	29.9 ±0.2
MMD	66.1 ±0.1	27.2 ±0.2	55.9 ±0.1	59.3 ±0.2	71.9 ±0.1	60.0 ±0.2	56.7 ±0.0	27.2 ±0.2	64.2 ±0.1	54.0 ±0.0	33.9 ±0.2	25.4 ±0.2
DANN	65.5 ±0.3	26.9 ±0.4	55.2 ±0.1	57.4 ±0.2	70.6 ±0.1	59.0 ±0.2	55.8 ±0.1	26.9 ±0.4	63.0 ±0.1	52.7 ±0.1	34.2 ±0.4	26.8 ±0.4
CDANN	65.9 ±0.1	27.7 ±0.1	55.3 ±0.1	57.6 ±0.2	70.9 ±0.2	58.7 ±0.1	56.0 ±0.1	27.7 ±0.1	63.2 ±0.0	52.7 ±0.2	34.3 ±0.5	27.6 ±0.1
MTL	68.2 ±0.2	29.3 ±0.2	57.3 ±0.1	62.1 ±0.1	72.9 ±0.1	61.8 ±0.2	58.6 ±0.1	29.3 ±0.2	65.9 ±0.1	56.0 ±0.4	35.4 ±0.1	28.2 ±0.3
SagNet	68.5 ±0.1	29.4 ±0.2	57.8 ±0.2	62.1 ±0.2	73.3 ±0.1	62.4 ±0.1	58.9 ±0.0	29.4 ±0.2	66.3 ±0.1	56.4 ±0.0	36.2 ±0.3	27.2 ±0.4
Fish	68.7 ±0.1	29.1 ±0.1	58.4 ±0.1	64.1 ±0.1	73.9 ±0.1	63.7 ±0.1	59.6 ±0.1	29.1 ±0.1	67.1 ±0.1	57.2 ±0.1	36.8 ±0.4	27.8 ±0.3
Focal	67.6 ±0.1	27.5 ±0.1	56.5 ±0.3	62.3 ±0.3	71.7 ±0.3	61.4 ±0.3	57.8 ±0.2	27.5 ±0.1	65.2 ±0.2	55.1 ±0.2	35.8 ±0.1	26.3 ±0.1
CBLoss	68.3 ±0.2	30.1 ±0.1	57.8 ±0.1	60.8 ±0.1	73.3 ±0.2	63.3 ±0.1	58.9 ±0.1	30.1 ±0.1	64.3 ±0.0	61.0 ±0.3	42.5 ±0.4	28.1 ±0.2
LDAM	68.8 ±0.2	29.2 ±0.2	57.1 ±0.1	65.0 ±0.0	72.3 ±0.1	63.1 ±0.1	59.2 ±0.0	29.2 ±0.2	66.6 ±0.0	57.0 ±0.0	37.1 ±0.2	27.8 ±0.3
BSoftmax	68.5 ±0.1	29.9 ±0.1	57.8 ±0.1	60.5 ±0.3	73.4 ±0.1	63.3 ±0.0	58.9 ±0.1	29.9 ±0.1	64.3 ±0.1	60.9 ±0.3	42.4 ±0.6	28.2 ±0.1
SSP	69.7 ±0.1	31.6 ±0.2	58.8 ±0.1	59.7 ±0.3	73.9 ±0.1	64.2 ±0.1	59.7 ±0.0	31.6 ±0.2	64.3 ±0.1	62.6 ±0.1	45.0 ±0.3	30.5 ±0.0
CRT	70.0 ±0.1	31.6 ±0.1	59.2 ±0.2	64.0 ±0.1	73.4 ±0.1	64.4 ±0.1	60.4 ±0.2	31.6 ±0.1	66.8 ±0.0	61.6 ±0.1	45.7 ±0.1	29.7 ±0.1
BoDA$_r$	70.0 ±0.1	32.6 ±0.1	59.1 ±0.1	61.2 ±0.4	73.3 ±0.1	64.1 ±0.1	60.1 ±0.2	32.6 ±0.1	65.7 ±0.2	60.6 ±0.1	42.6 ±0.3	30.5 ±0.2
BoDA-M$_r$	70.6 ±0.1	32.2 ±0.2	57.7 ±0.3	**65.5** ±0.3	70.2 ±0.1	64.5 ±0.1	60.1 ±0.2	32.2 ±0.2	65.9 ±0.2	60.7 ±0.1	42.9 ±0.3	30.0 ±0.1
BoDA$_{r,c}$	**72.0** ±0.2	**33.4** ±0.1	60.7 ±0.2	63.6 ±0.2	**74.6** ±0.1	65.5 ±0.2	**61.7** ±0.1	**33.4** ±0.1	**67.0** ±0.1	62.7 ±0.1	46.0 ±0.2	**32.2** ±0.3
BoDA-M$_{r,c}$	71.8 ±0.1	33.3 ±0.1	**60.8** ±0.1	63.7 ±0.3	**74.6** ±0.1	**65.8** ±0.2	**61.7** ±0.2	33.3 ±0.1	**67.0** ±0.1	**63.0** ±0.3	**46.6** ±0.4	31.8 ±0.2
BoDA *vs.* ERM	**+3.4**	**+4.0**	**+3.7**	**+0.9**	**+2.5**	**+4.1**	**+3.1**	**+4.0**	**+1.0**	**+6.9**	**+10.7**	**+4.6**

C.6 Complete Results for DG

We provide detailed results of Table 4-9 across five DG benchmarks [133]. Results for all algorithms except BoDA are directly copied from [133].

C.6.1 VLCS

Table C-9: **Complete domain generalization results on VLCS.**

Algorithm	C	L	S	V	Avg
ERM	97.7 ±0.4	64.3 ±0.9	73.4 ±0.5	74.6 ±1.3	77.5
IRM	98.6 ±0.1	64.9 ±0.9	73.4 ±0.6	77.3 ±0.9	78.5
GroupDRO	97.3 ±0.3	63.4 ±0.9	69.5 ±0.8	76.7 ±0.7	76.7
Mixup	98.3 ±0.6	64.8 ±1.0	72.1 ±0.5	74.3 ±0.8	77.4
MLDG	97.4 ±0.2	65.2 ±0.7	71.0 ±1.4	75.3 ±1.0	77.2
CORAL	98.3 ±0.1	**66.1** ±1.2	73.4 ±0.3	77.5 ±1.2	**78.8**
MMD	97.7 ±0.1	64.0 ±1.1	72.8 ±0.2	75.3 ±3.3	77.5
DANN	**99.0** ±0.3	65.1 ±1.4	73.1 ±0.3	77.2 ±0.6	78.6
CDANN	97.1 ±0.3	65.1 ±1.2	70.7 ±0.8	77.1 ±1.5	77.5
MTL	97.8 ±0.4	64.3 ±0.3	71.5 ±0.7	75.3 ±1.7	77.2
SagNet	97.9 ±0.4	64.5 ±0.5	71.4 ±1.3	77.5 ±0.5	77.8
ARM	98.7 ±0.2	63.6 ±0.7	71.3 ±1.2	76.7 ±0.6	77.6
VREx	98.4 ±0.3	64.4 ±1.4	74.1 ±0.4	76.2 ±1.3	78.3
RSC	97.9 ±0.1	62.5 ±0.7	72.3 ±1.2	75.6 ±0.8	77.1
BoDA	98.1 ±0.3	64.5 ±0.4	**74.3** ±0.3	**78.0** ±0.6	78.5

C.6.2 PACS

Table C-10: **Complete domain generalization results on PACS.**

Algorithm	A	C	P	S	Avg
ERM	84.7 ±0.4	80.8 ±0.6	97.2 ±0.3	79.3 ±1.0	85.5
IRM	84.8 ±1.3	76.4 ±1.1	96.7 ±0.6	76.1 ±1.0	83.5
GroupDRO	83.5 ±0.9	79.1 ±0.6	96.7 ±0.3	78.3 ±2.0	84.4
Mixup	86.1 ±0.5	78.9 ±0.8	97.6 ±0.1	75.8 ±1.8	84.6
MLDG	85.5 ±1.4	80.1 ±1.7	97.4 ±0.3	76.6 ±1.1	84.9
CORAL	**88.3** ±0.2	80.0 ±0.5	97.5 ±0.3	78.8 ±1.3	86.2
MMD	86.1 ±1.4	79.4 ±0.9	96.6 ±0.2	76.5 ±0.5	84.6
DANN	86.4 ±0.8	77.4 ±0.8	97.3 ±0.4	73.5 ±2.3	83.7
CDANN	84.6 ±1.8	75.5 ±0.9	96.8 ±0.3	73.5 ±0.6	82.6
MTL	87.5 ±0.8	77.1 ±0.5	96.4 ±0.8	77.3 ±1.8	84.6
SagNet	87.4 ±1.0	80.7 ±0.6	97.1 ±0.1	80.0 ±0.4	86.3
ARM	86.8 ±0.6	76.8 ±0.5	97.4 ±0.3	79.3 ±1.2	85.1
VREx	86.0 ±1.6	79.1 ±0.6	96.9 ±0.5	77.7 ±1.7	84.9
RSC	85.4 ±0.8	79.7 ±1.8	97.6 ±0.3	78.2 ±1.2	85.2
BoDA	88.2 ±0.2	**81.7** ±0.3	**97.8** ±0.2	**80.2** ±0.3	**86.9**

C.6.3 OfficeHome

Table C-11: **Complete domain generalization results on OfficeHome.**

Algorithm	A	C	P	R	Avg
ERM	61.3 ±0.7	52.4 ±0.3	75.8 ±0.1	76.6 ±0.3	66.5
IRM	58.9 ±2.3	52.2 ±1.6	72.1 ±2.9	74.0 ±2.5	64.3
GroupDRO	60.4 ±0.7	52.7 ±1.0	75.0 ±0.7	76.0 ±0.7	66.0
Mixup	62.4 ±0.8	54.8 ±0.6	76.9 ±0.3	78.3 ±0.2	68.1
MLDG	61.5 ±0.9	53.2 ±0.6	75.0 ±1.2	77.5 ±0.4	66.8
CORAL	65.3 ±0.4	54.4 ±0.5	76.5 ±0.1	78.4 ±0.5	68.7
MMD	60.4 ±0.2	53.3 ±0.3	74.3 ±0.1	77.4 ±0.6	66.3
DANN	59.9 ±1.3	53.0 ±0.3	73.6 ±0.7	76.9 ±0.5	65.9
CDANN	61.5 ±1.4	50.4 ±2.4	74.4 ±0.9	76.6 ±0.8	65.8
MTL	61.5 ±0.7	52.4 ±0.6	74.9 ±0.4	76.8 ±0.4	66.4
SagNet	63.4 ±0.2	54.8 ±0.4	75.8 ±0.4	78.3 ±0.3	68.1
ARM	58.9 ±0.8	51.0 ±0.5	74.1 ±0.1	75.2 ±0.3	64.8
VREx	60.7 ±0.9	53.0 ±0.9	75.3 ±0.1	76.6 ±0.5	66.4
RSC	60.7 ±1.4	51.4 ±0.3	74.8 ±1.1	75.1 ±1.3	65.5
BoDA	**65.4** ±0.1	**55.4** ±0.3	**77.1** ±0.1	**79.5** ±0.3	**69.3**

C.6.4 TerraInc

Table C-12: **Complete domain generalization results on TerraInc.**

Algorithm	L100	L38	L43	L46	Avg
ERM	49.8 ±4.4	42.1 ±1.4	56.9 ±1.8	35.7 ±3.9	46.1
IRM	54.6 ±1.3	39.8 ±1.9	56.2 ±1.8	39.6 ±0.8	47.6
GroupDRO	41.2 ±0.7	38.6 ±2.1	56.7 ±0.9	36.4 ±2.1	43.2
Mixup	**59.6** ±2.0	42.2 ±1.4	55.9 ±0.8	33.9 ±1.4	47.9
MLDG	54.2 ±3.0	44.3 ±1.1	55.6 ±0.3	36.9 ±2.2	47.7
CORAL	51.6 ±2.4	42.2 ±1.0	57.0 ±1.0	39.8 ±2.9	47.6
MMD	41.9 ±3.0	34.8 ±1.0	57.0 ±1.9	35.2 ±1.8	42.2
DANN	51.1 ±3.5	40.6 ±0.6	57.4 ±0.5	37.7 ±1.8	46.7
CDANN	47.0 ±1.9	41.3 ±4.8	54.9 ±1.7	39.8 ±2.3	45.8
MTL	49.3 ±1.2	39.6 ±6.3	55.6 ±1.1	37.8 ±0.8	45.6
SagNet	53.0 ±2.9	43.0 ±2.5	57.9 ±0.6	40.4 ±1.3	48.6
ARM	49.3 ±0.7	38.3 ±2.4	55.8 ±0.8	38.7 ±1.3	45.5
VREx	48.2 ±4.3	41.7 ±1.3	56.8 ±0.8	38.7 ±3.1	46.4
RSC	50.2 ±2.2	39.2 ±1.4	56.3 ±1.4	40.8 ±0.6	46.6
BoDA	54.0 ±0.3	**46.5** ±0.2	**59.5** ±0.3	**41.0** ±0.4	**50.2**

■ C.6.5 DomainNet

Table C-13: **Complete domain generalization results on** DomainNet**.**

Algorithm	clip	info	paint	quick	real	sketch	Avg
ERM	58.1 ±0.3	18.8 ±0.3	46.7 ±0.3	12.2 ±0.4	59.6 ±0.1	49.8 ±0.4	40.9
IRM	48.5 ±2.8	15.0 ±1.5	38.3 ±4.3	10.9 ±0.5	48.2 ±5.2	42.3 ±3.1	33.9
GroupDRO	47.2 ±0.5	17.5 ±0.4	33.8 ±0.5	9.3 ±0.3	51.6 ±0.4	40.1 ±0.6	33.3
Mixup	55.7 ±0.3	18.5 ±0.5	44.3 ±0.5	12.5 ±0.4	55.8 ±0.3	48.2 ±0.5	39.2
MLDG	59.1 ±0.2	19.1 ±0.3	45.8 ±0.7	13.4 ±0.3	59.6 ±0.2	50.2 ±0.4	41.2
CORAL	59.2 ±0.1	19.7 ±0.2	46.6 ±0.3	13.4 ±0.4	59.8 ±0.2	50.1 ±0.6	41.5
MMD	32.1 ±13.3	11.0 ±4.6	26.8 ±11.3	8.7 ±2.1	32.7 ±13.8	28.9 ±11.9	23.4
DANN	53.1 ±0.2	18.3 ±0.1	44.2 ±0.7	11.8 ±0.1	55.5 ±0.4	46.8 ±0.6	38.3
CDANN	54.6 ±0.4	17.3 ±0.1	43.7 ±0.9	12.1 ±0.7	56.2 ±0.4	45.9 ±0.5	38.3
MTL	57.9 ±0.5	18.5 ±0.4	46.0 ±0.1	12.5 ±0.1	59.5 ±0.3	49.2 ±0.1	40.6
SagNet	57.7 ±0.3	19.0 ±0.2	45.3 ±0.3	12.7 ±0.5	58.1 ±0.5	48.8 ±0.2	40.3
ARM	49.7 ±0.3	16.3 ±0.5	40.9 ±1.1	9.4 ±0.1	53.4 ±0.4	43.5 ±0.4	35.5
VREx	47.3 ±3.5	16.0 ±1.5	35.8 ±4.6	10.9 ±0.3	49.6 ±4.9	42.0 ±3.0	33.6
RSC	55.0 ±1.2	18.3 ±0.5	44.4 ±0.6	12.2 ±0.2	55.7 ±0.7	47.8 ±0.9	38.9
BoDA	**62.1** ±0.4	**20.5** ±0.7	**48.0** ±0.1	**13.8** ±0.6	**60.6** ±0.4	**51.4** ±0.3	**42.7**

C.6.6 Averages

Table C-14: **Complete domain generalization results over all DG benchmarks.**

Algorithm	VLCS	PACS	OfficeHome	TerraInc	DomainNet	**Avg**
ERM	77.5 ±0.4	85.5 ±0.2	66.5 ±0.3	46.1 ±1.8	40.9 ±0.1	63.3
IRM	78.5 ±0.5	83.5 ±0.8	64.3 ±2.2	47.6 ±0.8	33.9 ±2.8	61.6
GroupDRO	76.7 ±0.6	84.4 ±0.8	66.0 ±0.7	43.2 ±1.1	33.3 ±0.2	60.7
Mixup	77.4 ±0.6	84.6 ±0.6	68.1 ±0.3	47.9 ±0.8	39.2 ±0.1	63.4
MLDG	77.2 ±0.4	84.9 ±1.0	66.8 ±0.6	47.7 ±0.9	41.2 ±0.1	63.6
CORAL	**78.8** ±0.6	86.2 ±0.3	68.7 ±0.3	47.6 ±1.0	41.5 ±0.1	64.5
MMD	77.5 ±0.9	84.6 ±0.5	66.3 ±0.1	42.2 ±1.6	23.4 ±9.5	58.8
DANN	78.6 ±0.4	83.6 ±0.4	65.9 ±0.6	46.7 ±0.5	38.3 ±0.1	62.6
CDANN	77.5 ±0.1	82.6 ±0.9	65.8 ±1.3	45.8 ±1.6	38.3 ±0.3	62.0
MTL	77.2 ±0.4	84.6 ±0.5	66.4 ±0.5	45.6 ±1.2	40.6 ±0.1	62.9
SagNet	77.8 ±0.5	86.3 ±0.2	68.1 ±0.1	48.6 ±1.0	40.3 ±0.1	64.2
ARM	77.6 ±0.3	85.1 ±0.4	64.8 ±0.3	45.5 ±0.3	35.5 ±0.2	61.7
VREx	78.3 ±0.2	84.9 ±0.6	66.4 ±0.6	46.4 ±0.6	33.6 ±2.9	61.9
RSC	77.1 ±0.5	85.2 ±0.9	65.5 ±0.9	46.6 ±1.0	38.9 ±0.5	62.7
BoDA	78.5 ±0.3	**86.9** ±0.4	**69.3** ±0.1	**50.2** ±0.4	**42.7** ±0.1	**65.5**

Table C-15: **Ablation study on effect of adding balanced distance in BoDA.**

	VLCS-MLT	PACS-MLT	OfficeHome-MLT	TerraInc-MLT	DomainNet-MLT	**Avg**
DA	76.6 ±0.4	96.8 ±0.2	80.7 ±0.3	76.4 ±0.5	58.9 ±0.2	77.9
BoDA	**77.3** ±0.2	**97.2** ±0.1	**82.3** ±0.1	**82.3** ±0.3	**61.7** ±0.1	**80.2**
Gains	**+0.7**	**+0.4**	**+1.6**	**+5.9**	**+2.8**	**+2.3**

Table C-16: **Ablation study on effect of distance calibration coefficient $\lambda_{d,c}^{d',c'}$ in BoDA.** We vary the value of ν and report the averaged results over all five MDLT datasets.

ν	0	0.5	0.7	0.9	1	1.1	1.2	1.5	ERM
BoDA	78.9	80.1	80.0	80.2	80.1	79.8	79.6	79.2	77.6

C.7 Additional Analysis and Studies

C.7.1 Ablation Studies for BoDA

Effect of Balanced Distance. We study the effect of adding balanced distance in BoDA compared to the vanilla DA loss. As Table C-15 demonstrates, incorporating balanced distance in BoDA is essential for addressing MDLT: we observe that BoDA improves over DA by a large margin, resulting in an averaged improvements of 2.3% over all MDLT benchmarks. The improvements are especially large on datasets with severe data imbalance across domains (e.g., TerraInc-MLT).

Effect of Different Distance Calibration Coefficient $\lambda_{d,c}^{d',c'}$. We further investigate the effect of different distance calibration coefficients in BoDA. Recall that $\lambda_{d,c}^{d',c'} = \left(N_{d',c'}/N_{d,c}\right)^{\nu}$ indicates how much we would like to transfer (d, c) to (d', c'), based on their relative sample sizes. We vary the value of ν, and study its effect on BoDA performance across all MDLT datasets. Table C-16 reveals several interesting findings. First, when $\nu = 0$ (i.e., no calibration is used as the coefficient is always equal to 1), BoDA performance is lower than those with a positive ν, confirming the effectiveness of the calibrated distance. Moreover, when we vary ν between $0.5 - 1.5$, the overall performance gains are similar across different choices, where ν around 0.9 seems to achieve the best results. Finally, when compared to ERM, we demonstrate that BoDA consistently obtains notable gains across different ν.

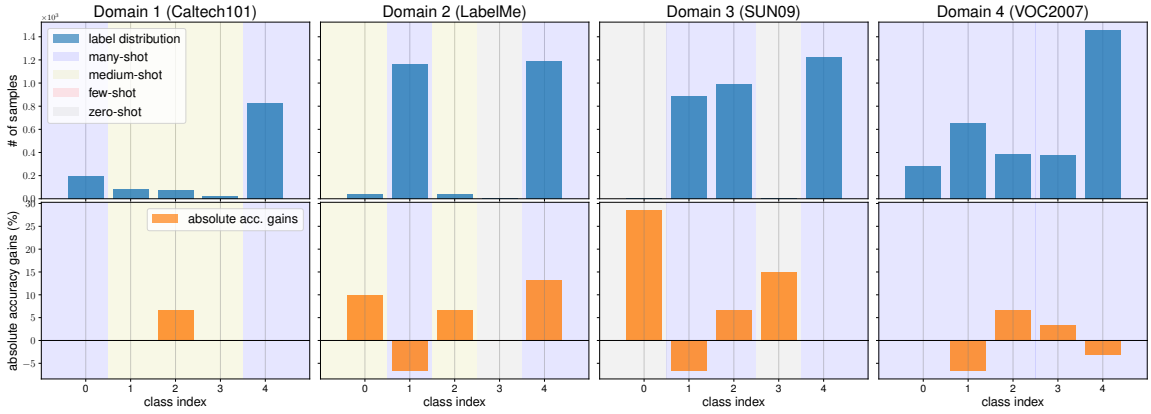


Figure C-1: **The absolute accuracy gains of** BoDA ***vs.*** **ERM over all domain-class pairs on** VLCS-MLT**.**

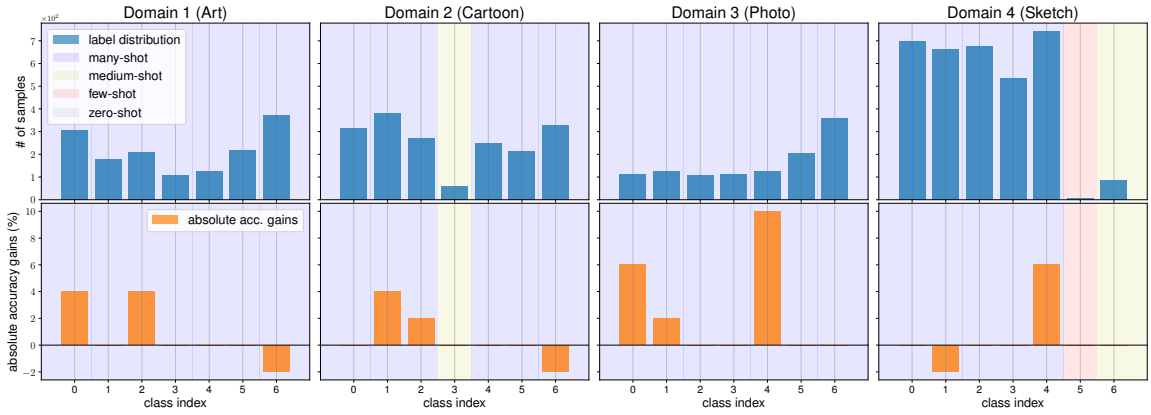


Figure C-2: **The absolute accuracy gains of** BoDA ***vs.*** **ERM over all domain-class pairs on** PACS-MLT**.**

■ C.7.2 Absolute Accuracy Gains on All MDLT Benchmarks

We provide additional results for understanding how BoDA performs across *all* domain-class pair when cross-domain imbalance occurs. Similar to Fig. 4-7 in the main text, we plot the absolute gains of BoDA over ERM on all five MDLT datasets, shown in Figs. C-1, C-2, C-3, C-4, and C-5. Across all datasets, we observe that BoDA establishes large improvements w.r.t. all regions, especially for the few-shot and zero-shot ones.

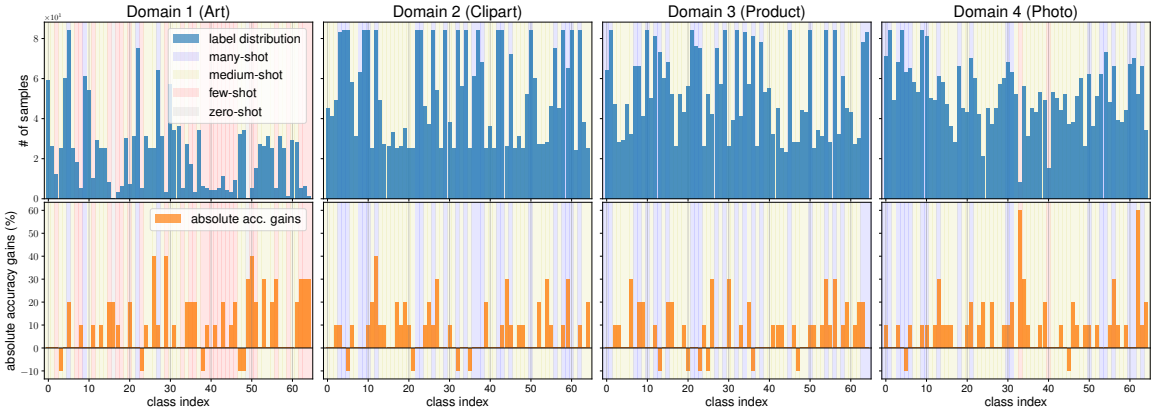


Figure C-3: **The absolute accuracy gains of** `BoDA` ***vs.*** **ERM over all domain-class pairs on** `OfficeHome-MLT`**.**

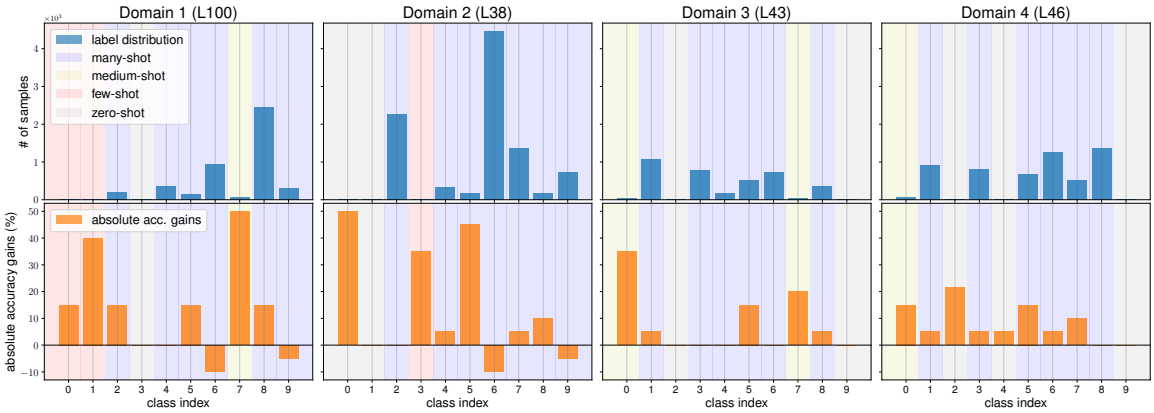


Figure C-4: **The absolute accuracy gains of** `BoDA` ***vs.*** **ERM over all domain-class pairs on** `TerraInc-MLT`**.**

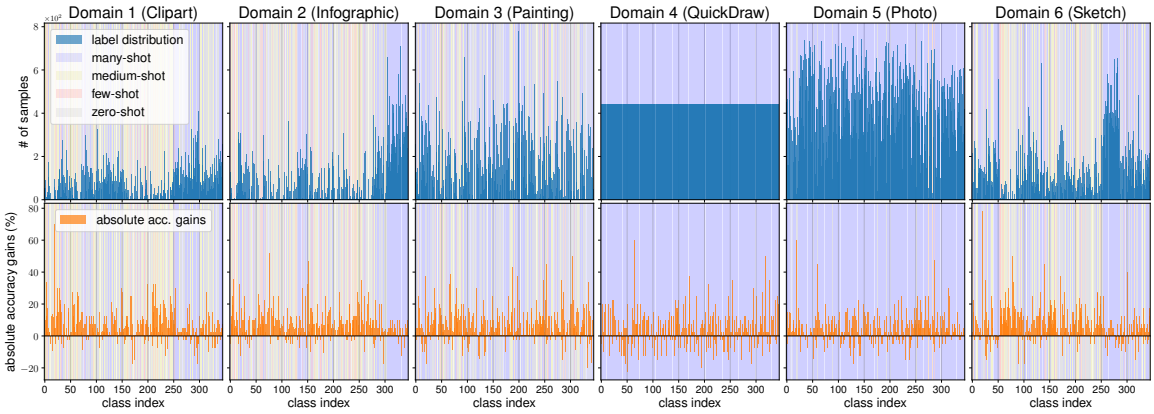


Figure C-5: **The absolute accuracy gains of** `BoDA` ***vs.*** **ERM over all domain-class pairs on** `DomainNet-MLT`**.**

■ C.7.3 Robustness to Diverse Skewed Label Distributions

We investigate how BoDA performs under arbitrary label imbalance across domains, especially when the cross-domain label distributions are both *imbalance* and *divergent*. We again employ the Digits-MLT dataset, and manually vary the label proportions for each domain.

As Fig. C-6 demonstrates, when the label distributions for two domains are balanced and identical, both ERM and BoDA maintains discriminative representations. If the label distributions become imbalanced but still identical across domains, ERM is still able to align similar classes in the two domains, but with majority classes being closer in terms of transferability than minority classes. In contrast, BoDA maintains consistent transferability regardless of number of samples within each class. Finally, as the label distributions become further mismatched across domains, ERM is not able to align the domains and produces a clear gap; by contrast, BoDA maintains consistent and transferable representations even under severe data imbalance. As a result, BoDA substantially boosts the performance upon ERM, with an average gains of 6.4% across all label configurations.

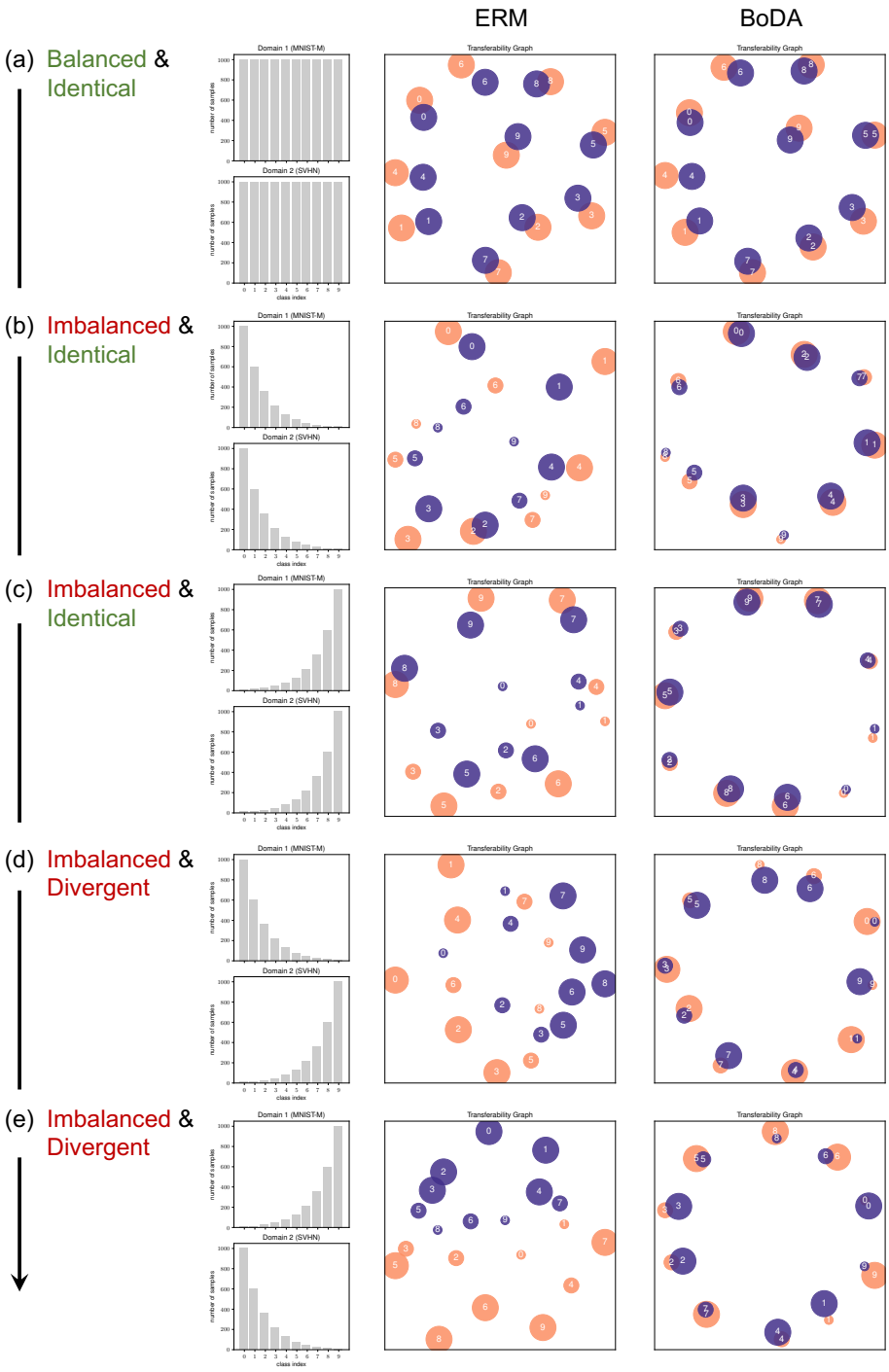


Figure C-6: **The evolving patterns of the transferability graph of** `BoDA` ***vs.*** **ERM across different label configurations on** `Digits-MLT`**.** Label distributions for two domains are **(a)** balanced and identical; **(b)(c)** imbalanced and identical; **(d)(e)** imbalanced and divergent. `BoDA` maintains consistent and transferable representations across all label configurations, and leads to much better test accuracy.

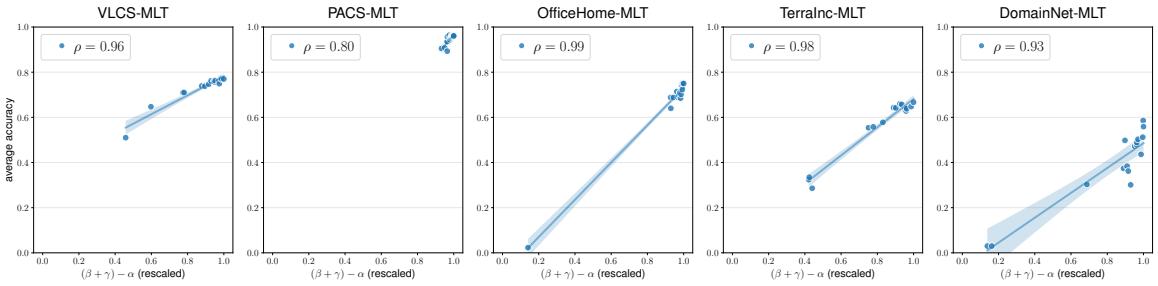


Figure C-7: **Correspondence between $(\beta+\gamma)-\alpha$ quantity and test accuracy across different MDLT datasets.** Each point within each plot corresponds to a model trained with ERM using different hyperparameters.

■ C.7.4 Transferability *vs.* Generalization on More Datasets

We provide further results on transferability statistics *vs.* generalization on real MDLT datasets, in addition to results on `Digits-MLT` as we showed in the main text.

Specifically, on all five MDLT datasets, we train 20 ERM models with varying hyperparameters, calculate the (α, β, γ) statistics for each model, and plot its classification accuracy against $(\beta+\gamma)-\alpha$. Fig. C-7 reveals similar and consistent findings, that the (α, β, γ) statistics characterize model performance in MDLT. Across all datasets, the $(\beta+\gamma)-\alpha$ quantity displays a very strong correlation with test performance across the entire range, suggesting that the (α, β, γ) statistics govern the success of learning in MDLT.

■ C.7.5 Additional Visualization of Feature Discrepancy

We provide additional results for understanding `BoDA`, i.e., how `BoDA` calibrates the feature statistics. Fig. C-8 shows the feature discrepancy of `BoDA` *vs.* ERM across different label configurations on `Digits-MLT`. In addition to the mean distance we showed in the main text, we show also the feature covariance distance between training and test data, and plot them for both domains. Similarly, solid lines plot the distance between training and test data from the same domain-class pairs. Dashed lines plot the distance between test data from a particular domain-class pair and the training data with which it shares the same class but differs in the domain. The figure also shows regions with different data densities using colors blue, yellow, red.

As the figure confirms, across different label distributions, `BoDA` consistently learns better representations especially for the tail data (i.e., the red regions), where the feature mean/covariance distance between training and test data becomes smaller and more

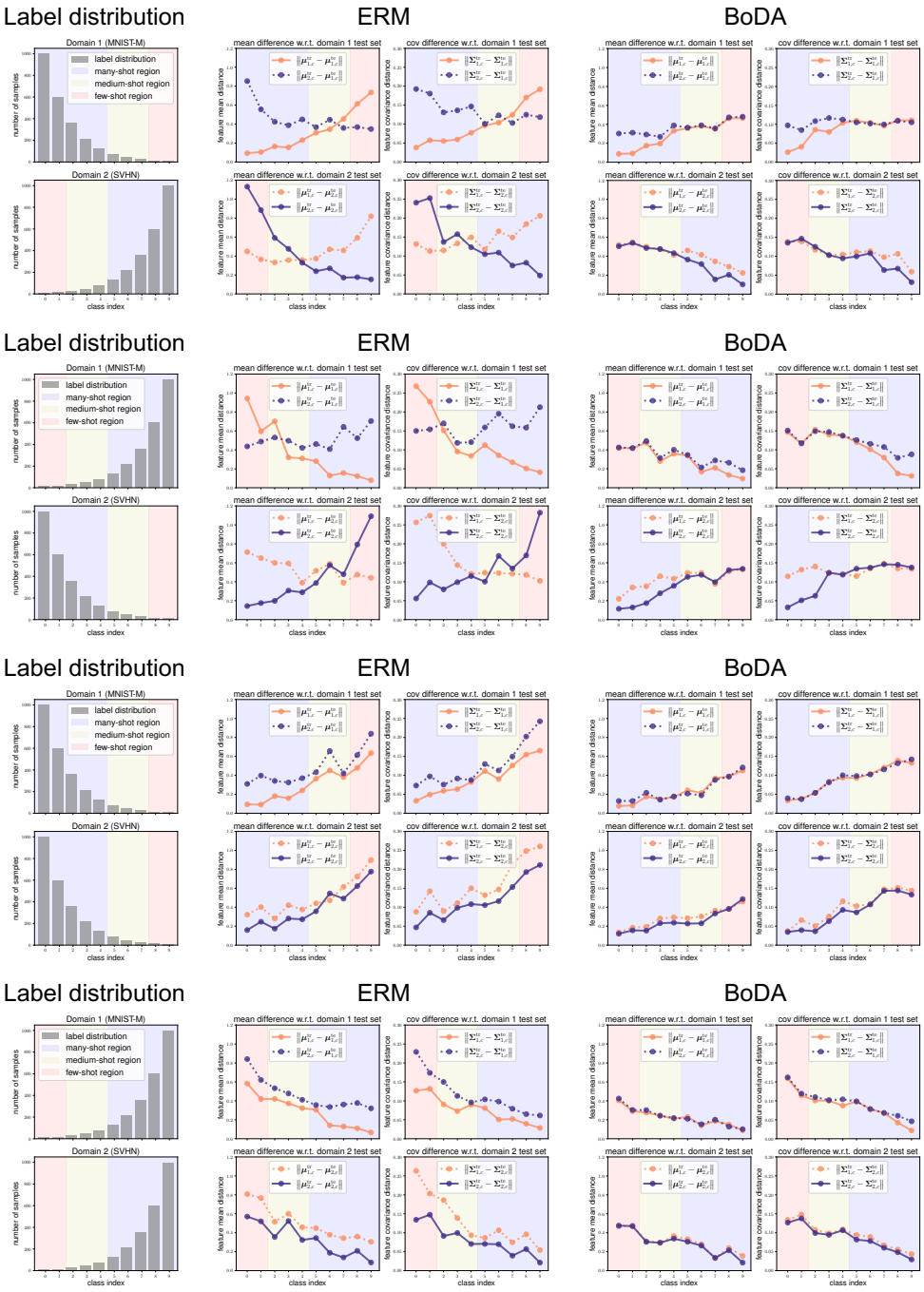


Figure C-8: **Feature discrepancy of `BoDA` *vs.* ERM across different label configurations on `Digits-MLT`.** Each row plots a per-domain label distribution, and the feature mean / covariance distance between training and test data on each domain for both ERM and `BoDA`. `BoDA` enables better learned tail (d, c) with smaller feature discrepancy.

aligned across domains. Comparing `BoDA` with ERM further demonstrates that `BoDA` maintains consistent and transferable representations with smaller feature discrepancy.

APPENDIX D
Details and Results for Subpopulation Shift Analysis

D.1 Details of the Subpopulation Shift Benchmark

D.1.1 Dataset Details

We explore subpopulation shift using **12** real-world datasets from a variety of domains including computer vision, natural language processing, and healthcare applications. We provide example inputs for each dataset in Table D-1 and Table D-2. Note that we omit showing examples for `MIMIC-CXR`, `MIMICNotes`, and `CXRMultisite` to comply with the *PhysioNet Credentialed Health Data Use Agreement*. Below, we provide detailed descriptions for each dataset in our benchmark.

`Waterbirds` [175]. `Waterbirds` is a commonly used binary classification image dataset in the spurious correlations setting, constructed by placing images from the Caltech-UCSD Birds-200-2011 (CUB) dataset [175] over backgrounds from the Places dataset [347]. The task is to classify whether a bird is a landbird or a waterbird, where the spurious attribute is the background (water or land). We use standard train/val/test splits given by prior work [160].

`CelebA` [176]. `CelebA` is a binary classification image dataset consisting of over 200,000 celebrity face images. The task, which is also used widely in the spurious correlations literature, is to predict the hair color of the person (blond *vs.* non-blond), where the spurious

Table D-1: **Example inputs for image datasets in our benchmark.** We omit showing samples for `MIMIC-CXR` and `CXRMultisite` to comply with the *PhysioNet Credentialed Health Data Use Agreement*.

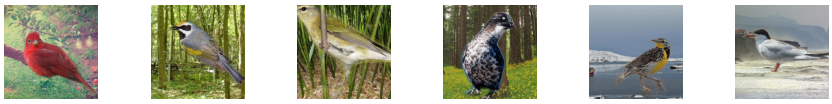
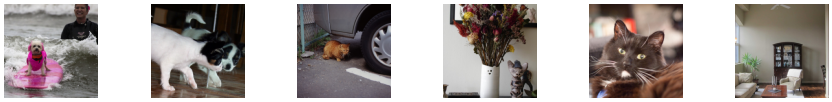
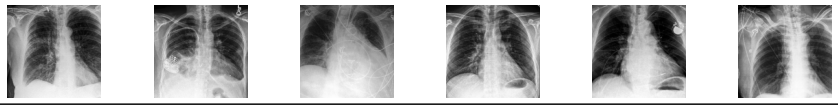
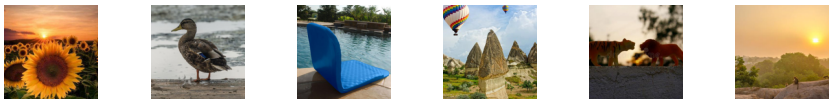
Dataset	Examples
`Waterbirds`	
`CelebA`	
`MetaShift`	
`CheXpert`	
`NICO++`	
`ImageNetBG`	
`Living17`	

Table D-2: **Example inputs for text datasets in our benchmark.** We omit showing samples for `MIMICNotes` to comply with the *PhysioNet Credentialed Health Data Use Agreement*.

Dataset	Examples
`CivilComments`	"Munchins looks like a munchins. The man who dont want to show his taxes, will tell you everything..." "The democratic party removed the filibuster to steamroll its agenda. Suck it up boys and girls." "so you dont use oil? no gasoline? no plastic? man you ignorant losers are pathetic."
`MultiNLI`	"The analysis proves that there is no link between PM and bronchitis." "Postal Service were to reduce delivery frequency." "The famous tenements (or lands) began to be built."

correlation is the gender. We also use standard dataset splits from prior work [160]. The dataset is licensed under the *Creative Commons Attribution 4.0 International* license.

`MetaShift` [177]. `MetaShift` is a general method of creating image datasets from the Visual Genome project [348]. Here, we make use of the pre-processed Cat *vs.* Dog dataset, where the goal is to distinguish between the two animals. The spurious attribute is the image

background, where cats and more likely to be indoors, and dogs are more likely to be outdoors. We use the "unmixed" version generated from the authors' codebase.

`CivilComments` **[179].** `CivilComments` is a binary classification text dataset, where the goal is to predict whether a internet comment contains toxic language. The spurious attribute is whether the text contains reference to eight demographic identities (*male, female, LGBTQ, Christian, Muslim, other religions, Black,* and *White*). We use the standard splits provided by the WILDS benchmark [142].

`MultiNLI` **[180].** `MultiNLI` is a text classification dataset with 3 classes, where the target is the natural language inference relationship between the premise and the hypothesis (neutral, contradiction, or entailment). The spurious attribute is whether negation appears in the text, as negation is highly correlated with the contradiction label. We use standard train/val/test splits given by prior work [160].

`MIMIC-CXR` **[181].** `MIMIC-CXR` is a chest X-ray dataset originating from the Beth Israel Deaconess Medical Center from Boston, Massachusetts containing over 300,000 images. We use "No Finding" as the label, where a positive label means that the patient has no illness. Inspired by prior work [291], we use the intersection of race (*White, Black, Other*) and gender as attributes. We randomly split the dataset into 85% train, 5% validation, and 10% test splits.

`CheXpert` **[182].** `CheXpert` is a chest X-ray dataset originating from the Stanford University Medical center containing over 200,000 images. We use the same data processing setup as `MIMIC-CXR`.

`CXRMultisite` **[184].** `CXRMultisite` is a dataset proposed by [184] which combines MIMIC-CXR [181] and CheXpert [182] to create a semi-synthetic spurious correlation. The task is to predict pneumonia, and the dataset is constructed such that 90% of the patients with pnuemonia are from MIMIC-CXR, and 90% of the healthy patients are from CheXpert. Thus, the site where the image was taken is the spurious correlation. We create this correlation by subsampling. We randomly split the dataset into 85% train, 5% validation, and 10% test splits.

`MIMICNotes` **[349].** `MIMICNotes` is a dataset used in a prior work [183] showing differences in error rate between demographic groups in predicting mortality from clinical notes in MIMIC-III [349]. Following their work, we reproduce their dataset which consists of fea-

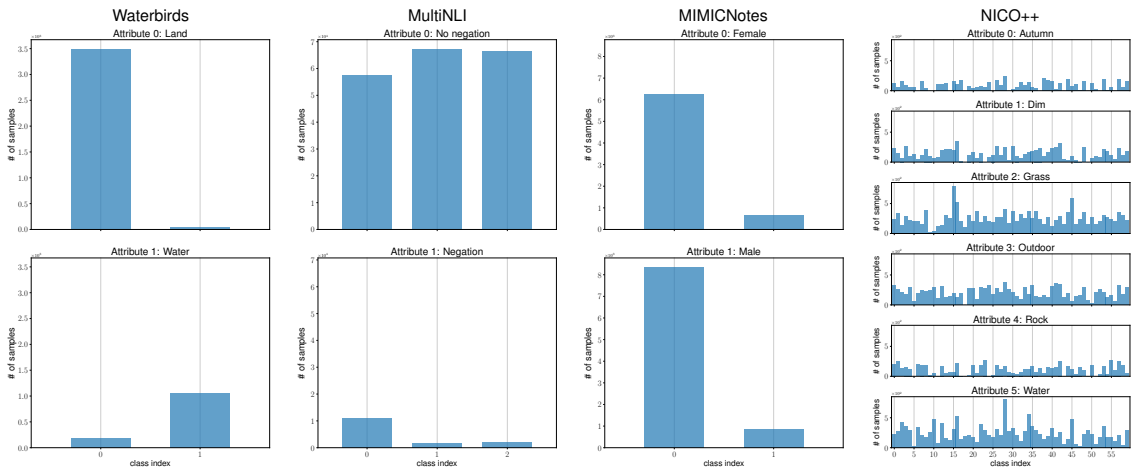
turizing the first 48 hours of clinical text from a patient's hospital stay using the top 5,000 TF-IDF features. We use gender as the attribute.

`NICO++` **[169].** `NICO++` is a large-scale benchmark for domain generalization. Here, we use data from Track 1 (the common context generalization) of their challenge. We only use their training dataset, which consists of 60 classes and 6 common attributes (*autumn, dim, grass, outdoor, rock, water*). To transform this dataset into the attribute generalization setting, we select all (attribute, label) pairs with less than 75 samples, and remove them from our training split, so they are only used for validation and testing. For each (attribute, label) pair, we use 25 samples for validation and 50 samples for testing, and use the remaining data as training samples.

`ImageNetBG` **[178].** `ImageNetBG` is a benchmark created with the goal of evaluating the reliance of ImageNet classifiers on the background. The authors first created a subset of ImageNet with 9 classes (ImageNet-9), and annotated bounding boxes so that backgrounds can be removed. In our setup, we train models on the original IN-9L (with backgrounds), and evaluate our model on `MIXED-RAND`. Note that attribute (i.e., the label of the background) is not available for this dataset. This can be thought of as an attribute generalization setting, as we do not observe test backgrounds during training.

`Living17` **[150].** `Living17` is a dataset created as part of the BREEDS benchmark for subpopulation shift. Their setup is slightly different from a traditional subpopulation shift setting, where subpopulations are defined using a WordNet hierarchy, and the goal is to generalize to unseen subclasses in the same hierarchy level. As such, it is difficult to define the notion of an "attribute" in this setting. In particular, the `Living17` dataset consists of images of living objects across 17 classes. We train our models on the source subclasses and evaluate them on the target subclasses.

Label distribution for different types of subpopulation shift. Finally, we provide typical label distributions for different subpopulation shift types in Fig. D-1. As highlighted, different shifts exhibit distinct types of label distributions, resulting in different properties in learning. For `NICO++` (Fig. D-1(d)), certain attributes have no training samples in certain classes.



(a) Spurious correlations. (b) Attribute imbalance. (c) Class imbalance. (d) Attr. generalization.

Figure D-1: **Typical label distributions for different types of subpopulation shift.**

■ D.1.2 Algorithm Details

Our benchmark contains a large number of algorithms that span different learning strategies. We group them according to their categories, and provide detailed descriptions for each algorithm below.

- *Vanilla:* The empirical risk minimization (**ERM**) [127] minimizes the sum of errors across all samples.

- *Subgroup robust methods:* Group distributionally robust optimization (**GroupDRO**) [134] performs ERM while increasing the importance of groups with larger errors. **CVaRDRO** [185] proposes a variant of GroupDRO that dynamically weights data samples that have the highest losses. **LfF** [186] trains two models simultaneously, where the first model is biased and the second one is debiased by re-weighting the gradient of the loss. Just train twice (**JTT**) [158] first trains an ERM model to identify minority groups in the training set and then trains a second ERM model with the identified samples being re-weighted. **LISA** [153] learns invariant predictors through data interpolation within and across attributes. Deep feature re-weighting (**DFR**) [154] first trains an ERM model, then retrains the last layer of the model using a balanced validation set with group annotations.

- *Data augmentation:* **Mixup** [187] performs ERM on linear interpolations of randomly sampled training examples and their labels.

- *Domain-invariant representation learning:* Invariant risk minimization (**IRM**) [66] learns

a feature representation such that the optimal linear classifier on top of that representation matches across domains. Deep correlation alignment (**CORAL**) [111] matches the mean and covariance of feature distributions. Maximum mean discrepancy (**MMD**) [137] matches the MMD [346] of feature distributions. Note that all methods in this category require group annotations during training.

- *Imbalanced learning:* **ReSample** [188] and **ReWeight** [188] simply re-sample or re-weight the inputs according to the number of samples per class. Focal loss (**Focal**) [99] reduces the relative loss for well-classified samples and focuses on difficult samples. Class-balanced loss (**CBLoss**) [72] proposes re-weighting by the inverse effective number of samples. The LDAM loss (**LDAM**) [71] employs a modified marginal loss that favors minority samples more. Balanced-Softmax (**BSoftmax**) [100] extends Softmax to an unbiased estimation that considers the number of samples in each class. Classifier re-training (**CRT**) [82] decomposes the representation and classifier learning into two stages, where it fine-tunes the classifier using class-balanced sampling with representation fixed in the second stage. **ReWeightCRT** [82] is a re-weighting variant of CRT.

■ D.1.3 Evaluation Metrics

We describe in detail all the evaluation metrics we used in our experiments.

Average & Worst Accuracy. The average accuracy is defined as the accuracy over all samples. For worst-group accuracy (WGA), we compute the accuracy over all subgroups in the test set and report the worst one. When viewing each class as a subgroup, WGA degenerates to the worst-class accuracy.

Average & Worst Precision. Precision is defined as $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, where TP is the number of true positives and FP the number of false positives. Average precision simply takes the average precision score over all classes, whereas the worst precision reports the lowest precision value across classes.

Average & Worst F1-score. The F1-score is defined as the harmonic mean of precision and recall. Average F1-score simply takes the average F1-score over all classes, whereas the worst F1-score reports the lowest value across all classes.

Adjusted Accuracy. Adjusted accuracy is defined as the average accuracy on a group-balanced dataset, which accounts for the data imbalance over subgroups.

Balanced Accuracy. Balanced accuracy is defined as the average of recall obtained on each class, taking the imbalance over classes into account.

AUROC. Following the common evaluation practice for the medical datasets used in our benchmark [181, 182], we also include the area under the receiver operating characteristic curve (AUROC) for evaluation.

ECE [189]. The expected calibration error (ECE) is defined as the difference in expected accuracy and expected confidence, which measures how close the output pseudo-probabilities match with the actual probabilities of a correct prediction (lower the better).

D.1.4 Model Selection Protocol

There has been an increasing interest in model selection within the literature on out-of-distribution generalization [133]. In subpopulation shift, model selection becomes essential especially when attributes are completely unknown in both training and validation set. Significant drop (over 20%) in worst-group test accuracy has been reported if using the highest *average* validation accuracy as the model selection criterion without any group annotations [154].

Our benchmark provides different model selection strategies based on various evaluation metrics as described in Appendix D.1.3. Throughout the thesis, we mainly use *worst-group accuracy* as the metric for model selection (which degenerates to *worst-class accuracy* when attributes are unknown in the validation set). Nevertheless, one can specify any aforementioned metric during model selection stage for experimenting with different selection strategies.

D.2 Experimental Settings

D.2.1 Implementation Details

Following [133, 154], we use pretrained ResNet-50 model [64] as the backbone network for image datasets (except for `Living17`, which we train from scratch), and use pretrained BERT model [160] for all text datasets. We employ a three-layer MLP for `MIMICNotes` dataset given its simplicity. For all image datasets, we follow standard pre-processing steps [160]: resize and center crop the image to 224×224 pixels, and perform normalization using the ImageNet channel statistics. Following the literature [154, 160], we use the

Table D-3: **Hyperparameters search space for all experiments.**

Condition	Parameter	Default value	Random distribution
General:			
ResNet	learning rate	0.001	$10^{\text{Uniform}(-4,-2)}$
	batch size	108	$2^{\text{Uniform}(6,7)}$
BERT	learning rate	0.00001	$10^{\text{Uniform}(-5.5,-4)}$
	batch size	32	$2^{\text{Uniform}(3,5.5)}$
	dropout	0.5	$\text{RandomChoice}([0, 0.1, 0.5])$
MLP	learning rate	0.001	$10^{\text{Uniform}(-4,-2)}$
	batch size	256	$2^{\text{Uniform}(7,10)}$
Algorithm-specific:			
IRM	lambda	100	$10^{\text{Uniform}(-1,5)}$
	iterations of penalty annealing	500	$10^{\text{Uniform}(0,4)}$
GroupDRO	eta	0.01	$10^{\text{Uniform}(-3,-1)}$
Mixup	alpha	0.2	$10^{\text{Uniform}(0,4)}$
CVaRDRO	alpha	0.1	$10^{\text{Uniform}(-2,0)}$
JTT	first stage step fraction	0.5	$\text{Uniform}(0.2, 0.8)$
	lambda	10	$10^{\text{Uniform}(0,2.5)}$
LISA	alpha	2	$10^{\text{Uniform}(-1,1)}$
	p_select	0.5	$\text{Uniform}(0, 1)$
LfF	q	0.7	$\text{Uniform}(0.05, 0.95)$
DFR	regularization	0.1	$10^{\text{Uniform}(-2,0.5)}$
CORAL, MMD	gamma	1	$10^{\text{Uniform}(-1,1)}$
Focal	gamma	1	$0.5 * 10^{\text{Uniform}(0,1)}$
CBLoss	beta	0.9999	$1 - 10^{\text{Uniform}(-5,-2)}$
LDAM	max_m	0.5	$10^{\text{Uniform}(-1,-0.1)}$
	scale	30	$\text{RandomChoice}([10, 30])$

AdamW optimizer [345] for all text datasets, and use SGD with momentum for all image datasets. We train all models for 5,000 steps on `Waterbirds` and `MetaShift`, 10,000 steps on `MIMICNotes` and `ImageNetBG`, 20,000 steps on `CheXpert` and `CXRMultisite`, and 30,000 steps on all other datasets to ensure convergence.

■ D.2.2 Hyperparameters Search Protocol

For a fair evaluation across different algorithms, following the training protocol in [133], for each algorithm we conduct a random search of 16 trials over a joint distribution of its all hyperparameters. We then use the validation set to select the best hyperparameters for each algorithm, fix them and rerun the experiments under 3 different random seeds to

report the final average results (and standard deviation). Such process ensures the comparison is best-versus-best, and the hyperparameters are optimized for all algorithms.

We detail the hyperparameter choices for each algorithm in Table D-3.

■ D.3 Additional Analysis and Studies

■ D.3.1 Quantifying the Degree of Different Shifts

In order to quantify the degree of each shift for each dataset relative to others, we use several simple metrics (see Table D-4, Table D-5, and Table D-6). For spurious correlations, we use:

- The Mutual Information (**MI**) between A and Y, $I(A;Y)$.
- The Normalized Mutual Information (**NMI**) between A and Y, where norm $I(A;Y) = 1$ indicates that the two are perfectly correlated:

$$\text{norm } I(A;Y) = \frac{2I(A;Y)}{H(Y) + H(A)}.$$

- **Cramer's V**, which is an association measure based on the Chi-squared test statistic. It has a range of $[0, 1]$, where 1 indicates perfect correlation.
- **Tschuprow's T**, which is closely related to Cramer's V. It also has a range of $[0, 1]$.

Note that we only examine the correlation between A and Y, but not the degree of effectiveness to which A can be inferred from X. This is an important component, as the model can not take advantage of the spurious correlation if it could not be learnt easily. However, we would expect that most attributes (e.g., words in text, image backgrounds) should be easily inferred from the inputs for the datasets we examine.

For attribute and class imbalance, we use the following metrics (shown for the class imbalance case):

- **Entropy**: $H(Y)$.
- **Normalized Entropy**, where norm $H(Y) = 1$ means that the distribution is uniform (i.e., no imbalance):

$$\text{norm } H(Y) = \frac{H(Y)}{\log |\text{supp}(Y)|}.$$

Table D-4: **Metrics for quantifying the degree of *spurious correlations*.**

Dataset	MI$^{\uparrow}$	NMI$^{\uparrow}$	Cramer$^{\uparrow}$	Tschuprow$^{\uparrow}$
Waterbirds	0.37	0.67	0.87	0.87
CelebA	0.06	0.11	0.31	0.31
MetaShift	0.09	0.13	0.41	0.41
CivilComments	0.02	0.02	0.19	0.11
MultiNLI	0.03	0.04	0.25	0.21
MIMIC-CXR	0.01	0.01	0.15	0.10
MIMICNotes	$< 1e^{-4}$	$< 1e^{-4}$	0.01	0.01
CXRMultisite	0.03	0.13	0.32	0.32
CheXpert	$< 1e^{-3}$	$< 1e^{-3}$	0.03	0.02
NICO++	0.11	0.04	0.20	0.11
ImageNetBG	–	–	–	–
Living17	–	–	–	–

Table D-5: **Metrics for quantifying the degree of *attribute imbalance*.**

Dataset	Entropy$^{\downarrow}$	N. Entropy$^{\downarrow}$	$p_{\text{max}} - p_{\text{min}}{}^{\uparrow}$
Waterbirds	0.82	0.82	0.48
CelebA	0.98	0.98	0.16
MetaShift	0.99	0.99	0.14
CivilComments	2.78	0.93	0.20
MultiNLI	0.37	0.37	0.86
MIMIC-CXR	2.33	0.90	0.27
MIMICNotes	0.99	0.99	0.14
CXRMultisite	0.51	0.51	0.77
CheXpert	2.20	0.85	0.32
NICO++	2.47	0.96	0.17
ImageNetBG	–	–	–
Living17	–	–	–

- Difference between the probability of the most frequent class and the probability of the least frequent class ($p_{\text{max}} - p_{\text{min}}$).

For attribute generalization, we simply examine whether there exist any subpopulations in the test set which do not appear during training.

Table D-6: **Metrics for quantifying the degree of *class imbalance*.**

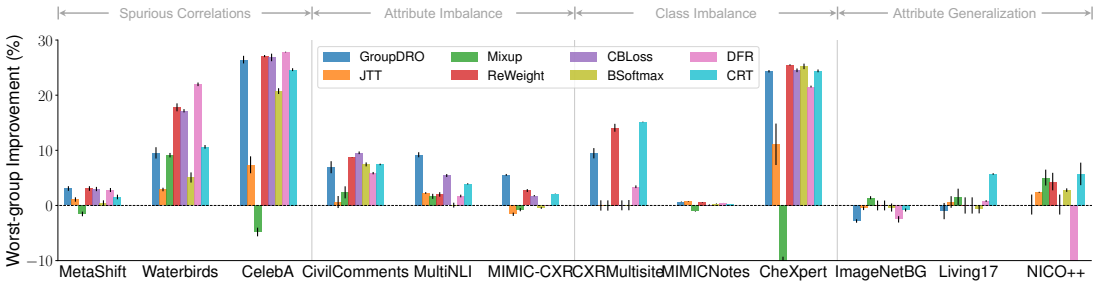
Dataset	Entropy$^{\downarrow}$	N. Entropy$^{\downarrow}$	$p_{\max} - p_{\min}{}^{\uparrow}$
Waterbirds	0.78	0.78	0.54
CelebA	0.61	0.61	0.70
MetaShift	0.99	0.99	0.13
CivilComments	0.67	0.67	0.65
MultiNLI	1.58	0.99	0.001
MIMIC-CXR	0.97	0.97	0.20
MIMICNotes	0.45	0.45	0.81
CXRMultisite	0.12	0.12	0.97
CheXpert	0.47	0.47	0.80
NICO++	5.81	0.98	0.03
ImageNetBG	3.17	1	0
Living17	4.09	1	0

■ D.3.2 Improvements across Different Shifts & Settings

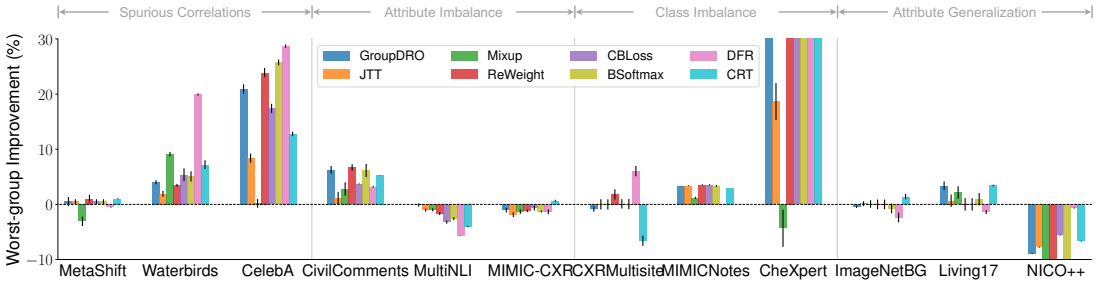
We show in Fig. D-2 the complete results on worst-group performance improvements over ERM under different settings. As can be observed from all figures, algorithmic advances have been made for *spurious correlations* and *class imbalance*, where consistent improvements can be obtained across different training & validation settings. Yet, small overall improvements are observed for *attribute imbalance*, while almost no performance gains can be obtained for *attribute generalization*, indicating the limitation of SOTA algorithms on tackling these types of subpopulation shift.

■ D.3.3 Model Selection without Validation Attributes

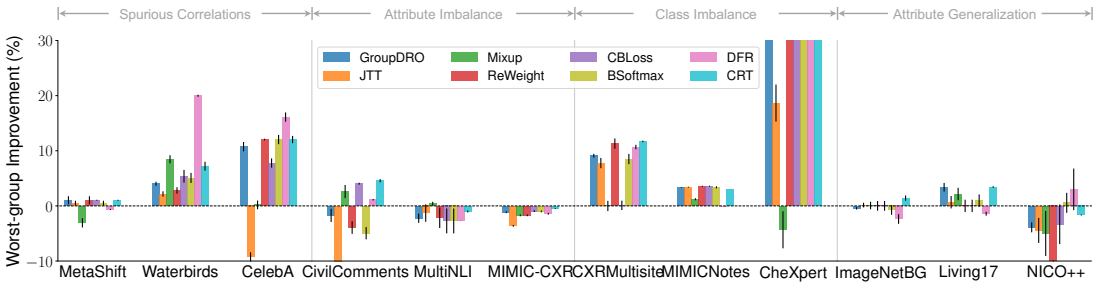
In the main thesis, we examine the feasibility of different metrics for model selection without group-annotated validation data. We further confirm this in Table D-7 by showing the results for more selection strategies with all metrics across all datasets in our benchmark. Specifically, when using *worst-class accuracy* as the model selection criterion, on average we achieve only **2.4%** degrade of worst-group accuracy compared to oracle selection method. The selection criterion also performs the best over all other selection metrics on 10 out of



(a) Train & validation attributes both known (*oracle selection*).



(b) Train attributes unknown, but validation attributes known (*worst-group accuracy selection*).



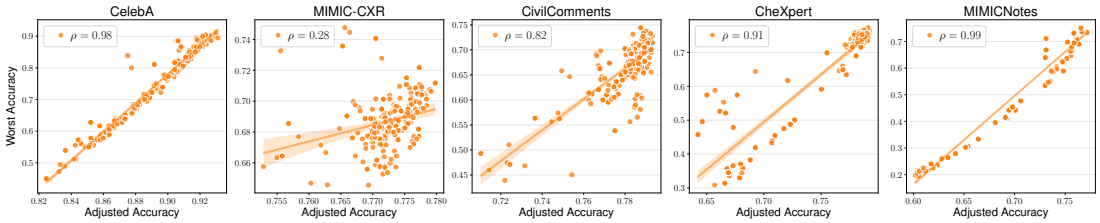
(c) Train & validation attributes both unknown (*worst-class accuracy selection*).

Figure D-2: **Complete results on worst-group performance improvements over ERM under different settings.**

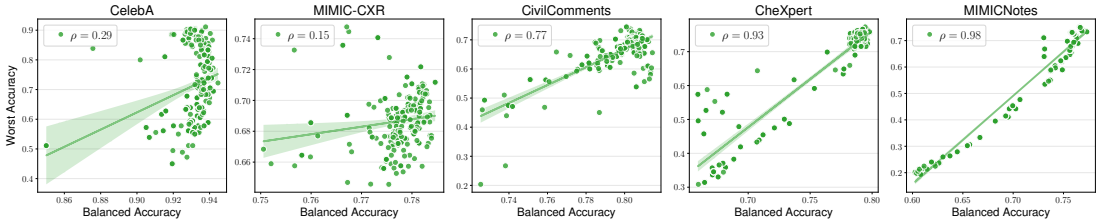
Table D-7: **Test-set worst-group accuracy difference (%) between each selection strategy on each dataset, relative to the oracle which selects the best test-set worst-group accuracy.** Note that we have only defined AUPRC and Brier score for the binary classification case.

Selection Strategy	CXRMultisite	CelebA	CheXpert	CivilComments	ImageNetBG	Living17	MIMIC-CXR	MIMICNotes	MetaShift	MultiNLI	NICO++	Waterbirds	Avg
Max Worst-Class Accuracy	-6.9 ±10.7	-5.0 ±6.3	**-0.4** ±0.8	**-3.2** ±5.2	**-0.7** ±1.3	**-1.6** ±2.3	**-0.9** ±1.0	**-0.1** ±0.5	**-1.5** ±3.0	**-1.9** ±2.9	**-5.3** ±5.6	**-0.8** ±1.4	**-2.4**
Max Balanced Accuracy	-6.9 ±10.7	-4.4 ±5.4	-1.3 ±2.5	-3.5 ±5.8	-0.9 ±1.6	-4.5 ±5.4	-2.9 ±4.9	-2.3 ±6.2	-1.7 ±3.0	-3.7 ±3.9	-7.0 ±5.8	-1.3 ±1.9	-3.4
Min Class Accuracy Diff	**-6.2** ±10.3	-6.1 ±9.1	-1.9 ±5.3	-4.1 ±8.0	-2.8 ±13.0	-5.1 ±10.0	-1.9 ±5.0	-0.3 ±1.2	-2.2 ±4.6	-5.7 ±8.6	-27.2 ±15.4	-2.4 ±4.8	-5.5
Max Worst-Class F1	-7.7 ±11.3	-13.4 ±10.4	-5.4 ±6.7	-3.2 ±3.8	-0.8 ±1.2	-3.5 ±4.4	-2.5 ±2.2	-4.4 ±8.7	-1.8 ±3.3	-2.3 ±3.0	-6.7 ±6.3	-2.6 ±3.5	-4.5
Max Macro Avg F1	-8.2 ±11.6	-14.3 ±10.6	-7.7 ±9.8	-5.1 ±4.7	-0.9 ±1.5	-4.4 ±5.3	-2.8 ±4.5	-8.2 ±13.2	-1.8 ±2.9	-3.3 ±3.4	-7.0 ±5.8	-3.1 ±4.0	-5.6
Min Per-Class Recall Stdev.	-6.2 ±10.3	-6.1 ±9.1	-1.9 ±5.3	-4.1 ±8.0	-2.3 ±11.5	-5.5 ±9.1	-1.9 ±5.0	-0.3 ±1.2	-2.2 ±4.6	-5.6 ±8.7	-29.7 ±14.3	-2.4 ±4.8	-5.7
Max Weighted Avg Precision	-8.3 ±11.5	-13.5 ±10.1	-6.3 ±11.1	-5.7 ±8.6	-0.8 ±1.3	-7.5 ±7.8	-4.3 ±6.4	-12.6 ±21.5	-3.3 ±8.0	-3.4 ±4.7	-6.8 ±5.5	-4.9 ±10.1	-6.5
Max Overall AUROC	-10.0 ±12.5	-12.2 ±10.3	-10.4 ±13.0	-8.2 ±9.0	-1.1 ±2.1	-5.5 ±6.7	-6.6 ±9.9	-10.0 ±16.5	-3.2 ±7.0	-4.4 ±5.8	-6.9 ±6.3	-2.6 ±6.1	-6.7
Max Overall AUPRC	-10.0 ±12.5	-13.0 ±10.3	-11.6 ±11.9	-8.1 ±8.9	-	-	-7.3 ±10.2	-9.6 ±16.3	-2.7 ±6.2	-	-	-4.0 ±9.5	-8.3
Min Overall BCE	-8.2 ±11.5	-18.1 ±13.2	-18.7 ±16.4	-13.1 ±12.3	-0.9 ±1.6	-7.2 ±7.3	-7.2 ±12.0	-14.3 ±20.7	-3.7 ±7.7	-6.2 ±7.8	-7.6 ±6.1	-12.5 ±18.4	-9.8
Max Per-class Precision	-8.2 ±11.7	**-3.0** ±8.9	-6.8 ±12.5	-14.8 ±24.3	-7.6 ±18.4	-19.3 ±15.9	-9.4 ±12.7	-12.6 ±22.4	-9.9 ±17.4	-6.6 ±10.1	-14.8 ±11.8	-5.3 ±12.4	-9.8
Max Overall Accuracy	-8.2 ±11.4	-18.6 ±12.0	-30.9 ±24.9	-13.7 ±9.5	-0.9 ±1.6	-4.5 ±5.4	-5.1 ±6.3	-19.9 ±26.0	-1.9 ±3.3	-3.7 ±3.9	-7.1 ±5.8	-7.2 ±11.7	-10.2
Min Overall Brier Score	-8.2 ±11.5	-18.8 ±13.1	-19.6 ±16.6	-13.5 ±12.3	-	-	-7.1 ±12.0	-15.1 ±21.6	-2.7 ±5.3	-	-	-6.9 ±11.0	-11.5
Min Overall ECE	-8.2 ±11.5	-20.5 ±15.7	-20.3 ±17.4	-14.4 ±13.5	-16.9 ±33.6	-28.8 ±19.6	-12.3 ±18.2	-16.2 ±22.7	-20.9 ±28.8	-24.6 ±19.0	-20.0 ±14.3	-11.0 ±17.9	-17.9

12 datasets, indicating its effectiveness for reliable model selection without *any* attribute information.

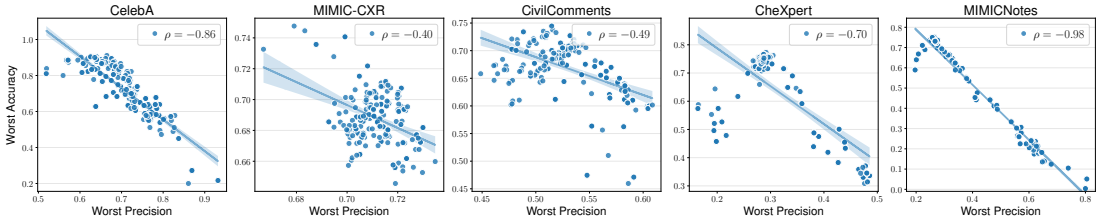


(a) Adjusted accuracy.

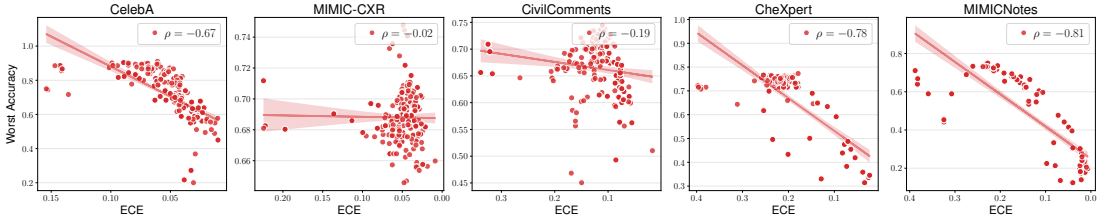


(b) Balanced accuracy.

Figure D-3: **Accuracy on the line.** We show metrics that are *positively* correlated with worst-group accuracy.



(a) Worst precision.



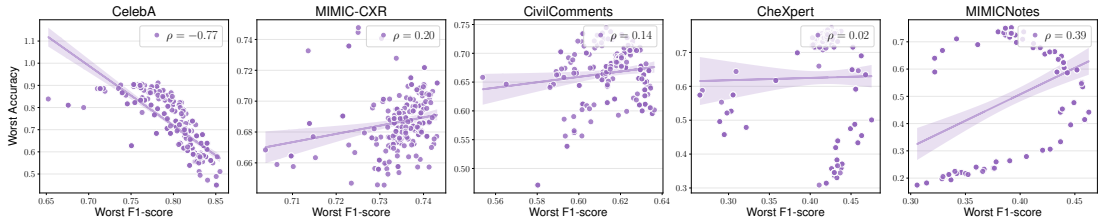
(b) ECE.

Figure D-4: **Accuracy on the inverse line.** We show metrics that are *negatively* correlated with worst-group accuracy.

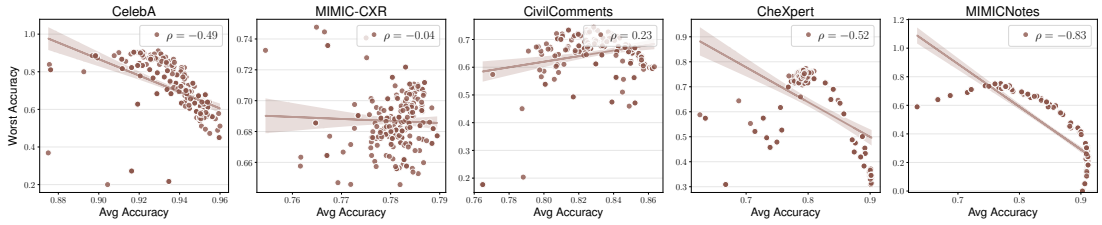
■ D.3.4 Rethinking Evaluation Metrics in Subpopulation Shift

We provide complete results on the correlation between worst-group accuracy (WGA) and other metrics we consider in our benchmark.

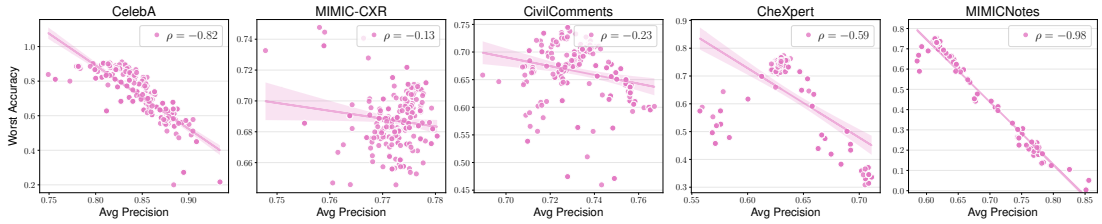
Accuracy on the line. In the main thesis we show that certain metrics exhibit high linear correlation with WGA. We further show in Fig. D-3 with a full list of metrics that exhibit consistent positive correlation across diverse datasets. Specifically, both adjusted accuracy and balanced accuracy display the "*accuracy on the line*" property, which has also been



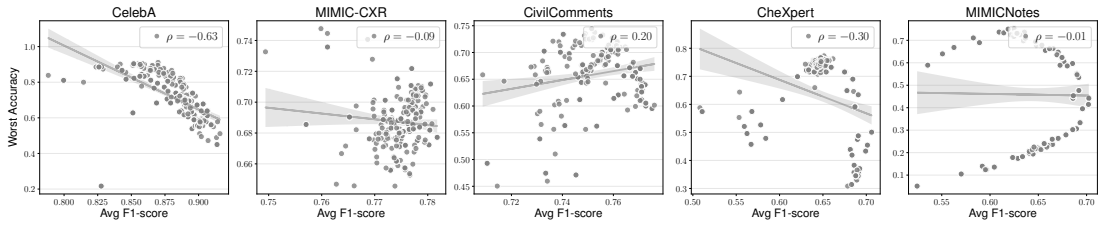
(a) Worst F1-score.



(b) Average accuracy.



(c) Average precision.



(d) Average F1-score.

Figure D-5: **Accuracy not on the line.** We show metrics that do not demonstrate consistent correlations across datasets with worst-group accuracy.

confirmed in prior work [154].

Accuracy on the inverse line. More interestingly, we further establish the intrinsic trade-off between WGA and certain metrics. Fig. D-4 shows that both worst-case precision and ECE exhibit clear *negative* correlation with WGA, demonstrating the fundamental trade-off between WGA and several important metrics in subpopulation shift. These intriguing observations highlight the need for considering more realistic evaluation metrics in subpopulation shift beyond just using WGA.

Accuracy not on the line. Finally, we display also other metrics that do not show either positive or negative correlation with WGA (Fig. D-5). As observed, the correlation between

these metrics and WGA shows inconsistent behavior across datasets. Interestingly, this phenomenon also indicates the potential bad performance on these metrics when merely optimizing for better WGA. We leave the exploration of other metrics and the rationale behind these behaviors for future work.

■ D.3.5 Impact of Architecture, Pretraining Method, and Pretraining Dataset

In this section, we examine the impact of model architecture and the source of the initial model weights on the worst group accuracy. Similar to the experiments above, we consider the following settings:

- **Known Attributes.** Attributes are known in both training and validation, and validation set worst-group accuracy is used as the model selection criteria.
- **Unknown Attributes.** Attributes are unknown during training and validation. Following our findings in Sec. 5.4.4, we use worst-class accuracy as the model selection criteria.

We experiment with ERM, JTT, and DFR as representative methods; `CivilComments` as the representative text dataset, and `Waterbirds`, `CheXpert`, and `NICO++` as representative image datasets.

For the text modality, we consider the following architectures and initial weights:

- **BERT**$_{\text{BASE}}$ [350]: A contextual language model based on the transformer architecture pretrained on BookCorpus and English Wikipedia data using the masked language model and next sentence prediction tasks.
- **SciBERT** [351]: Same architecture as BERT$_{\text{BASE}}$, but pretrained on scientific papers from Semantic Scholar, and has higher reported performance on scientific NLP tasks.
- **DistilBERT** [352]: A knowledge distilled [353] version of BERT$_{\text{BASE}}$ with 40% fewer parameters, pretrained using the same datasets as BERT$_{\text{BASE}}$.
- **GPT-2** [354]: An autoregressive language model based on the transformer decoder, pretrained using text from webpages upvoted on Reddit.
- **RoBERTa**$_{\text{BASE}}$ [355]: Same architecture as BERT$_{\text{BASE}}$, but pretrained with a more efficient procedure and using a collection of corpora much larger than BERT$_{\text{BASE}}$.

For the image modality, we consider **ResNet-50** [64] and vision transformers (**ViT-B**) [356]. We consider model weights initialized with the following pretraining methods that span supervised and self-supervised manners:

- **Supervised** pretraining [357].
- **SimCLR** [36]: Self-supervised contrastive pretraining using image augmentations.
- **Barlow** Twins [358]: Self-supervised pretraining via redundancy reduction.
- **DINO** [359]: Self-distillation with no labels.
- **CLIP** [360]: Using associated text as supervision. We select only the vision encoder.

We consider model weights initialized using the above pretraining methods on the following pretraining datasets:

- **ImageNet-1K** [170]: 1.2 million images belonging to 1,000 classes, introduced as part of the ILSVRC2012 visual recognition challenge [361].
- **ImageNet-21K** [192]: A superset of ImageNet-1K, consisting of 14 million images belonging to 21,841 classes.
- **SWAG** [193]: 3.6 billion images collected from public Instagram posts, weakly supervised using their associated hashtags.
- **LAION-2B** [362]: 2.32 billion English image-text pairs constructed from Common Crawl.
- **OpenAI-CLIP** [360]: 400 million image-text pairs collected by OpenAI in training their CLIP model.

As model weights for many combinations of the above architectures, pretraining methods, and pretraining datasets are not available, we only experiment with the subset of combinations of weights that exist in public repositories.

Based on our experimental results on `CivilComments` (Table D-8), we find that $\text{BERT}_{\text{BASE}}$ is competitive in performance, even outperforming its successor $\text{RoBERTa}_{\text{BASE}}$ on many tasks. In addition, DistilBERT and GPT-2 exhibits much worse performance especially on ERM models.

Based on our experimental results on image datasets (Tables D-9 and D-10), we find the following:

Table D-8: **Test-set worst-group accuracy on `CivilComments` for different text architectures and pretraining methods.**

Arch	Unknown Attributes			Known Attributes		
	ERM	JTT	DFR	ERM	JTT	DFR
BERT	**65.6**	**69.6**	62.4	**66.2**	65.0	**69.7**
SciBERT	61.1	58.3	**62.5**	61.1	58.3	68.0
DistilBERT	51.8	55.1	61.8	59.6	66.2	67.6
GPT-2	14.7	49.0	51.7	14.7	49.0	51.9
RoBERTa	61.0	58.0	61.6	63.1	**66.7**	68.2

Table D-9: **Test-set worst-group accuracy for three image datasets with *known attributes*, varying the model architecture and source of model initial weights.** Best results of each column are in **bold** and the second best are underlined.

Arch	Pretrain Method	Pretrain Dataset	CheXpert			NICO++			Waterbirds			Avg
			ERM	JTT	DFR	ERM	JTT	DFR	ERM	JTT	DFR	
ResNet	Barlow	ImageNet-1K	46.2	66.0	74.7	40.0	40.0	20.0	67.3	72.4	88.3	57.2
	DINO	ImageNet-1K	43.0	71.5	72.8	39.5	40.0	4.0	72.9	72.5	89.1	56.1
	SimCLR	ImageNet-1K	47.9	**72.3**	**74.8**	30.0	30.0	16.0	70.1	68.1	81.2	54.5
	Supervised	ImageNet-1K	**59.2**	61.7	72.2	25.0	30.0	20.0	76.5	74.3	**90.2**	56.6
	Supervised	ImageNet-21K	51.4	68.0	70.0	40.0	46.0	**40.0**	74.5	75.9	**90.2**	61.8
ViT-B	CLIP	Laion-2B	49.2	58.5	69.1	33.3	40.0	33.3	39.6	46.9	75.5	49.5
	CLIP	OpenAI-CLIP	42.2	55.8	68.8	33.3	40.0	30.0	40.4	40.4	78.2	47.7
	DINO	ImageNet-1K	43.4	71.8	72.4	30.0	40.0	32.0	63.9	64.6	**90.2**	56.5
	Supervised	ImageNet-1K	40.4	64.5	70.1	20.0	33.3	0.0	51.2	52.6	80.4	45.8
	Supervised	ImageNet-21K	47.5	69.1	69.1	48.0	**50.0**	18.0	69.9	73.8	87.2	59.2
	Supervised	SWAG	48.7	67.3	72.5	**50.0**	**50.0**	34.0	**82.7**	**81.2**	87.5	**63.8**

Table D-10: **Test-set worst-group accuracy for three image datasets with *unknown attributes*, varying the model architecture and source of model initial weights.** Best results of each column are in **bold** and the second best are underlined.

Arch	Pretrain Method	Pretrain Dataset	CheXpert			NICO++			Waterbirds			Avg
			ERM	JTT	DFR	ERM	JTT	DFR	ERM	JTT	DFR	
ResNet	Barlow	ImageNet-1K	46.2	66.0	73.7	33.3	40.0	40.0	67.3	72.4	89.8	58.7
	DINO	ImageNet-1K	43.0	71.5	73.3	39.5	40.0	12.0	72.9	72.5	87.9	57.0
	SimCLR	ImageNet-1K	47.9	**72.3**	74.6	30.0	30.0	26.0	70.1	69.0	79.2	55.5
	Supervised	ImageNet-1K	**59.2**	61.7	**75.4**	40.0	30.0	33.3	67.0	74.3	89.6	58.9
	Supervised	ImageNet-21K	45.3	69.3	69.9	40.0	40.0	40.0	74.5	75.9	88.3	60.4
ViT-B	CLIP	Laion-2B	49.2	58.5	69.7	30.0	30.0	40.0	45.2	46.9	78.4	49.8
	CLIP	OpenAI-CLIP	42.2	57.4	70.4	33.3	40.0	40.0	26.5	44.4	77.4	48.0
	DINO	ImageNet-1K	43.4	69.4	72.3	40.0	41.2	37.5	63.9	64.6	**90.0**	58.0
	Supervised	ImageNet-1K	40.4	69.5	71.5	33.3	33.3	16.7	49.4	52.6	81.2	49.8
	Supervised	ImageNet-21K	47.5	69.7	71.3	**50.0**	**50.0**	38.0	69.9	73.8	88.9	62.1
	Supervised	SWAG	52.5	63.8	71.3	**50.0**	**50.0**	**50.0**	**82.7**	**81.2**	88.6	**65.6**

- **Optimal architecture is dataset dependent.** Contrary to prior work [363], we find mixed results when comparing the worst-group performance for ResNet and ViT-B. Specifically, ResNets seem to work better on `CheXpert` and `Waterbirds`, while vision transformers work better on `NICO++`.

- **Supervised pretraining outperforms others.** Similar to prior work [154], we find that supervised pretraining outperforms self-supervised learning for the most part, though some self-supervised pretraining methods are still competitive. The results also warrant better self-supervised schemes for subgroup shifts [1].

- **Larger pretraining datasets yield better results.** The biggest impact on worst-group accuracy by far appears to be the dataset on which the initial model weights are derived. This is especially true for `NICO++` and `Waterbirds`, where going from ImageNet-1K to ImageNet-21K to SWAG almost always leads to a significant increase in worst-group accuracy, indicating that larger and more diverse pretraining datasets seem to increase performance. The effectiveness of SWAG-pretrained ViTs on `Waterbirds` has also been discussed in prior work [364].

■ D.4 Complete Results

We provide complete evaluation results in this section. As confirmed earlier, model selection and attribute availability play critical roles in subpopulation shift evaluation. To provide a thorough analysis, we investigate the following three settings:

- **Attributes are known in both training & validation (Appendix D.4.1).** When attributes are known in both training and validation set, which corresponds to the most ideal scenario, we use "*test set worst-group accuracy*" as an oracle selection method to identify the best possible performance for each algorithm.

- **Attributes are unknown in training, but known in validation (Appendix D.4.2).** When attributes are still known in validation, we use "*validation set worst-group accuracy*" to select models. We ignore algorithms that require attribute information in the training set (i.e., IRM, MMD, CORAL) when reporting results under this setting.

- **Attributes are unknown in both training & validation (Appendix D.4.3).** When attributes are completely unknown, we still use "*validation set worst-group accuracy*" for model selection, which however degenerates to "*worst-class accuracy*". We again ignore algorithms that require attribute information in the training set.

■ D.4.1 Attributes Known in Both Training & Validation

Waterbirds

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	84.1 ±1.7	69.1 ±4.7	77.4 ±2.0	60.7 ±3.2	79.4 ±2.1	69.5 ±2.9	83.1 ±2.0	83.1 ±2.0	91.0 ±1.4	12.9 ±1.7
Mixup	89.5 ±0.4	78.2 ±0.4	83.9 ±0.6	71.6 ±1.3	85.9 ±0.4	78.8 ±0.6	88.9 ±0.3	88.9 ±0.3	94.7 ±0.2	7.0 ±0.6
GroupDRO	88.8 ±1.8	78.6 ±1.0	83.6 ±2.4	70.9 ±5.0	85.3 ±2.0	78.1 ±2.6	88.5 ±0.8	88.5 ±0.8	95.5 ±0.5	9.1 ±2.2
IRM	88.4 ±0.1	74.5 ±1.5	82.5 ±0.2	69.5 ±0.6	84.3 ±0.1	76.4 ±0.1	87.1 ±0.3	87.1 ±0.3	94.0 ±0.3	9.5 ±0.2
CVaRDRO	89.8 ±0.4	75.5 ±2.2	84.5 ±0.7	73.2 ±1.7	86.1 ±0.3	79.0 ±0.4	88.5 ±0.3	88.5 ±0.3	95.4 ±0.2	8.2 ±0.2
JTT	88.8 ±0.6	72.0 ±0.3	83.1 ±0.8	71.2 ±1.5	84.7 ±0.6	76.9 ±0.8	86.9 ±0.3	86.9 ±0.3	94.1 ±0.1	9.0 ±0.4
LfF	87.0 ±0.3	75.2 ±0.7	80.7 ±0.3	66.2 ±0.5	82.8 ±0.3	74.3 ±0.5	86.2 ±0.3	86.2 ±0.3	93.3 ±0.3	9.4 ±0.5
LISA	92.8 ±0.2	88.7 ±0.6	88.4 ±0.4	79.5 ±0.8	90.0 ±0.3	84.8 ±0.4	92.0 ±0.1	92.0 ±0.1	97.0 ±0.1	5.4 ±0.3
MMD	93.0 ±0.1	83.9 ±1.4	89.5 ±0.4	83.1 ±1.1	90.0 ±0.1	84.5 ±0.1	90.5 ±0.2	90.5 ±0.2	96.2 ±0.1	6.4 ±0.5
ReSample	89.4 ±0.9	77.7 ±1.2	84.0 ±1.4	72.1 ±3.1	85.7 ±1.0	78.4 ±1.4	88.3 ±0.4	88.3 ±0.4	95.2 ±0.3	8.0 ±1.1
ReWeight	91.8 ±0.2	86.9 ±0.7	87.1 ±0.3	77.5 ±0.8	88.7 ±0.2	82.7 ±0.3	90.7 ±0.1	90.7 ±0.1	95.8 ±0.1	7.0 ±0.6
SqrtReWeight	88.7 ±0.3	78.6 ±0.1	82.8 ±0.4	69.6 ±1.1	84.9 ±0.3	77.3 ±0.4	88.1 ±0.2	88.1 ±0.2	94.5 ±0.1	8.2 ±0.7
CBLoss	91.3 ±0.7	86.2 ±0.3	86.5 ±1.1	76.4 ±2.3	88.2 ±0.7	82.0 ±1.0	90.4 ±0.1	90.4 ±0.1	95.7 ±0.0	8.2 ±1.4
Focal	89.3 ±0.2	71.6 ±0.8	83.7 ±0.3	72.4 ±0.5	85.2 ±0.3	77.5 ±0.4	87.1 ±0.3	87.1 ±0.3	94.2 ±0.2	6.9 ±0.1
LDAM	87.3 ±0.5	71.0 ±1.8	81.2 ±0.6	67.7 ±1.5	83.0 ±0.5	74.4 ±0.6	85.7 ±0.2	85.7 ±0.2	93.3 ±0.2	13.7 ±2.2
BSoftmax	88.4 ±1.3	74.1 ±0.9	82.7 ±1.6	70.1 ±3.0	84.4 ±1.5	76.5 ±2.1	87.0 ±1.0	87.0 ±1.0	94.1 ±1.0	9.8 ±1.2
DFR	92.3 ±0.2	91.0 ±0.3	87.5 ±0.3	77.5 ±0.6	89.5 ±0.2	84.1 ±0.3	92.1 ±0.1	92.1 ±0.1	97.4 ±0.1	7.1 ±0.6
CRT	90.5 ±0.0	79.7 ±0.3	85.3 ±0.0	74.5 ±0.0	87.0 ±0.1	80.3 ±0.1	89.3 ±0.1	89.3 ±0.1	95.7 ±0.0	7.9 ±0.1
ReWeightCRT	91.2 ±0.1	78.4 ±0.1	86.4 ±0.2	76.8 ±0.3	87.7 ±0.1	81.2 ±0.2	89.4 ±0.1	89.4 ±0.1	95.8 ±0.1	6.3 ±0.2

CelebA

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	95.1 ±0.2	62.6 ±1.5	87.5 ±0.5	76.4 ±1.2	90.1 ±0.3	83.1 ±0.5	86.9 ±0.2	93.4 ±0.2	98.4 ±0.1	3.3 ±0.5
Mixup	95.4 ±0.1	57.8 ±0.8	88.4 ±0.3	78.5 ±0.7	90.6 ±0.2	83.8 ±0.3	85.8 ±0.2	93.1 ±0.1	98.4 ±0.1	2.5 ±0.2
GroupDRO	91.4 ±0.6	89.0 ±0.7	80.4 ±0.8	61.5 ±1.7	84.9 ±0.8	74.9 ±1.2	92.6 ±0.1	93.3 ±0.2	98.1 ±0.0	8.0 ±0.9
IRM	94.7 ±0.8	63.0 ±2.5	87.0 ±1.9	75.3 ±3.9	89.6 ±1.1	82.2 ±1.8	86.9 ±0.5	93.3 ±0.3	98.5 ±0.0	3.4 ±1.3
CVaRDRO	95.2 ±0.1	64.1 ±2.8	88.4 ±0.6	78.6 ±1.4	90.1 ±0.1	83.0 ±0.2	86.7 ±0.9	92.2 ±0.7	98.2 ±0.1	2.6 ±0.3
JTT	90.4 ±2.3	70.0 ±10.2	80.5 ±4.2	62.5 ±8.7	83.4 ±3.3	72.6 ±5.1	86.4 ±1.6	90.3 ±1.1	93.2 ±2.2	4.1 ±1.4
LfF	81.1 ±5.6	53.0 ±4.3	71.8 ±4.1	45.2 ±8.3	73.2 ±5.6	59.0 ±7.3	78.3 ±3.0	85.3 ±2.9	94.1 ±1.2	27.9 ±5.5
LISA	92.6 ±0.1	86.5 ±1.2	82.2 ±0.2	65.1 ±0.4	86.6 ±0.2	77.6 ±0.3	92.0 ±0.3	94.0 ±0.1	98.5 ±0.0	7.7 ±0.3
MMD	92.5 ±0.7	24.4 ±2.0	91.4 ±1.4	90.1 ±2.2	79.8 ±2.1	63.7 ±3.9	68.5 ±0.7	74.3 ±2.1	96.0 ±0.9	3.6 ±0.2
ReSample	92.0 ±0.8	87.4 ±0.8	81.4 ±1.2	63.6 ±2.6	85.6 ±1.0	76.0 ±1.5	92.0 ±0.2	93.1 ±0.1	98.1 ±0.0	7.4 ±1.1
ReWeight	91.9 ±0.5	89.7 ±0.2	81.2 ±0.8	63.2 ±1.8	85.4 ±0.7	75.7 ±1.0	92.6 ±0.2	93.0 ±0.2	98.0 ±0.1	7.9 ±0.9
SqrtReWeight	93.6 ±0.1	82.4 ±0.5	84.0 ±0.2	69.0 ±0.3	87.9 ±0.2	79.6 ±0.3	91.2 ±0.1	93.8 ±0.1	98.4 ±0.1	5.8 ±0.2
CBLoss	91.2 ±0.7	89.4 ±0.7	80.2 ±1.1	61.0 ±2.3	84.6 ±1.0	74.5 ±1.6	92.6 ±0.2	93.2 ±0.3	98.0 ±0.1	8.4 ±1.0
Focal	94.9 ±0.3	59.1 ±2.0	87.5 ±0.8	76.7 ±1.7	89.7 ±0.4	82.4 ±0.6	85.6 ±0.5	92.5 ±0.4	98.2 ±0.1	3.2 ±0.4
LDAM	94.5 ±0.2	59.6 ±2.4	86.5 ±0.8	74.7 ±1.9	89.0 ±0.2	81.3 ±0.3	85.6 ±0.8	92.3 ±0.7	98.0 ±0.1	28.3 ±2.7
BSoftmax	91.9 ±0.1	83.3 ±0.5	81.1 ±0.2	62.9 ±0.4	85.6 ±0.2	76.1 ±0.3	91.1 ±0.2	93.9 ±0.1	98.6 ±0.0	8.4 ±0.2
DFR	91.9 ±0.1	90.4 ±0.1	81.2 ±0.2	63.2 ±0.3	85.5 ±0.1	75.8 ±0.2	92.3 ±0.0	93.1 ±0.1	97.9 ±0.0	8.9 ±0.1
CRT	92.7 ±0.1	87.2 ±0.3	82.4 ±0.1	65.7 ±0.2	86.5 ±0.1	77.4 ±0.1	91.8 ±0.1	93.4 ±0.0	98.2 ±0.0	6.6 ±0.1
ReWeightCRT	92.5 ±0.2	87.2 ±0.3	82.1 ±0.3	65.1 ±0.6	86.3 ±0.2	77.1 ±0.3	91.8 ±0.0	93.4 ±0.0	98.2 ±0.0	7.1 ±0.3

CivilComments

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	85.4 ±0.2	63.7 ±1.1	75.4 ±0.2	57.8 ±0.6	77.0 ±0.0	63.1 ±0.1	77.7 ±0.2	79.2 ±0.3	90.0 ±0.0	8.1 ±0.2
Mixup	84.9 ±0.3	66.1 ±1.3	74.8 ±0.4	56.4 ±0.9	76.6 ±0.2	62.7 ±0.2	77.9 ±0.3	79.3 ±0.3	89.7 ±0.0	8.4 ±1.0
GroupDRO	81.8 ±0.6	70.6 ±1.2	72.0 ±0.5	49.6 ±1.1	74.2 ±0.5	60.3 ±0.6	78.5 ±0.2	79.9 ±0.2	88.8 ±0.2	12.2 ±0.9
IRM	85.5 ±0.0	63.2 ±0.8	75.5 ±0.1	57.8 ±0.2	77.1 ±0.0	63.3 ±0.1	77.8 ±0.1	79.4 ±0.1	89.9 ±0.1	7.4 ±0.6
CVaRDRO	83.5 ±0.3	68.7 ±1.3	73.5 ±0.3	52.8 ±0.6	75.9 ±0.2	62.4 ±0.2	78.6 ±0.2	80.7 ±0.1	89.8 ±0.1	32.9 ±0.4
JTT	83.3 ±0.1	64.3 ±1.5	72.8 ±0.1	52.4 ±0.3	74.8 ±0.1	60.3 ±0.2	76.8 ±0.2	78.4 ±0.2	88.2 ±0.1	10.2 ±0.3
LfF	65.5 ±5.6	51.0 ±6.1	60.4 ±3.5	31.2 ±5.6	58.5 ±5.0	41.9 ±5.6	64.8 ±4.2	65.6 ±4.5	69.2 ±6.5	26.4 ±2.4
LISA	82.7 ±0.1	73.7 ±0.3	72.6 ±0.1	51.1 ±0.2	75.0 ±0.1	61.1 ±0.1	78.7 ±0.2	80.1 ±0.1	89.1 ±0.1	11.7 ±0.3
MMD	84.6 ±0.2	54.5 ±1.4	73.9 ±0.4	56.7 ±0.7	74.4 ±0.4	58.2 ±0.7	73.6 ±0.6	74.9 ±0.5	86.1 ±0.7	5.0 ±1.5
ReSample	82.2 ±0.0	73.3 ±0.5	72.4 ±0.0	50.2 ±0.1	74.8 ±0.0	61.1 ±0.0	79.2 ±0.0	80.6 ±0.0	89.3 ±0.1	12.2 ±0.2
ReWeight	82.5 ±0.0	72.5 ±0.0	72.6 ±0.1	50.8 ±0.1	75.0 ±0.1	61.4 ±0.1	79.1 ±0.1	80.6 ±0.1	89.5 ±0.0	12.0 ±0.2
SqrtReWeight	83.3 ±0.5	71.7 ±0.4	73.3 ±0.4	52.5 ±1.0	75.7 ±0.4	62.0 ±0.4	78.9 ±0.1	80.4 ±0.1	89.7 ±0.0	10.3 ±0.8
CBLoss	82.9 ±0.1	73.3 ±0.2	72.9 ±0.1	51.5 ±0.2	75.4 ±0.1	61.7 ±0.1	79.2 ±0.1	80.6 ±0.1	89.6 ±0.1	11.1 ±0.3
Focal	85.5 ±0.2	62.0 ±1.0	75.5 ±0.4	58.5 ±0.8	76.8 ±0.3	62.5 ±0.4	76.9 ±0.4	78.4 ±0.4	89.1 ±0.3	6.7 ±0.4
LDAM	81.9 ±2.2	37.4 ±8.1	69.6 ±3.5	49.9 ±5.9	69.7 ±3.4	50.6 ±5.5	67.5 ±4.0	70.0 ±3.3	79.7 ±4.2	21.1 ±0.3
BSoftmax	83.8 ±0.0	71.2 ±0.4	73.8 ±0.0	53.5 ±0.0	76.1 ±0.0	62.5 ±0.0	78.7 ±0.1	80.4 ±0.0	89.8 ±0.0	10.3 ±0.1
DFR	83.3 ±0.0	69.6 ±0.2	73.2 ±0.0	52.3 ±0.1	75.6 ±0.0	61.8 ±0.0	78.1 ±0.0	80.2 ±0.0	89.5 ±0.0	16.6 ±0.3
CRT	83.8 ±0.0	71.1 ±0.1	73.8 ±0.0	53.5 ±0.0	76.1 ±0.0	62.5 ±0.0	78.6 ±0.0	80.4 ±0.0	89.4 ±0.0	11.2 ±0.3
ReWeightCRT	83.8 ±0.1	71.0 ±0.1	73.8 ±0.1	53.5 ±0.2	76.1 ±0.0	62.4 ±0.0	78.5 ±0.0	80.4 ±0.1	89.6 ±0.0	10.7 ±0.1

MultiNLI

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	80.9 ±0.1	66.8 ±0.5	81.1 ±0.1	76.0 ±0.2	80.9 ±0.1	77.8 ±0.1	79.7 ±0.0	80.9 ±0.1	93.6 ±0.1	8.1 ±0.3
Mixup	81.4 ±0.3	68.5 ±0.6	81.6 ±0.3	76.0 ±0.5	81.4 ±0.3	78.0 ±0.2	80.1 ±0.3	81.4 ±0.3	93.6 ±0.1	9.4 ±0.9
GroupDRO	81.1 ±0.3	76.0 ±0.7	81.4 ±0.3	74.7 ±0.5	81.1 ±0.3	77.8 ±0.1	80.8 ±0.3	81.1 ±0.3	93.7 ±0.1	9.8 ±1.0
IRM	77.8 ±0.6	63.6 ±1.3	78.3 ±0.5	71.0 ±1.1	77.9 ±0.6	74.8 ±0.4	76.6 ±0.5	77.8 ±0.6	91.5 ±0.3	11.2 ±1.8
CVaRDRO	75.1 ±0.1	63.0 ±1.5	76.2 ±0.2	65.6 ±0.2	75.2 ±0.1	72.1 ±0.2	74.2 ±0.4	75.1 ±0.1	86.3 ±0.2	41.4 ±0.1
JTT	80.9 ±0.5	69.1 ±0.1	81.3 ±0.4	74.3 ±1.1	81.0 ±0.5	77.6 ±0.5	80.0 ±0.4	80.9 ±0.5	93.7 ±0.2	7.0 ±1.5
LfF	71.7 ±1.1	63.6 ±2.9	71.8 ±1.1	68.7 ±0.7	71.7 ±1.1	68.5 ±1.8	70.8 ±1.4	71.7 ±1.1	87.0 ±0.8	4.4 ±0.6
LISA	80.3 ±0.4	73.3 ±1.0	80.4 ±0.4	75.9 ±0.3	80.3 ±0.4	76.7 ±0.4	79.8 ±0.5	80.3 ±0.4	92.7 ±0.2	4.3 ±0.4
MMD	78.8 ±0.1	69.1 ±1.5	79.3 ±0.2	71.7 ±0.7	78.9 ±0.1	75.5 ±0.1	78.0 ±0.4	78.8 ±0.1	91.7 ±0.1	11.6 ±0.3
ReSample	77.2 ±0.2	72.3 ±0.8	77.6 ±0.0	70.7 ±1.0	77.3 ±0.1	73.8 ±0.1	77.6 ±0.3	77.2 ±0.2	90.9 ±0.0	10.8 ±0.2
ReWeight	81.0 ±0.2	68.8 ±0.4	81.1 ±0.2	76.0 ±0.7	81.0 ±0.2	77.4 ±0.1	79.6 ±0.1	81.0 ±0.2	93.5 ±0.1	8.1 ±0.1
SqrtReWeight	80.7 ±0.3	69.5 ±0.7	81.0 ±0.3	74.6 ±0.5	80.8 ±0.3	77.5 ±0.3	79.9 ±0.4	80.7 ±0.3	93.4 ±0.2	9.2 ±1.0
CBLoss	80.6 ±0.1	72.2 ±0.3	80.8 ±0.1	74.9 ±0.3	80.6 ±0.1	77.5 ±0.1	80.1 ±0.1	80.6 ±0.1	93.4 ±0.1	7.5 ±0.5
Focal	80.7 ±0.2	69.4 ±0.7	81.2 ±0.2	73.7 ±0.6	80.8 ±0.2	77.3 ±0.1	79.6 ±0.2	80.7 ±0.2	93.6 ±0.1	4.4 ±1.0
LDAM	80.7 ±0.3	69.6 ±1.6	81.1 ±0.1	73.9 ±0.9	80.8 ±0.2	77.4 ±0.2	79.7 ±0.3	80.7 ±0.3	93.5 ±0.1	33.4 ±0.3
BSoftmax	80.9 ±0.1	66.9 ±0.4	81.1 ±0.1	75.9 ±0.3	80.9 ±0.1	77.7 ±0.1	79.7 ±0.0	80.9 ±0.1	93.6 ±0.1	8.1 ±0.2
DFR	81.7 ±0.0	68.5 ±0.2	82.1 ±0.0	75.6 ±0.2	81.7 ±0.0	77.9 ±0.0	81.2 ±0.0	81.7 ±0.0	93.2 ±0.0	8.8 ±0.3
CRT	81.9 ±0.0	70.7 ±0.1	82.2 ±0.0	75.9 ±0.1	82.0 ±0.0	78.3 ±0.0	81.1 ±0.0	81.9 ±0.0	93.9 ±0.0	11.5 ±0.0
ReWeightCRT	81.3 ±0.0	69.0 ±0.2	81.4 ±0.0	77.0 ±0.1	81.3 ±0.0	77.6 ±0.0	80.5 ±0.0	81.3 ±0.0	93.7 ±0.0	6.9 ±0.1

MetaShift

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	91.3 ±0.3	82.6 ±0.4	91.2 ±0.3	90.6 ±0.4	91.2 ±0.3	90.7 ±0.3	89.3 ±0.3	91.2 ±0.3	97.3 ±0.2	6.3 ±0.9
Mixup	91.6 ±0.3	81.0 ±0.8	91.7 ±0.3	90.6 ±0.2	91.6 ±0.3	91.0 ±0.3	89.4 ±0.3	91.6 ±0.3	97.3 ±0.1	2.3 ±0.1
GroupDRO	91.0 ±0.1	85.6 ±0.4	90.9 ±0.1	90.0 ±0.6	90.9 ±0.1	90.4 ±0.0	89.8 ±0.1	91.0 ±0.1	97.5 ±0.0	3.2 ±0.5
IRM	91.8 ±0.4	83.0 ±0.1	91.8 ±0.4	90.5 ±1.0	91.7 ±0.4	91.3 ±0.4	89.7 ±0.5	91.7 ±0.4	97.6 ±0.2	5.3 ±0.2
CVaRDRO	92.1 ±0.2	84.6 ±0.0	92.1 ±0.2	90.8 ±0.6	92.1 ±0.2	91.6 ±0.2	90.4 ±0.2	92.1 ±0.2	97.7 ±0.0	4.9 ±0.3
JTT	91.2 ±0.5	83.6 ±0.4	91.3 ±0.6	89.3 ±1.1	91.1 ±0.5	90.6 ±0.4	89.6 ±0.8	91.1 ±0.5	97.4 ±0.0	5.9 ±0.7
LfF	80.2 ±0.3	73.1 ±1.6	80.5 ±0.3	77.2 ±1.3	80.1 ±0.3	78.8 ±0.3	80.3 ±0.6	80.1 ±0.1	90.6 ±0.6	8.3 ±1.5
LISA	89.5 ±0.4	84.1 ±0.4	89.6 ±0.4	88.4 ±0.3	89.5 ±0.5	88.8 ±0.6	88.5 ±0.3	89.5 ±0.5	96.0 ±0.1	25.4 ±0.2
MMD	89.4 ±0.1	85.9 ±0.7	89.5 ±0.2	88.3 ±0.2	89.3 ±0.1	88.4 ±0.1	89.4 ±0.0	89.2 ±0.1	95.4 ±0.3	3.2 ±0.3
ReSample	91.2 ±0.1	85.6 ±0.4	91.1 ±0.1	90.8 ±0.1	91.1 ±0.1	90.5 ±0.1	90.0 ±0.2	91.1 ±0.1	97.4 ±0.1	5.2 ±0.2
ReWeight	91.7 ±0.4	85.6 ±0.4	91.8 ±0.4	90.2 ±0.6	91.7 ±0.3	91.1 ±0.3	90.6 ±0.5	91.6 ±0.3	97.5 ±0.1	4.2 ±0.2
SqrtReWeight	91.5 ±0.2	84.6 ±0.7	91.5 ±0.2	89.7 ±0.2	91.5 ±0.2	91.1 ±0.2	89.7 ±0.3	91.6 ±0.2	97.7 ±0.0	3.6 ±0.6
CBLoss	91.7 ±0.4	85.5 ±0.4	91.8 ±0.4	90.2 ±0.7	91.6 ±0.3	91.1 ±0.3	90.6 ±0.4	91.6 ±0.3	97.5 ±0.1	4.1 ±0.2
Focal	91.7 ±0.2	81.5 ±0.0	91.7 ±0.2	91.1 ±0.6	91.7 ±0.2	91.2 ±0.2	89.5 ±0.2	91.7 ±0.2	97.7 ±0.0	5.2 ±1.6
LDAM	91.5 ±0.1	83.6 ±0.4	91.5 ±0.1	90.7 ±0.3	91.5 ±0.1	90.9 ±0.1	89.8 ±0.1	91.5 ±0.1	97.5 ±0.1	10.8 ±0.6
BSoftmax	91.6 ±0.2	83.1 ±0.7	91.6 ±0.2	89.8 ±0.3	91.6 ±0.2	91.2 ±0.2	89.4 ±0.3	91.7 ±0.1	97.7 ±0.0	4.0 ±0.6
DFR	88.4 ±0.3	85.4 ±0.4	88.4 ±0.3	86.8 ±0.3	88.4 ±0.3	87.8 ±0.4	87.7 ±0.3	88.5 ±0.3	95.6 ±0.1	5.7 ±0.2
CRT	91.3 ±0.2	84.1 ±0.4	91.3 ±0.2	90.2 ±0.2	91.3 ±0.2	90.8 ±0.2	89.6 ±0.2	91.3 ±0.2	97.3 ±0.0	7.4 ±0.1
ReWeightCRT	91.2 ±0.1	85.6 ±0.4	91.1 ±0.1	90.1 ±0.1	91.2 ±0.1	90.7 ±0.0	89.8 ±0.1	91.2 ±0.0	96.8 ±0.1	7.8 ±0.1

ImageNetBG

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	88.4 ±0.1	81.0 ±0.9	88.5 ±0.1	80.4 ±0.1	88.4 ±0.1	81.3 ±0.1	88.4 ±0.1	88.4 ±0.1	99.0 ±0.0	5.8 ±0.2
Mixup	88.5 ±0.3	82.4 ±0.3	88.7 ±0.2	80.4 ±1.3	88.5 ±0.3	81.9 ±0.8	88.5 ±0.3	88.5 ±0.3	98.8 ±0.1	3.3 ±1.0
GroupDRO	87.3 ±0.2	78.2 ±0.3	87.5 ±0.2	77.9 ±1.1	87.3 ±0.2	80.0 ±0.9	87.3 ±0.2	87.3 ±0.2	98.9 ±0.0	4.3 ±0.5
IRM	88.7 ±0.1	81.3 ±0.3	88.8 ±0.1	81.6 ±0.2	88.7 ±0.1	81.7 ±0.1	88.7 ±0.1	88.7 ±0.1	99.1 ±0.0	5.2 ±0.1
CVaRDRO	88.2 ±0.1	80.7 ±1.1	88.4 ±0.1	78.6 ±1.9	88.3 ±0.1	80.7 ±0.5	88.2 ±0.1	88.2 ±0.1	99.0 ±0.0	4.9 ±0.4
JTT	87.2 ±0.1	80.5 ±0.3	87.5 ±0.2	78.0 ±0.7	87.2 ±0.1	80.2 ±0.6	87.2 ±0.1	87.2 ±0.1	98.9 ±0.0	2.4 ±0.5
LfF	85.3 ±0.3	76.7 ±0.5	85.6 ±0.3	74.0 ±2.2	85.3 ±0.3	75.8 ±1.3	85.3 ±0.3	85.3 ±0.3	98.5 ±0.0	2.6 ±0.4
LISA	86.2 ±0.3	76.1 ±0.8	86.3 ±0.3	75.5 ±1.0	86.2 ±0.3	77.1 ±0.5	86.2 ±0.3	86.2 ±0.3	98.3 ±0.1	4.2 ±0.2
MMD	88.2 ±0.2	80.8 ±0.5	88.4 ±0.2	80.0 ±1.1	88.2 ±0.2	80.7 ±0.3	88.2 ±0.2	88.2 ±0.2	99.0 ±0.0	5.8 ±0.2
ReSample	88.5 ±0.2	81.0 ±0.4	88.7 ±0.2	79.9 ±1.0	88.5 ±0.2	81.5 ±0.4	88.5 ±0.2	88.5 ±0.2	99.0 ±0.0	6.0 ±0.2
ReWeight	88.4 ±0.1	81.0 ±0.9	88.5 ±0.1	80.4 ±0.1	88.4 ±0.1	81.3 ±0.1	88.4 ±0.1	88.4 ±0.1	99.0 ±0.0	5.8 ±0.2
SqrtReWeight	88.3 ±0.1	80.1 ±0.2	88.4 ±0.1	80.5 ±0.5	88.3 ±0.1	80.9 ±0.4	88.3 ±0.1	88.3 ±0.1	99.0 ±0.0	5.3 ±0.3
CBLoss	88.4 ±0.1	81.0 ±0.9	88.5 ±0.1	80.4 ±0.1	88.4 ±0.1	81.3 ±0.1	88.4 ±0.1	88.4 ±0.1	99.0 ±0.0	5.8 ±0.2
Focal	87.2 ±0.1	78.4 ±0.1	87.3 ±0.2	78.7 ±0.7	87.2 ±0.1	78.9 ±0.5	87.2 ±0.1	87.2 ±0.1	98.8 ±0.0	4.4 ±1.1
LDAM	88.0 ±0.1	80.1 ±0.3	88.3 ±0.0	80.1 ±0.6	88.1 ±0.1	81.4 ±0.3	88.0 ±0.1	88.0 ±0.1	98.7 ±0.1	48.3 ±1.9
BSoftmax	88.3 ±0.1	80.7 ±0.7	88.4 ±0.1	79.4 ±0.9	88.3 ±0.1	80.8 ±0.4	88.3 ±0.1	88.3 ±0.1	99.0 ±0.0	6.0 ±0.2
DFR	87.2 ±0.2	78.5 ±0.6	87.2 ±0.3	78.2 ±1.2	87.2 ±0.2	78.8 ±0.9	87.2 ±0.2	87.2 ±0.2	98.8 ±0.0	9.9 ±1.3
CRT	88.4 ±0.1	80.2 ±0.3	88.4 ±0.1	80.4 ±0.8	88.3 ±0.1	80.7 ±0.3	88.4 ±0.1	88.4 ±0.1	99.0 ±0.0	4.5 ±0.5
ReWeightCRT	88.6 ±0.0	79.4 ±0.2	88.7 ±0.0	81.6 ±0.7	88.6 ±0.0	81.5 ±0.2	88.6 ±0.0	88.6 ±0.0	99.1 ±0.0	4.5 ±0.8

NICO++

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	84.5 ±0.5	37.6 ±2.0	85.5 ±0.3	54.5 ±2.8	84.6 ±0.4	65.8 ±1.1	84.0 ±0.5	84.3 ±0.5	99.3 ±0.0	10.4 ±0.1
Mixup	84.0 ±0.6	42.7 ±1.4	85.2 ±0.5	53.0 ±1.6	84.2 ±0.6	63.4 ±1.1	83.7 ±0.6	83.9 ±0.6	99.3 ±0.0	2.5 ±1.0
GroupDRO	83.2 ±0.4	37.8 ±1.8	84.5 ±0.4	55.5 ±1.0	83.3 ±0.4	63.6 ±0.6	82.7 ±0.4	83.0 ±0.4	99.3 ±0.0	8.7 ±0.6
IRM	84.4 ±0.7	40.0 ±0.0	85.1 ±0.5	63.0 ±2.0	84.4 ±0.6	65.9 ±1.2	83.9 ±0.7	84.3 ±0.7	99.4 ±0.0	7.0 ±1.4
CVaRDRO	83.6 ±0.6	36.7 ±2.7	85.0 ±0.4	55.7 ±2.3	83.8 ±0.6	64.3 ±1.5	83.2 ±0.6	83.5 ±0.6	99.4 ±0.0	7.9 ±1.1
JTT	85.1 ±0.3	40.0 ±0.0	86.0 ±0.3	54.8 ±2.7	85.2 ±0.3	65.4 ±1.8	84.7 ±0.3	85.0 ±0.3	99.4 ±0.0	10.2 ±0.2
LfF	78.3 ±0.4	30.4 ±1.3	80.7 ±0.2	45.6 ±1.3	78.6 ±0.4	52.5 ±0.6	78.0 ±0.3	78.3 ±0.4	99.2 ±0.0	1.4 ±0.3
LISA	84.7 ±0.3	42.7 ±2.2	85.7 ±0.2	54.7 ±1.4	84.8 ±0.3	65.4 ±1.2	84.2 ±0.3	84.6 ±0.3	99.2 ±0.0	11.9 ±1.6
MMD	84.9 ±0.1	40.7 ±0.5	85.8 ±0.1	57.0 ±1.2	85.0 ±0.1	66.3 ±0.7	84.5 ±0.1	84.8 ±0.1	99.4 ±0.0	9.2 ±0.4
ReSample	84.8 ±0.3	40.0 ±0.0	85.8 ±0.3	58.6 ±2.6	84.9 ±0.4	65.4 ±1.7	84.4 ±0.4	84.7 ±0.4	99.4 ±0.0	8.8 ±0.2
ReWeight	85.7 ±0.2	41.9 ±1.6	86.6 ±0.1	57.3 ±3.8	85.8 ±0.1	65.0 ±1.7	85.3 ±0.2	85.6 ±0.2	99.4 ±0.0	9.8 ±0.3
SqrtReWeight	84.7 ±0.7	40.0 ±0.0	85.7 ±0.4	57.5 ±1.3	84.8 ±0.6	65.7 ±1.5	84.2 ±0.6	84.6 ±0.7	99.4 ±0.0	8.1 ±1.1
CBLoss	84.5 ±0.4	37.8 ±1.8	85.2 ±0.5	61.1 ±0.8	84.5 ±0.5	66.1 ±1.4	84.0 ±0.4	84.3 ±0.4	99.4 ±0.0	8.3 ±1.2
Focal	83.8 ±1.4	36.7 ±2.7	85.0 ±1.1	54.2 ±3.7	83.9 ±1.4	63.8 ±3.0	83.3 ±1.4	83.6 ±1.4	99.4 ±0.1	4.8 ±0.7
LDAM	82.8 ±0.4	42.0 ±0.9	84.4 ±0.3	51.1 ±2.7	83.0 ±0.4	62.0 ±1.6	82.4 ±0.4	82.7 ±0.4	98.7 ±0.1	68.7 ±2.2
BSoftmax	84.0 ±0.5	40.4 ±0.3	84.8 ±0.3	61.4 ±1.1	84.1 ±0.4	65.2 ±1.1	83.7 ±0.5	84.0 ±0.5	99.4 ±0.0	7.0 ±1.2
DFR	75.6 ±0.5	23.7 ±0.7	77.4 ±0.4	37.7 ±3.2	75.8 ±0.4	46.0 ±2.6	75.3 ±0.5	75.5 ±0.5	98.6 ±0.0	19.4 ±0.5
CRT	85.2 ±0.3	43.3 ±2.7	85.7 ±0.2	64.6 ±0.6	85.2 ±0.3	69.2 ±0.3	84.7 ±0.3	85.0 ±0.3	99.4 ±0.0	7.9 ±0.6
ReWeightCRT	85.0 ±0.1	23.3 ±1.4	85.5 ±0.1	61.6 ±0.3	85.0 ±0.1	67.0 ±0.1	84.3 ±0.1	84.8 ±0.1	99.3 ±0.0	3.6 ±0.3

MIMIC-CXR

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	78.2 ±0.1	68.9 ±0.3	77.3 ±0.1	71.1 ±0.1	77.5 ±0.1	73.6 ±0.1	77.2 ±0.0	77.8 ±0.1	85.2 ±0.1	3.4 ±0.2
Mixup	78.3 ±0.0	68.1 ±0.9	77.4 ±0.0	71.6 ±0.2	77.5 ±0.0	73.4 ±0.1	77.2 ±0.1	77.8 ±0.1	85.1 ±0.1	3.6 ±0.2
GroupDRO	76.9 ±0.3	74.4 ±0.2	76.1 ±0.2	68.7 ±0.5	76.3 ±0.2	72.7 ±0.1	76.7 ±0.1	76.9 ±0.2	83.7 ±0.1	4.7 ±0.1
IRM	78.2 ±0.0	67.7 ±0.2	77.3 ±0.0	71.4 ±0.1	77.5 ±0.0	73.5 ±0.1	77.2 ±0.1	77.8 ±0.1	85.2 ±0.1	3.4 ±0.2
CVaRDRO	78.3 ±0.1	68.6 ±0.4	77.4 ±0.1	71.1 ±0.3	77.7 ±0.1	73.9 ±0.0	77.4 ±0.0	78.1 ±0.0	85.1 ±0.0	7.8 ±0.3
JTT	78.1 ±0.0	67.3 ±0.7	77.1 ±0.0	71.4 ±0.2	77.3 ±0.0	73.2 ±0.1	77.0 ±0.0	77.5 ±0.1	84.9 ±0.0	3.4 ±0.1
LfF	73.3 ±0.9	62.6 ±2.6	72.3 ±1.0	65.2 ±1.0	72.4 ±1.0	67.7 ±1.4	72.4 ±1.1	72.8 ±1.1	79.3 ±1.3	12.3 ±0.7
LISA	77.9 ±0.1	70.4 ±0.2	77.0 ±0.1	70.6 ±0.3	77.2 ±0.1	73.3 ±0.1	77.2 ±0.1	77.6 ±0.1	84.9 ±0.1	4.0 ±0.6
MMD	76.8 ±0.2	68.0 ±0.6	75.9 ±0.2	70.2 ±0.4	76.0 ±0.2	71.5 ±0.3	76.0 ±0.3	76.2 ±0.2	83.4 ±0.2	8.8 ±2.0
ReSample	78.1 ±0.1	71.9 ±0.2	77.3 ±0.1	70.7 ±0.3	77.5 ±0.1	73.8 ±0.2	77.6 ±0.1	78.0 ±0.1	85.0 ±0.1	5.5 ±0.8
ReWeight	78.2 ±0.1	71.6 ±0.3	77.4 ±0.1	70.9 ±0.3	77.6 ±0.1	73.8 ±0.1	77.6 ±0.1	78.0 ±0.1	85.1 ±0.1	4.2 ±0.2
SqrtReWeight	78.2 ±0.2	70.3 ±0.2	77.3 ±0.2	71.0 ±0.3	77.5 ±0.2	73.6 ±0.2	77.3 ±0.3	77.9 ±0.2	85.2 ±0.2	4.1 ±0.2
CBLoss	78.4 ±0.1	70.7 ±0.1	77.5 ±0.1	71.6 ±0.2	77.7 ±0.1	73.8 ±0.1	77.6 ±0.1	78.0 ±0.1	85.2 ±0.0	4.1 ±0.4
Focal	78.3 ±0.1	68.7 ±0.4	77.4 ±0.1	70.8 ±0.2	77.6 ±0.1	73.9 ±0.0	77.4 ±0.1	78.1 ±0.0	85.4 ±0.0	10.1 ±0.6
LDAM	77.7 ±0.6	68.6 ±1.1	76.8 ±0.6	70.4 ±0.9	77.0 ±0.6	73.1 ±0.7	76.9 ±0.6	77.4 ±0.6	84.6 ±0.6	22.0 ±0.2
BSoftmax	77.8 ±0.2	68.4 ±0.2	76.9 ±0.2	70.2 ±0.3	77.1 ±0.2	73.3 ±0.2	77.0 ±0.2	77.6 ±0.2	84.9 ±0.2	5.0 ±0.2
DFR	78.0 ±0.0	68.9 ±0.0	77.1 ±0.0	70.9 ±0.0	77.3 ±0.0	73.3 ±0.0	77.0 ±0.0	77.6 ±0.0	84.9 ±0.0	7.0 ±0.1
CRT	78.5 ±0.0	71.0 ±0.0	77.6 ±0.0	71.5 ±0.1	77.9 ±0.0	74.0 ±0.0	77.7 ±0.0	78.2 ±0.0	85.4 ±0.0	4.1 ±0.1
ReWeightCRT	78.5 ±0.0	70.8 ±0.0	77.6 ±0.0	71.5 ±0.1	77.8 ±0.0	73.9 ±0.0	77.7 ±0.0	78.2 ±0.0	85.4 ±0.0	4.3 ±0.0

MIMICNotes

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	91.1 ±0.1	18.7 ±2.7	77.6 ±0.9	63.1 ±2.0	63.2 ±1.6	31.2 ±3.1	59.7 ±1.3	59.9 ±1.3	85.3 ±0.1	2.1 ±0.8
Mixup	91.1 ±0.0	22.7 ±3.2	76.8 ±0.7	61.2 ±1.6	65.1 ±1.6	35.0 ±3.2	61.5 ±1.6	61.7 ±1.7	85.4 ±0.0	2.0 ±0.8
GroupDRO	76.1 ±0.7	72.6 ±0.5	61.3 ±0.1	25.7 ±0.4	61.8 ±0.4	38.6 ±0.3	76.2 ±0.3	76.5 ±0.2	85.0 ±0.1	22.2 ±0.6
IRM	91.0 ±0.0	22.5 ±2.5	76.3 ±0.5	60.1 ±1.2	65.2 ±1.2	35.3 ±2.4	61.5 ±1.2	61.7 ±1.2	85.3 ±0.0	1.9 ±0.2
CVaRDRO	90.9 ±0.1	23.0 ±4.6	76.5 ±1.2	60.6 ±2.9	64.6 ±2.4	34.0 ±4.8	61.4 ±2.3	61.6 ±2.4	85.1 ±0.1	4.2 ±1.6
JTT	71.3 ±3.7	65.9 ±2.8	60.3 ±0.7	23.4 ±1.9	58.6 ±2.3	36.0 ±1.8	75.5 ±0.4	75.6 ±0.4	84.9 ±0.1	27.5 ±3.9
LfF	84.0 ±1.2	62.7 ±2.1	64.6 ±0.7	33.6 ±1.6	67.1 ±0.8	43.6 ±0.8	74.7 ±0.4	74.7 ±0.5	85.1 ±0.0	12.5 ±1.2
LISA	85.2 ±1.4	58.0 ±3.1	65.5 ±0.9	35.7 ±2.1	68.0 ±0.9	44.5 ±0.9	74.0 ±0.6	74.2 ±0.7	85.3 ±0.0	15.5 ±1.5
MMD	91.2 ±0.1	23.0 ±0.5	76.8 ±0.3	61.0 ±0.5	65.9 ±0.5	36.5 ±0.9	61.9 ±0.4	62.1 ±0.4	85.3 ±0.0	1.4 ±0.1
ReSample	80.4 ±1.8	68.0 ±3.0	63.0 ±0.8	29.7 ±1.8	64.9 ±1.2	41.6 ±1.2	75.8 ±0.4	76.1 ±0.4	85.3 ±0.0	18.8 ±2.2
ReWeight	84.8 ±0.8	60.5 ±2.5	65.2 ±0.6	34.8 ±1.3	67.8 ±0.5	44.4 ±0.5	74.5 ±0.5	74.7 ±0.5	85.2 ±0.0	14.1 ±0.9
SqrtReWeight	90.1 ±0.3	37.2 ±4.5	71.8 ±1.1	49.9 ±2.5	69.1 ±0.9	43.7 ±1.9	67.6 ±1.7	67.8 ±1.7	85.2 ±0.1	4.2 ±1.0
CBLoss	83.2 ±1.2	63.3 ±2.2	64.1 ±0.6	32.5 ±1.5	66.6 ±0.8	43.0 ±0.8	74.8 ±0.4	74.9 ±0.5	85.2 ±0.1	14.7 ±1.3
Focal	91.0 ±0.0	19.1 ±2.3	77.1 ±0.6	62.1 ±1.4	63.6 ±1.3	31.9 ±2.6	59.9 ±1.1	60.2 ±1.1	85.3 ±0.1	8.1 ±0.7
LDAM	90.6 ±0.1	5.3 ±2.4	84.4 ±0.8	78.1 ±1.7	52.5 ±2.1	10.0 ±4.1	52.7 ±1.2	52.7 ±1.2	84.9 ±0.1	28.9 ±1.0
BSoftmax	76.9 ±0.9	73.1 ±1.0	61.7 ±0.2	26.5 ±0.6	62.5 ±0.5	39.3 ±0.4	76.6 ±0.2	76.7 ±0.2	85.4 ±0.0	23.5 ±1.1
DFR	43.1 ±19.8	6.7 ±5.5	51.9 ±2.8	7.3 ±3.0	28.3 ±9.1	7.2 ±5.8	53.4 ±2.8	53.4 ±2.8	84.5 ±0.0	40.1 ±0.3
CRT	82.1 ±3.5	56.2 ±13.8	65.9 ±3.5	36.8 ±8.2	63.4 ±0.5	37.5 ±1.4	70.9 ±4.0	71.0 ±4.0	84.3 ±0.0	28.3 ±4.3
ReWeightCRT	83.5 ±2.6	58.7 ±6.8	64.6 ±1.5	33.9 ±3.5	66.1 ±1.5	42.0 ±1.2	72.9 ±1.5	73.0 ±1.5	84.3 ±0.0	28.9 ±2.2

CXRMultisite

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	93.1 ±0.1	0.3 ±0.1
Mixup	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	92.9 ±0.1	0.3 ±0.0
GroupDRO	84.0 ±10.9	19.3 ±13.7	55.7 ±2.0	12.7 ±4.1	51.2 ±5.0	12.4 ±2.9	55.9 ±2.5	59.7 ±2.5	79.4 ±2.9	29.3 ±6.3
IRM	77.5 ±17.0	8.8 ±7.2	49.6 ±0.3	0.7 ±0.5	42.4 ±5.9	1.3 ±1.0	51.1 ±0.9	51.8 ±1.5	64.2 ±7.3	47.3 ±1.1
CVaRDRO	98.3 ±0.0	0.0 ±0.0	61.2 ±4.9	24.0 ±9.8	50.7 ±0.7	2.2 ±1.5	50.2 ±0.2	50.6 ±0.4	93.0 ±0.0	0.9 ±0.3
JTT	94.1 ±0.9	0.0 ±0.0	59.0 ±0.7	18.5 ±1.4	62.9 ±0.8	28.9 ±1.2	55.2 ±0.9	82.2 ±2.4	93.2 ±0.1	6.4 ±0.5
LfF	9.9 ±6.7	5.4 ±4.4	17.4 ±13.5	0.6 ±0.5	8.5 ±5.6	1.2 ±1.0	50.5 ±0.4	51.7 ±1.4	60.6 ±1.6	82.6 ±12.8
LISA	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	90.3 ±0.0	8.9 ±1.6
MMD	87.4 ±8.9	12.8 ±10.4	49.6 ±0.4	0.8 ±0.6	47.0 ±2.1	1.4 ±1.2	50.6 ±0.5	52.0 ±1.7	56.5 ±1.9	15.4 ±12.5
ReSample	96.4 ±0.3	1.1 ±0.5	57.9 ±0.3	17.1 ±0.5	59.8 ±0.1	21.4 ±0.3	54.0 ±0.1	63.4 ±1.2	89.7 ±0.1	4.1 ±0.2
ReWeight	88.0 ±5.3	19.4 ±7.9	52.9 ±0.7	6.9 ±1.4	52.0 ±2.2	10.8 ±2.1	56.7 ±1.7	64.1 ±4.4	75.7 ±2.4	37.2 ±4.1
SqrtReWeight	98.0 ±0.1	0.0 ±0.0	65.5 ±0.3	32.4 ±0.6	60.7 ±1.8	22.5 ±3.7	53.4 ±0.7	58.9 ±2.1	92.9 ±0.1	4.1 ±0.9
CBLoss	98.0 ±0.0	0.0 ±0.0	64.7 ±0.3	30.9 ±0.6	59.2 ±1.0	19.4 ±1.9	52.5 ±0.3	56.9 ±1.0	92.5 ±0.0	6.0 ±0.8
Focal	98.3 ±0.0	0.0 ±0.0	55.4 ±5.1	12.5 ±10.2	49.7 ±0.1	0.3 ±0.2	50.0 ±0.0	50.1 ±0.1	93.2 ±0.0	11.5 ±0.6
LDAM	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	92.9 ±0.0	33.3 ±0.0
BSoftmax	89.1 ±0.2	0.5 ±0.1	56.2 ±0.1	12.5 ±0.2	58.1 ±0.2	22.0 ±0.3	50.4 ±0.0	90.0 ±0.1	92.9 ±0.1	19.9 ±1.3
DFR	79.3 ±8.8	22.2 ±9.9	54.9 ±2.7	10.9 ±5.6	48.2 ±3.5	9.0 ±1.4	55.5 ±1.8	63.9 ±4.7	78.9 ±5.3	41.5 ±1.9
CRT	87.0 ±2.7	17.2 ±6.5	54.2 ±1.4	9.1 ±2.8	54.2 ±2.9	15.5 ±4.2	58.2 ±1.5	73.4 ±3.0	81.5 ±4.1	35.1 ±3.4
ReWeightCRT	82.5 ±6.3	27.8 ±11.4	56.0 ±3.5	13.0 ±7.1	51.8 ±4.1	13.6 ±4.5	58.7 ±2.1	66.8 ±2.8	81.1 ±4.8	29.9 ±8.6

CheXpert

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	86.9 ±0.5	50.2 ±3.8	66.3 ±0.3	37.6 ±0.8	68.6 ±0.2	44.6 ±0.5	72.5 ±1.1	72.8 ±1.1	84.8 ±0.3	8.6 ±0.4
Mixup	81.9 ±6.2	37.4 ±3.5	63.5 ±5.0	33.9 ±9.3	62.5 ±5.9	35.7 ±7.6	63.8 ±4.7	64.1 ±4.6	76.1 ±8.5	16.1 ±9.0
GroupDRO	78.9 ±0.3	74.5 ±0.2	62.8 ±0.1	28.3 ±0.3	64.4 ±0.2	41.7 ±0.3	78.4 ±0.1	79.0 ±0.1	86.0 ±0.1	21.1 ±1.0
IRM	89.8 ±0.3	34.4 ±1.7	70.1 ±0.7	46.5 ±1.6	68.6 ±0.1	42.7 ±0.3	67.5 ±0.6	67.5 ±0.6	85.8 ±0.3	4.4 ±1.2
CVaRDRO	66.2 ±2.7	57.9 ±0.4	56.4 ±0.5	17.7 ±1.0	52.9 ±1.6	27.9 ±1.1	66.1 ±0.6	67.0 ±0.6	73.0 ±0.6	40.4 ±0.0
JTT	73.0 ±1.9	61.3 ±4.9	58.6 ±1.1	21.6 ±1.7	57.9 ±1.7	32.8 ±2.4	69.8 ±2.6	71.0 ±2.5	77.6 ±2.3	26.3 ±2.0
LfF	22.3 ±10.2	13.7 ±9.8	37.3 ±5.8	9.0 ±0.7	19.5 ±8.3	8.8 ±3.9	46.2 ±2.9	46.2 ±3.1	30.5 ±10.1	65.7 ±10.2
LISA	79.2 ±0.8	75.6 ±0.6	63.1 ±0.4	28.8 ±0.8	64.8 ±0.6	42.3 ±0.7	78.8 ±0.3	79.4 ±0.1	86.5 ±0.1	21.5 ±1.2
MMD	86.9 ±0.5	50.2 ±3.8	66.3 ±0.3	37.6 ±0.8	68.6 ±0.2	44.6 ±0.5	72.5 ±1.1	72.8 ±1.1	84.8 ±0.3	8.6 ±0.4
ReSample	79.0 ±0.8	75.3 ±0.5	62.8 ±0.3	28.4 ±0.7	64.5 ±0.6	41.7 ±0.7	78.4 ±0.0	78.7 ±0.1	85.7 ±0.2	20.1 ±1.4
ReWeight	78.7 ±0.4	75.7 ±0.1	62.7 ±0.1	28.2 ±0.3	64.3 ±0.3	41.6 ±0.3	78.5 ±0.1	78.9 ±0.0	86.3 ±0.0	20.9 ±0.5
SqrtReWeight	82.1 ±1.5	70.0 ±2.3	64.3 ±0.7	31.8 ±1.7	66.7 ±1.1	44.1 ±1.2	77.7 ±0.5	78.3 ±0.5	86.5 ±0.1	18.8 ±2.2
CBLoss	79.1 ±0.1	74.7 ±0.3	62.7 ±0.0	28.3 ±0.1	64.3 ±0.1	41.4 ±0.1	77.9 ±0.0	78.4 ±0.1	85.7 ±0.1	22.1 ±0.5
Focal	89.3 ±0.3	42.1 ±4.0	69.6 ±0.4	44.7 ±1.1	69.8 ±0.4	45.5 ±1.0	70.4 ±1.1	70.4 ±1.3	86.5 ±0.1	16.1 ±1.7
LDAM	90.1 ±0.0	36.4 ±0.3	70.6 ±0.1	47.5 ±0.1	68.9 ±0.2	43.3 ±0.3	67.3 ±0.3	67.6 ±0.2	86.0 ±0.1	32.3 ±0.3
BSoftmax	79.1 ±0.4	75.4 ±0.5	63.0 ±0.2	28.6 ±0.4	64.7 ±0.3	42.1 ±0.4	78.4 ±0.2	79.2 ±0.1	86.4 ±0.1	23.9 ±0.2
DFR	78.2 ±0.4	71.7 ±0.2	62.4 ±0.2	27.6 ±0.4	63.8 ±0.3	40.9 ±0.3	77.5 ±0.1	78.6 ±0.0	85.5 ±0.0	39.5 ±0.1
CRT	79.1 ±0.2	74.6 ±0.3	62.8 ±0.1	28.4 ±0.2	64.4 ±0.2	41.6 ±0.3	78.0 ±0.2	78.6 ±0.2	85.8 ±0.2	21.2 ±0.3
ReWeightCRT	80.4 ±0.0	76.0 ±0.1	63.5 ±0.0	29.8 ±0.1	65.6 ±0.1	43.0 ±0.1	78.8 ±0.1	79.1 ±0.1	86.3 ±0.0	20.2 ±0.1

Living17

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	28.2 ±1.5	8.7 ±1.0	29.5 ±1.1	9.0 ±1.2	27.8 ±1.3	10.0 ±1.5	28.2 ±1.5	28.2 ±1.5	77.9 ±1.2	53.5 ±2.5
Mixup	29.8 ±1.8	9.3 ±1.4	32.5 ±2.3	9.5 ±1.3	29.8 ±1.8	10.1 ±1.5	29.8 ±1.8	29.8 ±1.8	78.3 ±1.2	34.6 ±1.7
GroupDRO	27.2 ±1.5	9.7 ±0.7	29.8 ±0.7	7.8 ±0.9	27.3 ±1.1	9.1 ±0.8	27.2 ±1.5	27.2 ±1.5	77.8 ±0.6	55.8 ±0.5
IRM	28.2 ±1.5	8.7 ±1.0	29.5 ±1.1	9.0 ±1.2	27.8 ±1.3	10.0 ±1.5	28.2 ±1.5	28.2 ±1.5	77.9 ±1.2	53.5 ±2.5
CVaRDRO	28.3 ±0.7	8.3 ±0.7	30.0 ±0.7	7.9 ±0.2	27.9 ±0.8	8.0 ±0.2	28.3 ±0.7	28.3 ±0.7	81.0 ±0.1	33.2 ±4.1
JTT	28.8 ±1.1	8.7 ±1.0	29.8 ±1.8	8.3 ±1.5	28.3 ±1.4	9.1 ±1.6	28.8 ±1.1	28.8 ±1.1	80.2 ±1.1	38.0 ±5.8
LfF	26.2 ±1.1	8.7 ±0.3	28.3 ±0.9	8.8 ±1.1	26.0 ±1.1	9.3 ±0.6	26.2 ±1.1	26.2 ±1.1	76.6 ±0.7	56.4 ±3.3
LISA	29.8 ±0.9	11.3 ±0.3	32.0 ±0.4	9.4 ±0.4	29.9 ±0.7	10.4 ±0.3	29.8 ±0.9	29.8 ±0.9	78.2 ±0.6	30.3 ±0.6
MMD	26.6 ±1.8	8.3 ±0.3	28.9 ±1.1	9.5 ±1.1	26.5 ±1.5	9.5 ±0.9	26.6 ±1.8	26.6 ±1.8	78.5 ±1.0	48.4 ±6.7
ReSample	30.7 ±2.1	10.3 ±2.3	33.1 ±1.2	10.5 ±1.3	30.7 ±2.0	11.2 ±1.4	30.7 ±2.1	30.7 ±2.1	80.9 ±0.4	47.5 ±3.1
ReWeight	28.2 ±1.5	8.7 ±1.0	29.5 ±1.1	9.0 ±1.2	27.8 ±1.3	10.0 ±1.5	28.2 ±1.5	28.2 ±1.5	77.9 ±1.2	53.5 ±2.5
SqrtReWeight	28.2 ±1.5	8.7 ±1.0	29.5 ±1.1	9.0 ±1.2	27.8 ±1.3	10.0 ±1.5	28.2 ±1.5	28.2 ±1.5	77.9 ±1.2	53.5 ±2.5
CBLoss	28.2 ±1.5	8.7 ±1.0	29.5 ±1.1	9.0 ±1.2	27.8 ±1.3	10.0 ±1.5	28.2 ±1.5	28.2 ±1.5	77.9 ±1.2	53.5 ±2.5
Focal	28.0 ±1.2	8.0 ±0.5	28.8 ±1.3	7.8 ±1.1	27.1 ±1.0	8.3 ±1.0	28.0 ±1.2	28.0 ±1.2	79.5 ±1.1	48.6 ±1.0
LDAM	24.7 ±0.8	7.0 ±0.5	28.3 ±0.6	6.0 ±0.4	24.5 ±0.6	6.7 ±0.3	24.7 ±0.8	24.7 ±0.8	78.1 ±1.2	9.7 ±2.7
BSoftmax	27.5 ±0.8	8.7 ±0.7	28.6 ±1.0	8.5 ±0.7	27.0 ±0.8	9.4 ±1.0	27.5 ±0.8	27.5 ±0.8	78.1 ±1.0	54.7 ±3.1
DFR	29.0 ±0.2	10.0 ±0.0	31.6 ±0.3	10.8 ±0.5	28.8 ±0.2	11.6 ±0.5	29.0 ±0.2	29.0 ±0.2	82.8 ±0.0	3.4 ±0.4
CRT	33.9 ±0.1	10.7 ±0.3	34.5 ±0.2	10.0 ±0.2	33.3 ±0.1	10.3 ±0.2	33.9 ±0.1	33.9 ±0.1	83.2 ±0.1	32.8 ±1.4
ReWeightCRT	33.7 ±0.1	7.7 ±0.3	33.9 ±0.1	15.3 ±0.6	33.1 ±0.1	11.5 ±0.4	33.7 ±0.1	33.7 ±0.1	82.5 ±0.0	41.4 ±0.2

Overall

Algorithm	Waterbirds	CelebA	CivilComments	MultiNLI	MetaShift	ImageNetBG	NICO++	MIMIC-CXR	MIMICNotes	CXRMultisite	CheXpert	Living17	Avg
ERM	69.1 ±4.7	62.6 ±1.5	63.7 ±1.1	66.8 ±0.5	82.6 ±0.4	81.0 ±0.9	37.6 ±2.0	68.9 ±0.3	83.1 ±0.1	50.1 ±0.9	50.2 ±3.8	28.2 ±1.5	62.0
Mixup	78.2 ±0.4	57.8 ±0.8	66.1 ±1.3	68.5 ±0.6	81.0 ±0.8	82.4 ±0.3	42.7 ±1.4	68.1 ±0.9	82.0 ±0.4	50.1 ±0.9	37.4 ±3.5	29.8 ±1.8	62.0
GroupDRO	78.6 ±1.0	89.0 ±0.7	70.6 ±1.2	76.0 ±0.7	85.6 ±0.4	78.2 ±0.3	37.8 ±1.8	74.4 ±0.2	83.7 ±0.1	59.6 ±1.0	74.5 ±0.2	27.2 ±1.5	69.6
IRM	74.5 ±1.5	63.0 ±2.5	63.2 ±0.8	63.6 ±1.3	83.0 ±0.1	81.3 ±0.3	40.0 ±0.0	67.7 ±0.2	83.2 ±0.2	47.9 ±1.1	34.4 ±1.7	28.2 ±1.5	60.8
CVaRDRO	75.5 ±2.2	64.1 ±2.8	68.7 ±1.3	63.0 ±1.5	84.6 ±0.0	80.7 ±1.1	36.7 ±2.7	68.6 ±0.4	81.9 ±0.1	50.2 ±0.9	57.9 ±0.4	28.3 ±0.7	63.3
JTT	72.0 ±0.3	70.0 ±10.2	64.3 ±1.5	69.1 ±0.1	83.6 ±0.4	80.5 ±0.3	40.0 ±0.0	67.3 ±0.7	83.8 ±0.1	50.1 ±0.9	61.3 ±4.9	28.8 ±1.1	64.2
LfF	75.2 ±0.7	53.0 ±4.3	51.0 ±6.1	63.6 ±2.9	73.1 ±1.6	76.7 ±0.5	30.4 ±1.3	62.6 ±2.6	84.0 ±0.1	50.1 ±0.9	13.7 ±9.8	26.2 ±1.1	55.0
LISA	88.7 ±0.6	86.5 ±1.2	73.7 ±0.3	73.3 ±1.0	84.1 ±0.4	76.1 ±0.8	42.7 ±2.2	70.4 ±0.2	83.6 ±0.2	48.9 ±1.3	75.6 ±0.6	29.8 ±0.9	69.5
MMD	83.9 ±1.4	24.4 ±2.0	54.5 ±1.4	69.1 ±1.5	85.9 ±0.7	80.8 ±0.5	40.7 ±0.5	68.0 ±0.6	82.0 ±0.5	50.1 ±0.9	50.2 ±3.8	26.6 ±1.8	59.7
ReSample	77.7 ±1.2	87.4 ±0.8	73.3 ±0.5	72.3 ±0.8	85.6 ±0.4	81.0 ±0.4	40.0 ±0.0	71.9 ±0.2	83.9 ±0.1	59.0 ±1.1	75.3 ±0.5	30.7 ±2.1	69.8
ReWeight	86.9 ±0.7	89.7 ±0.2	72.5 ±0.0	68.8 ±0.4	85.6 ±0.4	81.0 ±0.9	41.9 ±1.6	71.6 ±0.3	83.6 ±0.3	64.2 ±0.7	75.7 ±0.1	28.2 ±1.5	70.8
SqrtReWeight	78.6 ±0.1	82.4 ±0.5	71.7 ±0.4	69.5 ±0.7	84.6 ±0.7	80.1 ±0.2	40.0 ±0.0	70.3 ±0.2	83.1 ±0.1	50.0 ±1.0	70.0 ±2.3	28.2 ±1.5	67.4
CBLoss	86.2 ±0.3	89.4 ±0.7	73.3 ±0.2	72.2 ±0.3	85.5 ±0.4	81.0 ±0.9	37.8 ±1.8	70.7 ±0.1	83.2 ±0.1	50.1 ±0.9	74.7 ±0.3	28.2 ±1.5	69.4
Focal	71.6 ±0.8	59.1 ±2.0	62.0 ±1.0	69.4 ±0.7	81.5 ±0.0	78.4 ±0.1	36.7 ±2.7	68.7 ±0.4	71.1 ±9.9	50.0 ±0.9	42.1 ±4.0	28.0 ±1.2	59.9
LDAM	71.0 ±1.8	59.6 ±2.4	37.4 ±8.1	69.6 ±1.6	83.6 ±0.4	80.1 ±0.3	42.0 ±0.9	68.6 ±1.1	81.0 ±0.3	50.0 ±0.0	36.4 ±0.3	24.7 ±0.8	58.7
BSoftmax	74.1 ±0.9	83.3 ±0.5	71.2 ±0.4	66.9 ±0.4	83.1 ±0.7	80.7 ±0.7	40.4 ±0.3	68.4 ±0.2	83.4 ±0.3	50.1 ±1.1	75.4 ±0.5	27.5 ±0.8	67.0
DFR	91.0 ±0.3	90.4 ±0.1	69.6 ±0.2	68.5 ±0.2	85.4 ±0.4	78.5 ±0.6	23.7 ±0.7	68.9 ±0.0	83.6 ±0.0	53.5 ±0.2	71.7 ±0.2	29.0 ±0.2	67.8
CRT	79.7 ±0.3	87.2 ±0.3	71.1 ±0.1	70.7 ±0.1	84.1 ±0.4	80.2 ±0.3	43.3 ±2.7	71.0 ±0.0	83.4 ±0.0	65.2 ±0.1	74.6 ±0.3	33.9 ±0.1	70.4
ReWeightCRT	78.4 ±0.1	87.2 ±0.3	71.0 ±0.1	69.0 ±0.2	85.6 ±0.4	79.4 ±0.2	23.3 ±1.4	70.8 ±0.0	83.4 ±0.0	60.9 ±0.8	76.0 ±0.1	33.7 ±0.1	68.2

■ D.4.2 Attributes Unknown in Training, but Known in Validation

Waterbirds

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	84.1 ±1.7	69.1 ±4.7	77.4 ±2.0	60.7 ±3.2	79.4 ±2.1	69.5 ±2.9	83.1 ±2.0	83.1 ±2.0	91.0 ±1.4	12.9 ±1.7
Mixup	89.5 ±0.4	78.2 ±0.4	83.9 ±0.6	71.6 ±1.3	85.9 ±0.4	78.8 ±0.6	88.9 ±0.3	88.9 ±0.3	94.7 ±0.2	7.0 ±0.6
GroupDRO	86.9 ±0.9	73.1 ±0.4	80.7 ±1.1	66.1 ±2.2	82.8 ±0.9	74.4 ±1.2	86.3 ±0.5	86.3 ±0.5	94.0 ±0.3	10.5 ±0.8
CVaRDRO	89.9 ±0.4	75.5 ±2.2	84.5 ±0.7	73.2 ±1.7	86.2 ±0.3	79.0 ±0.4	88.5 ±0.3	88.5 ±0.3	95.4 ±0.2	8.3 ±0.2
JTT	88.9 ±0.6	71.0 ±0.5	83.2 ±0.8	71.5 ±1.7	84.7 ±0.6	76.8 ±0.8	86.8 ±0.2	86.8 ±0.2	94.2 ±0.1	9.1 ±0.3
LfF	86.5 ±0.6	74.7 ±1.0	80.1 ±0.7	64.7 ±1.4	82.3 ±0.6	73.8 ±0.8	86.2 ±0.3	86.2 ±0.3	93.5 ±0.2	9.9 ±0.8
LISA	89.5 ±0.4	78.2 ±0.4	83.9 ±0.6	71.6 ±1.3	85.9 ±0.4	78.8 ±0.6	88.9 ±0.3	88.9 ±0.3	94.7 ±0.2	7.0 ±0.6
ReSample	86.2 ±0.5	70.0 ±1.0	79.8 ±0.6	64.9 ±1.4	81.7 ±0.5	72.7 ±0.7	85.0 ±0.2	85.0 ±0.2	92.8 ±0.1	11.3 ±0.3
ReWeight	86.9 ±0.5	72.5 ±0.3	80.7 ±0.6	66.1 ±1.3	82.7 ±0.5	74.2 ±0.6	86.1 ±0.1	86.1 ±0.1	93.9 ±0.1	10.6 ±0.3
SqrtReWeight	89.7 ±0.4	71.3 ±1.4	84.3 ±0.6	73.6 ±0.9	85.7 ±0.6	78.2 ±0.8	87.4 ±0.5	87.4 ±0.5	94.5 ±0.4	8.8 ±0.4
CBLoss	86.8 ±0.6	74.4 ±1.2	80.4 ±0.7	65.5 ±1.3	82.6 ±0.7	74.0 ±1.0	86.2 ±0.6	86.2 ±0.6	93.5 ±0.4	11.3 ±0.4
Focal	89.3 ±0.2	71.6 ±0.8	83.7 ±0.3	72.4 ±0.5	85.2 ±0.3	77.5 ±0.4	87.1 ±0.3	87.1 ±0.3	94.2 ±0.2	6.9 ±0.1
LDAM	87.3 ±0.5	71.0 ±1.8	81.2 ±0.6	67.7 ±1.5	83.0 ±0.5	74.4 ±0.6	85.7 ±0.2	85.7 ±0.2	93.3 ±0.2	13.7 ±2.2
BSoftmax	88.4 ±1.2	74.1 ±0.9	82.6 ±1.6	69.9 ±2.9	84.4 ±1.5	76.4 ±2.0	87.0 ±1.0	87.0 ±1.0	94.0 ±0.9	9.9 ±1.2
DFR	92.2 ±0.2	89.0 ±0.2	87.7 ±0.3	78.4 ±0.5	89.2 ±0.2	83.6 ±0.3	91.2 ±0.1	91.2 ±0.1	96.8 ±0.1	6.8 ±0.4
CRT	89.2 ±0.1	76.3 ±0.8	83.5 ±0.1	71.3 ±0.4	85.3 ±0.1	77.8 ±0.1	87.9 ±0.1	87.9 ±0.1	94.8 ±0.0	9.2 ±0.2
ReWeightCRT	89.4 ±0.3	76.3 ±0.2	83.8 ±0.3	71.9 ±0.7	85.6 ±0.3	78.1 ±0.4	88.0 ±0.2	88.0 ±0.2	94.9 ±0.1	8.8 ±0.2

CelebA

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	95.0 ±0.1	57.6 ±0.8	87.4 ±0.2	76.3 ±0.3	89.9 ±0.1	82.7 ±0.2	85.6 ±0.2	93.0 ±0.0	98.4 ±0.0	2.9 ±0.1
Mixup	95.4 ±0.1	57.8 ±0.8	88.4 ±0.3	78.5 ±0.7	90.6 ±0.2	83.8 ±0.3	85.8 ±0.2	93.1 ±0.1	98.4 ±0.1	2.5 ±0.2
GroupDRO	92.4 ±0.2	78.5 ±1.1	81.9 ±0.3	64.5 ±0.5	86.3 ±0.2	77.1 ±0.4	90.1 ±0.3	93.9 ±0.1	98.5 ±0.0	7.4 ±0.5
CVaRDRO	95.1 ±0.1	62.2 ±3.1	87.8 ±0.2	77.1 ±0.5	90.2 ±0.1	83.2 ±0.1	86.8 ±0.7	93.2 ±0.2	98.4 ±0.1	3.0 ±0.2
JTT	88.1 ±0.5	66.0 ±11.9	75.2 ±1.1	54.1 ±1.0	76.7 ±3.3	60.5 ±6.6	81.7 ±5.5	82.7 ±6.7	91.3 ±0.2	5.0 ±1.0
LfF	81.1 ±5.6	53.0 ±4.3	71.8 ±4.1	45.2 ±8.3	73.2 ±5.6	59.0 ±7.3	78.3 ±3.0	85.3 ±2.9	94.1 ±1.2	27.9 ±5.5
LISA	95.4 ±0.1	57.8 ±0.8	88.4 ±0.3	78.5 ±0.7	90.6 ±0.2	83.8 ±0.3	85.8 ±0.2	93.1 ±0.1	98.4 ±0.1	2.5 ±0.2
ReSample	92.2 ±0.4	82.2 ±1.2	81.5 ±0.6	63.7 ±1.3	85.9 ±0.5	76.6 ±0.8	90.8 ±0.1	93.8 ±0.1	98.5 ±0.0	7.4 ±0.8
ReWeight	92.0 ±0.4	81.5 ±0.9	81.3 ±0.6	63.2 ±1.4	85.7 ±0.6	76.3 ±0.8	90.7 ±0.2	93.8 ±0.1	98.4 ±0.1	7.8 ±0.8
SqrtReWeight	93.7 ±0.2	72.0 ±2.2	84.3 ±0.5	69.4 ±1.1	88.1 ±0.4	80.0 ±0.6	89.0 ±0.5	94.0 ±0.0	98.4 ±0.0	5.3 ±0.4
CBLoss	93.8 ±0.3	75.0 ±2.4	84.5 ±0.7	69.9 ±1.6	88.3 ±0.5	80.3 ±0.8	89.8 ±0.5	94.0 ±0.1	98.5 ±0.0	5.1 ±0.6
Focal	94.9 ±0.3	59.1 ±2.0	87.5 ±0.8	76.7 ±1.7	89.7 ±0.4	82.4 ±0.6	85.6 ±0.5	92.5 ±0.4	98.2 ±0.1	3.2 ±0.4
LDAM	94.5 ±0.2	59.3 ±2.3	86.5 ±0.8	74.7 ±1.9	89.1 ±0.2	81.4 ±0.3	85.6 ±0.8	92.5 ±0.7	98.2 ±0.1	28.0 ±2.6
BSoftmax	91.9 ±0.1	83.3 ±0.5	81.1 ±0.2	62.9 ±0.4	85.6 ±0.2	76.1 ±0.3	91.1 ±0.2	93.9 ±0.1	98.6 ±0.0	8.4 ±0.2
DFR	91.2 ±0.1	86.3 ±0.3	80.0 ±0.1	61.0 ±0.2	84.4 ±0.1	74.1 ±0.1	90.8 ±0.0	92.6 ±0.0	97.9 ±0.0	14.1 ±0.0
CRT	94.1 ±0.1	70.4 ±0.4	85.1 ±0.2	71.3 ±0.5	88.6 ±0.1	80.6 ±0.2	88.5 ±0.1	93.5 ±0.1	98.4 ±0.0	4.5 ±0.2
ReWeightCRT	94.2 ±0.1	71.1 ±0.5	85.3 ±0.1	71.8 ±0.3	88.7 ±0.1	80.9 ±0.1	88.7 ±0.1	93.6 ±0.1	98.4 ±0.0	4.6 ±0.2

CivilComments

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	85.4 ±0.2	63.2 ±1.2	75.4 ±0.3	57.6 ±0.7	77.0 ±0.1	63.2 ±0.1	77.7 ±0.1	79.4 ±0.2	89.8 ±0.1	7.8 ±0.4
Mixup	84.9 ±0.3	66.1 ±1.3	74.8 ±0.3	56.2 ±0.8	76.7 ±0.2	62.8 ±0.3	78.0 ±0.1	79.6 ±0.1	89.8 ±0.1	9.0 ±0.6
GroupDRO	83.1 ±0.3	69.5 ±0.7	73.2 ±0.2	52.0 ±0.6	75.7 ±0.2	62.2 ±0.2	79.1 ±0.1	80.9 ±0.1	89.9 ±0.1	12.3 ±0.5
CVaRDRO	83.5 ±0.3	68.7 ±1.3	73.5 ±0.3	52.8 ±0.6	75.9 ±0.2	62.4 ±0.2	78.6 ±0.2	80.7 ±0.1	89.8 ±0.1	32.9 ±0.4
JTT	83.3 ±0.1	64.3 ±1.5	72.8 ±0.1	52.4 ±0.3	74.8 ±0.1	60.3 ±0.2	76.8 ±0.2	78.4 ±0.2	88.2 ±0.1	10.2 ±0.3
LfF	68.2 ±5.0	50.3 ±5.9	62.9 ±3.5	34.1 ±5.4	61.5 ±4.7	45.8 ±5.5	68.5 ±4.4	69.8 ±4.8	75.0 ±6.5	30.8 ±2.7
LISA	84.9 ±0.3	66.1 ±1.3	74.8 ±0.3	56.2 ±0.8	76.7 ±0.2	62.8 ±0.3	78.0 ±0.1	79.6 ±0.1	89.8 ±0.1	9.0 ±0.6
ReSample	82.5 ±0.6	68.2 ±0.7	72.7 ±0.5	51.0 ±1.3	75.0 ±0.5	61.2 ±0.5	78.4 ±0.1	80.3 ±0.1	89.3 ±0.1	13.8 ±1.0
ReWeight	83.1 ±0.1	69.9 ±0.6	73.2 ±0.1	52.0 ±0.1	75.6 ±0.1	62.1 ±0.1	78.8 ±0.1	80.7 ±0.1	89.8 ±0.0	11.0 ±0.2
SqrtReWeight	83.6 ±0.1	70.1 ±0.3	73.6 ±0.1	52.9 ±0.2	76.0 ±0.1	62.4 ±0.2	78.7 ±0.2	80.7 ±0.2	89.9 ±0.1	10.1 ±0.1
CBLoss	84.1 ±0.7	67.0 ±0.1	74.1 ±0.7	54.3 ±1.6	76.2 ±0.5	62.6 ±0.5	78.4 ±0.1	80.2 ±0.2	90.0 ±0.0	9.1 ±1.0
Focal	85.6 ±0.3	61.9 ±1.1	75.6 ±0.4	58.5 ±0.9	77.0 ±0.3	62.9 ±0.5	77.3 ±0.3	78.7 ±0.3	89.4 ±0.4	7.7 ±0.4
LDAM	81.8 ±2.2	37.0 ±7.9	69.4 ±3.4	49.7 ±5.8	69.4 ±3.3	49.9 ±5.2	67.1 ±3.9	69.5 ±3.2	79.0 ±3.8	21.0 ±0.2
BSoftmax	83.0 ±0.4	69.4 ±1.2	73.1 ±0.3	51.8 ±0.7	75.5 ±0.3	61.9 ±0.3	78.7 ±0.3	80.6 ±0.3	89.7 ±0.2	12.1 ±0.9
DFR	81.3 ±0.0	66.5 ±0.2	71.2 ±0.0	48.6 ±0.0	73.4 ±0.0	59.0 ±0.0	76.8 ±0.0	78.8 ±0.0	86.7 ±0.1	19.6 ±0.1
CRT	83.0 ±0.0	68.5 ±0.0	73.0 ±0.0	51.7 ±0.1	75.4 ±0.0	61.8 ±0.0	78.6 ±0.0	80.6 ±0.0	89.4 ±0.1	12.5 ±0.1
ReWeightCRT	83.4 ±0.0	68.2 ±0.4	73.3 ±0.0	52.5 ±0.1	75.7 ±0.0	62.0 ±0.0	78.3 ±0.0	80.4 ±0.0	89.4 ±0.0	11.5 ±0.2

MultiNLI

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	80.9 ±0.3	69.5 ±0.3	81.3 ±0.3	73.7 ±0.4	81.0 ±0.3	77.6 ±0.3	79.7 ±0.3	80.9 ±0.3	93.6 ±0.1	10.7 ±0.5
Mixup	81.4 ±0.3	68.5 ±0.6	81.6 ±0.3	76.0 ±0.5	81.4 ±0.3	78.0 ±0.2	80.1 ±0.3	81.4 ±0.3	93.6 ±0.1	9.4 ±0.9
GroupDRO	81.0 ±0.0	69.3 ±1.5	81.3 ±0.1	74.9 ±0.6	81.1 ±0.1	77.7 ±0.1	79.5 ±0.3	81.0 ±0.0	93.8 ±0.0	8.7 ±0.4
CVaRDRO	75.1 ±0.1	63.0 ±1.5	76.2 ±0.2	65.6 ±0.2	75.2 ±0.1	72.1 ±0.2	74.2 ±0.4	75.1 ±0.1	86.3 ±0.2	41.4 ±0.1
JTT	81.4 ±0.1	68.4 ±0.6	81.6 ±0.1	75.7 ±0.3	81.5 ±0.1	78.1 ±0.1	80.2 ±0.2	81.4 ±0.1	93.9 ±0.0	9.4 ±0.5
LfF	71.7 ±1.1	63.6 ±2.9	71.8 ±1.1	68.7 ±0.7	71.7 ±1.1	68.5 ±1.8	70.8 ±1.4	71.7 ±1.1	87.0 ±0.8	4.4 ±0.6
LISA	81.4 ±0.3	68.5 ±0.6	81.6 ±0.3	76.0 ±0.5	81.4 ±0.3	78.0 ±0.2	80.1 ±0.3	81.4 ±0.3	93.6 ±0.1	9.4 ±0.9
ReSample	81.4 ±0.3	67.5 ±0.4	81.7 ±0.3	74.7 ±0.6	81.4 ±0.3	77.8 ±0.3	79.9 ±0.2	81.4 ±0.3	93.8 ±0.1	11.3 ±1.3
ReWeight	79.2 ±0.4	67.8 ±1.2	79.5 ±0.3	73.0 ±0.5	79.3 ±0.4	75.7 ±0.3	78.4 ±0.2	79.2 ±0.4	92.5 ±0.1	13.1 ±1.5
SqrtReWeight	80.9 ±0.1	66.6 ±0.4	81.1 ±0.1	76.0 ±0.3	80.9 ±0.1	77.7 ±0.1	79.6 ±0.0	80.9 ±0.1	93.6 ±0.1	8.1 ±0.2
CBLoss	81.1 ±0.2	66.2 ±0.7	81.2 ±0.2	76.6 ±0.3	81.1 ±0.2	77.8 ±0.1	79.7 ±0.0	81.1 ±0.2	93.7 ±0.1	8.9 ±0.4
Focal	80.7 ±0.2	69.3 ±0.8	81.2 ±0.2	73.5 ±0.5	80.8 ±0.2	77.4 ±0.2	79.5 ±0.1	80.7 ±0.2	93.6 ±0.1	4.5 ±1.1
LDAM	80.7 ±0.3	69.6 ±1.6	81.1 ±0.1	73.9 ±0.9	80.8 ±0.2	77.4 ±0.2	79.7 ±0.3	80.7 ±0.3	93.5 ±0.1	33.4 ±0.3
BSoftmax	80.9 ±0.1	66.9 ±0.4	81.1 ±0.1	75.9 ±0.3	80.9 ±0.1	77.7 ±0.1	79.7 ±0.0	80.9 ±0.1	93.6 ±0.1	8.1 ±0.2
DFR	80.2 ±0.0	63.8 ±0.0	80.3 ±0.0	75.2 ±0.0	80.3 ±0.0	76.2 ±0.0	78.5 ±0.0	80.2 ±0.0	92.9 ±0.0	5.7 ±0.0
CRT	80.2 ±0.0	65.4 ±0.1	80.3 ±0.0	74.4 ±0.1	80.2 ±0.0	76.4 ±0.0	78.6 ±0.0	80.2 ±0.0	92.9 ±0.0	14.9 ±0.1
ReWeightCRT	80.2 ±0.0	65.3 ±0.1	80.3 ±0.0	74.4 ±0.0	80.2 ±0.0	76.4 ±0.0	78.6 ±0.0	80.2 ±0.0	92.9 ±0.0	15.0 ±0.2

MetaShift

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	91.5 ±0.2	82.1 ±0.8	91.5 ±0.1	90.7 ±0.3	91.4 ±0.2	90.9 ±0.2	89.4 ±0.3	91.4 ±0.2	97.5 ±0.1	5.7 ±0.4
Mixup	91.4 ±0.2	79.0 ±0.8	91.4 ±0.2	90.9 ±0.2	91.3 ±0.2	90.8 ±0.2	88.7 ±0.1	91.3 ±0.2	97.2 ±0.0	1.7 ±0.2
GroupDRO	91.5 ±0.3	82.6 ±1.1	91.5 ±0.3	90.8 ±0.3	91.5 ±0.3	91.0 ±0.3	89.5 ±0.4	91.5 ±0.3	97.5 ±0.1	5.9 ±0.5
CVaRDRO	91.5 ±0.2	82.6 ±1.1	91.5 ±0.2	90.7 ±0.5	91.5 ±0.2	91.0 ±0.2	89.5 ±0.4	91.5 ±0.2	97.5 ±0.1	7.8 ±2.1
JTT	91.5 ±0.2	82.6 ±0.4	91.5 ±0.2	91.0 ±0.1	91.5 ±0.2	90.9 ±0.3	89.6 ±0.1	91.4 ±0.3	97.6 ±0.1	6.5 ±0.1
LfF	80.3 ±0.4	72.6 ±1.2	80.6 ±0.3	77.5 ±1.4	80.2 ±0.3	78.9 ±0.4	80.4 ±0.6	80.2 ±0.2	90.7 ±0.6	8.0 ±1.3
LISA	91.4 ±0.2	79.0 ±0.8	91.4 ±0.2	90.9 ±0.2	91.3 ±0.2	90.8 ±0.2	88.7 ±0.1	91.3 ±0.2	97.2 ±0.0	1.7 ±0.2
ReSample	92.1 ±0.3	80.5 ±1.5	92.1 ±0.3	91.4 ±0.4	92.1 ±0.3	91.6 ±0.3	89.5 ±0.1	92.1 ±0.3	97.5 ±0.1	6.7 ±0.4
ReWeight	91.2 ±0.5	83.1 ±0.7	91.1 ±0.5	90.2 ±0.4	91.1 ±0.5	90.6 ±0.6	89.3 ±0.5	91.1 ±0.5	97.4 ±0.2	6.6 ±0.8
SqrtReWeight	91.1 ±0.2	82.1 ±0.8	91.1 ±0.2	90.5 ±0.2	91.1 ±0.2	90.5 ±0.2	89.1 ±0.3	91.1 ±0.2	97.4 ±0.1	6.7 ±0.3
CBLoss	91.2 ±0.1	82.6 ±0.4	91.2 ±0.1	90.7 ±0.2	91.2 ±0.1	90.6 ±0.1	89.3 ±0.2	91.1 ±0.1	97.4 ±0.0	6.7 ±0.1
Focal	91.6 ±0.2	81.0 ±0.4	91.6 ±0.3	91.2 ±0.4	91.6 ±0.2	91.0 ±0.2	89.5 ±0.2	91.5 ±0.2	97.6 ±0.0	3.3 ±0.4
LDAM	91.7 ±0.0	83.6 ±0.4	91.7 ±0.0	90.9 ±0.3	91.7 ±0.0	91.2 ±0.0	90.0 ±0.1	91.7 ±0.0	97.5 ±0.1	9.9 ±0.7
BSoftmax	91.2 ±0.3	82.6 ±0.4	91.1 ±0.3	90.2 ±0.2	91.2 ±0.3	90.7 ±0.3	89.2 ±0.3	91.2 ±0.3	97.4 ±0.1	6.8 ±0.3
DFR	90.5 ±0.4	81.5 ±0.0	90.5 ±0.4	89.2 ±0.3	90.5 ±0.4	90.0 ±0.4	88.2 ±0.3	90.6 ±0.4	96.7 ±0.0	3.2 ±0.2
CRT	91.5 ±0.0	83.1 ±0.0	91.4 ±0.0	90.6 ±0.1	91.4 ±0.0	90.9 ±0.0	89.5 ±0.0	91.4 ±0.0	97.3 ±0.0	6.8 ±0.0
ReWeightCRT	91.3 ±0.1	85.1 ±0.4	91.2 ±0.1	90.1 ±0.3	91.2 ±0.1	90.8 ±0.1	89.7 ±0.2	91.3 ±0.1	96.8 ±0.1	8.1 ±0.1

ImageNetBG

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	87.7 ±0.1	76.8 ±0.9	87.9 ±0.1	78.3 ±1.3	87.7 ±0.1	80.7 ±0.4	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
Mixup	87.9 ±0.2	76.9 ±0.7	88.4 ±0.1	76.6 ±2.6	88.0 ±0.2	80.5 ±1.0	87.9 ±0.2	87.9 ±0.2	98.7 ±0.0	4.7 ±1.6
GroupDRO	87.7 ±0.1	76.4 ±0.2	87.9 ±0.1	76.2 ±0.5	87.6 ±0.1	81.1 ±0.3	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
CVaRDRO	87.8 ±0.2	74.8 ±0.8	88.0 ±0.1	79.9 ±0.9	87.8 ±0.2	79.5 ±0.3	87.8 ±0.2	87.8 ±0.2	99.0 ±0.0	5.6 ±0.2
JTT	87.6 ±0.4	77.0 ±0.4	87.8 ±0.3	78.3 ±3.0	87.5 ±0.3	80.4 ±0.6	87.6 ±0.4	87.6 ±0.4	99.0 ±0.0	3.7 ±0.2
LfF	84.7 ±0.5	70.1 ±1.4	85.4 ±0.3	72.1 ±3.1	84.7 ±0.5	76.2 ±0.6	84.7 ±0.5	84.7 ±0.5	98.6 ±0.0	1.8 ±0.4
LISA	87.9 ±0.2	76.9 ±0.7	88.4 ±0.1	76.6 ±2.6	88.0 ±0.2	80.5 ±1.0	87.9 ±0.2	87.9 ±0.2	98.7 ±0.0	4.7 ±1.6
ReSample	88.2 ±0.4	77.7 ±1.1	88.4 ±0.4	79.7 ±1.0	88.2 ±0.4	80.6 ±1.0	88.2 ±0.4	88.2 ±0.4	99.0 ±0.0	5.4 ±0.5
ReWeight	87.7 ±0.1	76.8 ±0.9	87.9 ±0.1	78.3 ±1.3	87.7 ±0.1	80.7 ±0.4	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
SqrtReWeight	87.7 ±0.1	76.8 ±0.9	87.9 ±0.1	78.3 ±1.3	87.7 ±0.1	80.7 ±0.4	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
CBLoss	87.7 ±0.1	76.8 ±0.9	87.9 ±0.1	78.3 ±1.3	87.7 ±0.1	80.7 ±0.4	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
Focal	86.7 ±0.2	71.9 ±1.2	87.1 ±0.2	74.2 ±1.3	86.6 ±0.2	77.6 ±0.7	86.7 ±0.2	86.7 ±0.2	98.9 ±0.0	2.8 ±0.6
LDAM	88.2 ±0.1	76.7 ±0.5	88.5 ±0.1	77.6 ±1.1	88.1 ±0.1	81.3 ±0.4	88.2 ±0.1	88.2 ±0.1	98.8 ±0.0	45.9 ±0.7
BSoftmax	87.7 ±0.1	76.1 ±2.0	88.0 ±0.1	77.7 ±1.3	87.7 ±0.1	80.4 ±0.8	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.6 ±0.5
DFR	86.8 ±0.5	74.4 ±1.8	86.9 ±0.5	78.9 ±1.6	86.7 ±0.5	78.1 ±1.3	86.8 ±0.5	86.8 ±0.5	98.8 ±0.1	8.9 ±1.4
CRT	88.3 ±0.1	78.2 ±0.5	88.3 ±0.1	82.7 ±0.4	88.3 ±0.1	80.9 ±0.2	88.3 ±0.1	88.3 ±0.1	99.1 ±0.0	5.6 ±0.2
ReWeightCRT	88.4 ±0.1	77.5 ±0.7	88.5 ±0.1	82.1 ±0.4	88.4 ±0.1	81.2 ±0.3	88.4 ±0.1	88.4 ±0.1	99.1 ±0.0	5.4 ±0.2

NICO++

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	85.3 ±0.3	40.0 ±0.0	86.2 ±0.2	59.2 ±1.1	85.4 ±0.3	66.0 ±0.6	84.8 ±0.3	85.2 ±0.3	99.4 ±0.0	9.6 ±0.2
Mixup	85.5 ±0.2	30.0 ±4.1	86.5 ±0.1	55.7 ±2.5	85.7 ±0.2	66.5 ±1.6	85.0 ±0.2	85.4 ±0.2	99.1 ±0.0	2.0 ±0.3
GroupDRO	83.0 ±0.3	31.1 ±0.9	84.5 ±0.2	54.4 ±3.9	83.2 ±0.3	64.9 ±0.9	82.6 ±0.3	82.9 ±0.3	99.3 ±0.0	7.4 ±0.5
CVaRDRO	84.4 ±0.6	31.7 ±3.6	85.5 ±0.5	57.1 ±1.6	84.6 ±0.6	66.6 ±1.5	84.0 ±0.6	84.3 ±0.6	99.4 ±0.0	7.8 ±1.3
JTT	85.4 ±0.2	32.2 ±0.9	86.2 ±0.2	57.8 ±4.6	85.5 ±0.2	65.1 ±2.8	84.9 ±0.2	85.3 ±0.2	99.4 ±0.0	10.2 ±0.2
LfF	78.5 ±0.6	28.3 ±1.7	80.9 ±0.3	44.3 ±0.8	78.8 ±0.6	54.4 ±1.3	78.1 ±0.6	78.4 ±0.6	99.2 ±0.0	1.8 ±0.2
LISA	85.5 ±0.2	30.0 ±4.1	86.5 ±0.1	55.7 ±2.5	85.7 ±0.2	66.5 ±1.6	85.0 ±0.2	85.4 ±0.2	99.1 ±0.0	2.0 ±0.3
ReSample	84.8 ±0.4	23.3 ±1.4	85.7 ±0.3	58.9 ±2.7	84.9 ±0.4	65.5 ±1.1	84.2 ±0.4	84.7 ±0.4	99.3 ±0.0	10.1 ±0.0
ReWeight	85.8 ±0.1	25.0 ±0.0	86.6 ±0.1	59.8 ±1.1	85.9 ±0.1	69.1 ±0.5	85.3 ±0.1	85.7 ±0.1	99.4 ±0.0	9.4 ±0.2
SqrtReWeight	85.4 ±0.2	35.6 ±1.8	86.4 ±0.1	57.5 ±1.4	85.6 ±0.1	66.7 ±1.2	84.9 ±0.2	85.3 ±0.2	99.4 ±0.0	9.3 ±0.0
CBLoss	85.1 ±0.4	34.4 ±2.4	85.9 ±0.3	56.9 ±1.1	85.2 ±0.4	64.9 ±0.1	84.7 ±0.4	85.1 ±0.4	99.4 ±0.0	9.1 ±0.7
Focal	85.1 ±0.5	29.4 ±2.0	86.0 ±0.3	58.1 ±2.2	85.3 ±0.4	65.0 ±0.8	84.6 ±0.5	85.0 ±0.5	99.4 ±0.0	4.6 ±1.1
LDAM	84.7 ±0.4	26.7 ±1.4	85.6 ±0.3	60.4 ±1.4	84.8 ±0.4	65.3 ±0.7	84.2 ±0.4	84.6 ±0.4	98.9 ±0.1	62.6 ±2.0
BSoftmax	85.2 ±0.3	29.4 ±2.0	85.9 ±0.2	57.6 ±2.1	85.3 ±0.3	66.9 ±0.7	84.8 ±0.3	85.1 ±0.3	99.4 ±0.0	8.4 ±0.9
DFR	82.5 ±0.0	39.3 ±2.4	83.3 ±0.0	55.3 ±1.3	82.6 ±0.0	63.7 ±0.9	82.1 ±0.0	82.4 ±0.0	99.2 ±0.0	11.7 ±0.1
CRT	85.7 ±0.0	33.3 ±0.0	86.1 ±0.0	64.4 ±0.2	85.7 ±0.0	69.1 ±0.1	85.2 ±0.0	85.6 ±0.0	99.4 ±0.0	4.7 ±0.0
ReWeightCRT	85.8 ±0.1	33.3 ±0.0	86.1 ±0.1	66.1 ±0.5	85.8 ±0.1	69.9 ±0.2	85.4 ±0.1	85.7 ±0.1	99.4 ±0.0	6.2 ±0.2

MIMIC-CXR

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	78.6 ±0.2	68.5 ±0.4	77.7 ±0.2	72.2 ±0.5	77.8 ±0.2	73.7 ±0.2	77.5 ±0.2	78.1 ±0.2	85.5 ±0.2	2.6 ±0.2
Mixup	78.5 ±0.1	67.2 ±1.1	77.5 ±0.1	72.3 ±0.5	77.6 ±0.0	73.4 ±0.1	77.2 ±0.1	77.8 ±0.1	85.3 ±0.1	2.4 ±0.7
GroupDRO	78.3 ±0.1	67.4 ±0.9	77.4 ±0.1	71.9 ±0.2	77.6 ±0.1	73.4 ±0.1	77.2 ±0.1	77.8 ±0.1	85.1 ±0.0	3.7 ±0.4
CVaRDRO	78.4 ±0.3	67.5 ±0.1	77.4 ±0.3	72.1 ±0.6	77.6 ±0.3	73.4 ±0.2	77.2 ±0.2	77.7 ±0.2	84.9 ±0.2	5.7 ±0.3
JTT	78.2 ±0.1	66.6 ±0.8	77.3 ±0.1	71.9 ±0.3	77.4 ±0.1	73.2 ±0.1	77.0 ±0.1	77.6 ±0.1	85.0 ±0.1	3.3 ±0.1
LfF	73.9 ±1.1	62.1 ±2.4	72.9 ±1.1	66.5 ±1.1	73.0 ±1.2	68.0 ±1.6	72.9 ±1.2	73.2 ±1.2	79.9 ±1.5	11.3 ±0.3
LISA	78.5 ±0.1	67.2 ±1.1	77.5 ±0.1	72.3 ±0.5	77.6 ±0.0	73.4 ±0.1	77.2 ±0.1	77.8 ±0.1	85.3 ±0.1	2.4 ±0.7
ReSample	78.7 ±0.1	68.9 ±0.3	77.8 ±0.1	72.2 ±0.1	78.0 ±0.1	74.0 ±0.1	77.7 ±0.1	78.2 ±0.1	85.4 ±0.1	3.6 ±0.1
ReWeight	78.0 ±0.1	67.4 ±0.3	77.1 ±0.0	70.9 ±0.2	77.3 ±0.0	73.3 ±0.0	77.0 ±0.0	77.6 ±0.0	84.9 ±0.0	4.1 ±0.6
SqrtReWeight	78.5 ±0.2	68.9 ±0.5	77.6 ±0.2	71.6 ±0.3	77.7 ±0.2	73.8 ±0.2	77.4 ±0.2	78.1 ±0.2	85.4 ±0.1	3.6 ±0.4
CBLoss	78.6 ±0.1	67.8 ±0.6	77.6 ±0.1	72.3 ±0.4	77.8 ±0.1	73.6 ±0.2	77.4 ±0.1	78.0 ±0.1	85.4 ±0.1	3.2 ±0.3
Focal	78.3 ±0.1	67.3 ±1.2	77.4 ±0.1	71.6 ±0.1	77.5 ±0.1	73.5 ±0.2	77.2 ±0.2	77.8 ±0.1	85.3 ±0.1	10.0 ±0.6
LDAM	78.5 ±0.1	68.2 ±1.4	77.6 ±0.1	72.0 ±0.4	77.7 ±0.1	73.6 ±0.3	77.5 ±0.2	78.0 ±0.2	85.3 ±0.1	22.2 ±0.1
BSoftmax	78.2 ±0.2	67.2 ±0.2	77.3 ±0.2	71.9 ±0.5	77.4 ±0.2	73.2 ±0.1	77.1 ±0.1	77.6 ±0.1	85.1 ±0.1	3.6 ±0.5
DFR	78.3 ±0.0	67.1 ±0.4	77.3 ±0.0	72.0 ±0.2	77.5 ±0.0	73.2 ±0.1	77.1 ±0.0	77.6 ±0.1	85.0 ±0.0	20.0 ±0.1
CRT	78.3 ±0.0	69.1 ±0.2	77.4 ±0.0	71.1 ±0.1	77.6 ±0.0	73.7 ±0.0	77.4 ±0.0	78.0 ±0.0	85.3 ±0.0	4.7 ±0.1
ReWeightCRT	77.9 ±0.0	68.9 ±0.0	77.0 ±0.0	70.5 ±0.0	77.3 ±0.0	73.4 ±0.0	77.0 ±0.0	77.7 ±0.0	85.0 ±0.0	4.9 ±0.3

MIMICNotes

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	91.1 ±0.1	24.2 ±2.8	76.4 ±0.9	60.3 ±2.1	65.8 ±1.2	36.5 ±2.5	62.2 ±1.3	62.4 ±1.4	85.2 ±0.1	2.2 ±0.1
Mixup	91.1 ±0.0	22.7 ±3.2	76.8 ±0.7	61.2 ±1.6	65.1 ±1.6	35.0 ±3.2	61.5 ±1.6	61.7 ±1.7	85.4 ±0.0	2.0 ±0.8
GroupDRO	83.2 ±2.4	62.6 ±6.3	64.7 ±1.4	33.7 ±3.4	66.4 ±1.3	42.8 ±1.1	74.3 ±1.5	74.4 ±1.5	85.1 ±0.1	13.8 ±3.2
CVaRDRO	90.2 ±0.0	0.0 ±0.0	45.1 ±0.0	0.0 ±0.0	47.4 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	71.6 ±0.4	40.2 ±0.0
JTT	71.3 ±3.7	65.9 ±2.8	60.3 ±0.7	23.4 ±1.9	58.6 ±2.3	36.0 ±1.8	75.5 ±0.4	75.6 ±0.4	84.9 ±0.1	27.5 ±3.9
LfF	84.0 ±1.2	62.7 ±2.1	64.6 ±0.7	33.6 ±1.6	67.1 ±0.8	43.6 ±0.8	74.7 ±0.4	74.7 ±0.5	85.1 ±0.0	12.5 ±1.2
LISA	91.1 ±0.0	22.7 ±3.2	76.8 ±0.7	61.2 ±1.6	65.1 ±1.6	35.0 ±3.2	61.5 ±1.6	61.7 ±1.7	85.4 ±0.0	2.0 ±0.8
ReSample	81.4 ±1.5	67.1 ±2.6	63.3 ±0.6	30.5 ±1.5	65.4 ±1.0	42.0 ±1.0	75.4 ±0.3	75.6 ±0.4	85.1 ±0.0	17.4 ±1.9
ReWeight	82.7 ±0.7	65.5 ±1.3	63.8 ±0.4	31.7 ±0.9	66.3 ±0.5	42.8 ±0.5	75.3 ±0.2	75.4 ±0.3	85.2 ±0.1	15.9 ±0.7
SqrtReWeight	90.3 ±0.2	35.7 ±4.0	72.4 ±0.9	51.3 ±2.1	68.7 ±0.9	42.8 ±2.1	66.7 ±1.6	66.8 ±1.6	85.2 ±0.1	3.7 ±0.7
CBLoss	78.2 ±1.0	72.3 ±1.3	61.9 ±0.3	27.3 ±0.8	63.2 ±0.7	39.8 ±0.6	76.1 ±0.2	76.2 ±0.2	85.0 ±0.0	20.6 ±1.3
Focal	91.0 ±0.0	19.1 ±2.3	77.1 ±0.6	62.1 ±1.4	63.6 ±1.3	31.9 ±2.6	59.9 ±1.1	60.2 ±1.1	85.3 ±0.1	8.1 ±0.7
LDAM	90.6 ±0.1	5.3 ±2.4	84.4 ±0.8	78.1 ±1.7	52.5 ±2.1	10.0 ±4.1	52.7 ±1.2	52.7 ±1.2	84.9 ±0.1	28.9 ±1.0
BSoftmax	76.9 ±0.9	73.1 ±1.0	61.7 ±0.2	26.5 ±0.6	62.5 ±0.5	39.3 ±0.4	76.6 ±0.2	76.7 ±0.2	85.4 ±0.0	23.5 ±1.1
DFR	69.2 ±1.3	67.3 ±1.7	58.8 ±0.2	21.0 ±0.5	56.5 ±0.8	33.1 ±0.5	73.1 ±0.0	73.1 ±0.0	81.0 ±0.0	38.4 ±0.1
CRT	77.8 ±0.0	73.1 ±0.0	61.6 ±0.0	26.7 ±0.0	62.8 ±0.0	39.2 ±0.0	75.9 ±0.0	75.9 ±0.0	84.3 ±0.0	23.0 ±0.1
ReWeightCRT	81.2 ±2.8	63.9 ±7.6	63.6 ±1.6	31.4 ±3.9	64.7 ±1.6	40.7 ±1.3	73.8 ±1.6	73.9 ±1.6	84.3 ±0.0	26.5 ±2.4

CXRMultisite

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	93.1 ±0.1	0.3 ±0.1
Mixup	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	92.9 ±0.1	0.3 ±0.0
GroupDRO	90.2 ±0.1	0.0 ±0.0	56.7 ±0.1	13.6 ±0.1	59.2 ±0.1	23.7 ±0.2	50.2 ±0.1	90.4 ±0.0	92.8 ±0.2	13.5 ±0.7
CVaRDRO	98.3 ±0.0	0.0 ±0.0	61.2 ±4.9	24.0 ±9.8	50.7 ±0.7	2.2 ±1.5	50.2 ±0.2	50.6 ±0.4	93.0 ±0.0	0.9 ±0.3
JTT	94.1 ±0.9	0.0 ±0.0	59.0 ±0.7	18.5 ±1.4	62.9 ±0.8	28.9 ±1.2	55.2 ±0.9	82.2 ±2.4	93.2 ±0.1	6.4 ±0.5
LfF	9.9 ±6.7	5.4 ±4.4	17.4 ±13.5	0.6 ±0.5	8.5 ±5.6	1.2 ±1.0	50.5 ±0.4	51.7 ±1.4	60.6 ±1.6	82.6 ±12.8
LISA	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	92.9 ±0.1	0.3 ±0.0
ReSample	88.3 ±1.6	0.1 ±0.1	55.9 ±0.6	12.0 ±1.2	57.3 ±1.4	20.9 ±1.8	50.3 ±0.5	88.1 ±0.7	92.3 ±0.1	13.0 ±2.8
ReWeight	89.5 ±0.0	0.3 ±0.1	56.4 ±0.0	13.0 ±0.0	58.5 ±0.0	22.7 ±0.0	50.5 ±0.2	90.3 ±0.0	93.2 ±0.1	17.7 ±1.7
SqrtReWeight	94.5 ±0.4	0.0 ±0.0	59.4 ±0.2	19.3 ±0.6	63.7 ±0.3	30.2 ±0.4	56.3 ±0.1	82.3 ±1.6	93.3 ±0.0	6.0 ±0.2
CBLoss	1.7 ±0.0	0.0 ±0.0	0.8 ±0.0	0.0 ±0.0	1.7 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	62.7 ±1.5	98.3 ±0.0
Focal	98.3 ±0.0	0.0 ±0.0	55.4 ±5.1	12.5 ±10.2	49.7 ±0.1	0.3 ±0.2	50.0 ±0.0	50.1 ±0.1	93.2 ±0.0	11.5 ±0.6
LDAM	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	93.1 ±0.1	0.3 ±0.1
BSoftmax	89.1 ±0.2	0.5 ±0.1	56.2 ±0.1	12.5 ±0.2	58.1 ±0.2	22.0 ±0.3	50.4 ±0.0	90.0 ±0.1	92.9 ±0.1	19.9 ±1.3
DFR	89.7 ±0.1	0.6 ±0.1	56.5 ±0.0	13.2 ±0.1	58.7 ±0.1	23.0 ±0.1	50.4 ±0.0	90.4 ±0.0	92.8 ±0.1	47.3 ±0.1
CRT	90.4 ±0.1	1.1 ±0.5	56.9 ±0.0	13.9 ±0.1	59.5 ±0.1	24.2 ±0.1	51.2 ±0.1	90.2 ±0.1	93.3 ±0.0	15.7 ±0.9
ReWeightCRT	89.9 ±0.1	1.4 ±0.6	56.6 ±0.0	13.4 ±0.1	59.0 ±0.1	23.3 ±0.1	51.1 ±0.3	90.4 ±0.0	93.1 ±0.1	15.5 ±0.7

CheXpert

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	88.6 ±0.7	41.7 ±3.4	68.3 ±1.1	42.3 ±2.4	68.7 ±0.2	43.8 ±0.7	69.5 ±1.3	70.0 ±1.5	85.4 ±0.4	5.0 ±1.1
Mixup	81.9 ±6.2	37.4 ±3.5	63.5 ±5.0	33.9 ±9.3	62.5 ±5.9	35.7 ±7.6	63.8 ±4.7	64.1 ±4.6	76.1 ±8.5	16.1 ±9.0
GroupDRO	79.0 ±0.3	75.1 ±0.6	62.9 ±0.1	28.5 ±0.3	64.6 ±0.2	42.0 ±0.3	78.6 ±0.0	79.3 ±0.1	86.3 ±0.1	22.0 ±0.4
CVaRDRO	73.7 ±1.0	50.2 ±1.8	57.3 ±0.1	20.1 ±0.3	56.8 ±0.4	29.9 ±0.2	65.7 ±0.6	67.0 ±0.4	72.9 ±0.4	40.4 ±0.0
JTT	75.2 ±0.8	60.4 ±4.8	59.4 ±1.1	23.0 ±1.5	59.6 ±1.4	34.4 ±2.3	70.7 ±2.6	72.0 ±2.5	79.0 ±2.5	24.4 ±0.5
LfF	22.3 ±10.2	13.7 ±9.8	37.3 ±5.8	9.0 ±0.7	19.5 ±8.3	8.8 ±3.9	46.2 ±2.9	46.2 ±3.1	30.5 ±10.1	65.7 ±10.2
LISA	81.9 ±6.2	37.4 ±3.5	63.5 ±5.0	33.9 ±9.3	62.5 ±5.9	35.7 ±7.6	63.8 ±4.7	64.1 ±4.6	76.1 ±8.5	16.1 ±9.0
ReSample	77.6 ±0.4	73.0 ±0.6	62.3 ±0.1	27.2 ±0.3	63.4 ±0.3	40.6 ±0.3	78.0 ±0.3	78.7 ±0.3	85.9 ±0.4	19.1 ±1.0
ReWeight	79.2 ±0.5	73.8 ±1.0	62.9 ±0.3	28.6 ±0.5	64.6 ±0.4	42.0 ±0.5	78.5 ±0.3	79.0 ±0.2	86.2 ±0.1	20.6 ±0.4
SqrtReWeight	83.5 ±0.3	68.5 ±1.6	65.0 ±0.2	33.3 ±0.4	67.9 ±0.2	45.5 ±0.3	77.8 ±0.4	78.5 ±0.3	86.3 ±0.3	15.6 ±1.1
CBLoss	80.0 ±0.5	74.0 ±0.7	63.3 ±0.2	29.4 ±0.5	65.2 ±0.4	42.6 ±0.5	78.6 ±0.2	79.0 ±0.2	86.1 ±0.3	19.6 ±0.6
Focal	89.3 ±0.3	42.1 ±4.0	69.6 ±0.4	44.7 ±1.1	69.8 ±0.4	45.5 ±1.0	70.4 ±1.1	70.4 ±1.3	86.5 ±0.1	16.1 ±1.7
LDAM	90.1 ±0.0	34.5 ±1.5	70.6 ±0.1	47.6 ±0.1	68.6 ±0.3	42.5 ±0.7	66.8 ±0.5	67.0 ±0.5	85.5 ±0.3	31.7 ±0.4
BSoftmax	79.5 ±0.2	74.2 ±1.1	63.2 ±0.1	29.0 ±0.2	65.0 ±0.2	42.5 ±0.2	78.7 ±0.2	79.6 ±0.1	86.6 ±0.1	21.9 ±0.2
DFR	78.9 ±0.2	75.4 ±0.6	62.9 ±0.1	28.5 ±0.1	64.5 ±0.1	42.0 ±0.1	78.9 ±0.1	79.3 ±0.0	86.0 ±0.1	26.2 ±0.5
CRT	79.1 ±0.1	74.0 ±0.2	62.9 ±0.0	28.6 ±0.1	64.6 ±0.1	42.0 ±0.1	78.7 ±0.1	79.1 ±0.1	86.2 ±0.0	21.9 ±0.1
ReWeightCRT	79.0 ±0.0	73.9 ±0.2	62.9 ±0.0	28.5 ±0.0	64.5 ±0.0	41.9 ±0.0	78.8 ±0.1	79.2 ±0.1	86.3 ±0.1	22.3 ±0.1

Living17

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	27.7 ±1.1	5.7 ±1.5	28.2 ±1.0	8.2 ±1.6	27.1 ±1.1	6.9 ±1.7	27.7 ±1.1	27.7 ±1.1	77.3 ±1.3	59.6 ±0.5
Mixup	29.8 ±1.8	8.7 ±1.4	30.9 ±2.2	9.5 ±1.3	29.3 ±1.9	9.7 ±1.5	29.8 ±1.8	29.8 ±1.8	78.2 ±1.2	34.1 ±1.9
GroupDRO	31.1 ±1.0	6.0 ±1.4	32.1 ±0.9	9.6 ±0.2	30.8 ±0.7	7.6 ±0.8	31.1 ±1.0	31.1 ±1.0	80.1 ±0.8	53.5 ±0.8
CVaRDRO	27.3 ±1.6	4.0 ±0.5	29.2 ±1.7	5.1 ±0.8	26.5 ±1.5	4.8 ±0.6	27.3 ±1.6	27.3 ±1.6	81.0 ±0.2	28.8 ±6.8
JTT	28.3 ±1.1	5.7 ±2.2	31.1 ±0.9	8.0 ±1.7	28.3 ±1.3	7.2 ±2.3	28.3 ±1.1	28.3 ±1.1	81.0 ±0.8	36.9 ±3.2
LfF	26.4 ±1.3	7.0 ±1.2	28.3 ±0.8	9.6 ±1.8	26.1 ±1.2	8.7 ±1.7	26.4 ±1.3	26.4 ±1.3	76.6 ±0.6	61.0 ±0.7
LISA	29.8 ±1.8	8.7 ±1.4	30.9 ±2.2	9.5 ±1.3	29.3 ±1.9	9.7 ±1.5	29.8 ±1.8	29.8 ±1.8	78.2 ±1.2	34.1 ±1.9
ReSample	31.4 ±0.6	6.7 ±1.5	33.0 ±0.6	11.0 ±0.6	31.0 ±0.6	8.3 ±1.2	31.4 ±0.6	31.4 ±0.6	81.0 ±0.7	46.6 ±3.1
ReWeight	27.7 ±1.1	5.7 ±1.5	28.2 ±1.0	8.2 ±1.6	27.1 ±1.1	6.9 ±1.7	27.7 ±1.1	27.7 ±1.1	77.3 ±1.3	59.6 ±0.5
SqrtReWeight	27.7 ±1.1	5.7 ±1.5	28.2 ±1.0	8.2 ±1.6	27.1 ±1.1	6.9 ±1.7	27.7 ±1.1	27.7 ±1.1	77.3 ±1.3	59.6 ±0.5
CBLoss	27.7 ±1.1	5.7 ±1.5	28.2 ±1.0	8.2 ±1.6	27.1 ±1.1	6.9 ±1.7	27.7 ±1.1	27.7 ±1.1	77.3 ±1.3	59.6 ±0.5
Focal	26.9 ±0.6	5.3 ±0.3	28.8 ±1.0	7.1 ±1.0	27.0 ±0.7	6.3 ±0.5	26.9 ±0.6	26.9 ±0.6	78.7 ±0.5	49.9 ±1.9
LDAM	24.3 ±0.8	4.0 ±0.8	28.0 ±1.2	6.6 ±1.4	24.0 ±0.8	5.1 ±0.3	24.3 ±0.8	24.3 ±0.8	79.1 ±1.0	12.4 ±0.5
BSoftmax	28.6 ±1.4	6.7 ±0.7	30.7 ±1.0	8.2 ±0.8	28.3 ±1.3	7.3 ±0.5	28.6 ±1.4	28.6 ±1.4	78.0 ±1.0	56.5 ±1.7
DFR	26.3 ±0.4	6.0 ±0.9	27.4 ±0.3	8.6 ±0.8	25.7 ±0.2	7.5 ±1.1	26.3 ±0.4	26.3 ±0.4	79.4 ±0.1	13.8 ±0.4
CRT	31.1 ±0.1	6.3 ±0.3	31.8 ±0.0	7.5 ±0.3	30.5 ±0.1	6.8 ±0.3	31.1 ±0.1	31.1 ±0.1	80.3 ±0.1	49.6 ±1.7
ReWeightCRT	33.1 ±0.1	9.3 ±0.3	33.4 ±0.1	11.3 ±0.3	32.6 ±0.0	10.8 ±0.2	33.1 ±0.1	33.1 ±0.1	82.0 ±0.0	40.0 ±0.4

Overall

Algorithm	Waterbirds	CelebA	CivilComments	MultiNLI	MetaShift	ImageNetBG	NICO++	MIMIC-CXR	MIMICNotes	CXRMultisite	CheXpert	Living17	Avg
ERM	69.1 ±4.7	57.6 ±0.8	63.2 ±1.2	69.5 ±0.3	82.1 ±0.8	76.8 ±0.9	40.0 ±0.0	68.5 ±0.4	80.4 ±0.2	50.1 ±0.9	41.7 ±3.4	27.7 ±1.1	60.5
Mixup	78.2 ±0.4	57.8 ±0.8	66.1 ±1.3	68.5 ±0.6	79.0 ±0.8	76.9 ±0.7	30.0 ±4.1	67.2 ±1.1	81.6 ±0.6	50.1 ±0.9	37.4 ±3.5	29.8 ±1.8	60.2
GroupDRO	73.1 ±0.4	78.5 ±1.1	69.5 ±0.7	69.3 ±1.5	82.6 ±1.1	76.4 ±0.2	31.1 ±0.9	67.4 ±0.9	83.7 ±0.0	49.2 ±0.5	75.1 ±0.6	31.1 ±1.0	65.6
CVaRDRO	75.5 ±2.2	62.2 ±3.1	68.7 ±1.3	63.0 ±1.5	82.6 ±1.1	74.8 ±0.8	31.7 ±3.6	67.5 ±0.1	65.6 ±1.5	50.2 ±0.9	50.2 ±1.8	27.3 ±1.6	59.9
JTT	71.0 ±0.5	66.0 ±11.9	64.3 ±1.5	68.4 ±0.6	82.6 ±0.4	77.0 ±0.4	32.2 ±0.9	66.6 ±0.8	83.8 ±0.1	50.1 ±0.9	60.4 ±4.8	28.3 ±1.1	62.6
LfF	74.7 ±1.0	53.0 ±4.3	50.3 ±5.9	63.6 ±2.9	72.6 ±1.2	70.1 ±1.4	28.3 ±1.7	62.1 ±2.4	84.1 ±0.0	50.1 ±0.9	13.7 ±9.8	26.4 ±1.3	54.1
LISA	78.2 ±0.4	57.8 ±0.8	66.1 ±1.3	68.5 ±0.6	79.0 ±0.8	76.9 ±0.7	30.0 ±4.1	67.2 ±1.1	81.6 ±0.6	50.1 ±0.9	37.4 ±3.5	29.8 ±1.8	60.2
ReSample	70.0 ±1.0	82.2 ±1.2	68.2 ±0.7	67.5 ±0.4	80.5 ±1.5	77.7 ±1.1	23.3 ±1.4	68.9 ±0.3	82.4 ±0.5	47.8 ±2.5	73.0 ±0.6	31.4 ±0.6	64.4
ReWeight	72.5 ±0.3	81.5 ±0.9	69.9 ±0.6	67.8 ±1.2	83.1 ±0.7	76.8 ±0.9	25.0 ±0.0	67.4 ±0.3	84.0 ±0.1	51.9 ±2.3	73.8 ±1.0	27.7 ±1.1	65.1
SqrtReWeight	71.3 ±1.4	72.0 ±2.2	70.1 ±0.3	66.6 ±0.4	82.1 ±0.8	76.8 ±0.9	35.6 ±1.8	68.9 ±0.5	83.1 ±0.2	50.2 ±0.9	68.5 ±1.6	27.7 ±1.1	64.4
CBLoss	74.4 ±1.2	75.0 ±2.4	67.0 ±0.1	66.2 ±0.7	82.6 ±0.4	76.8 ±0.9	34.4 ±2.4	67.8 ±0.6	83.9 ±0.1	50.2 ±0.9	74.0 ±0.7	27.7 ±1.1	65.0
Focal	71.6 ±0.8	59.1 ±2.0	61.9 ±1.1	69.3 ±0.8	81.0 ±0.4	71.9 ±1.2	29.4 ±2.0	67.3 ±1.2	70.3 ±9.6	50.0 ±0.9	42.1 ±4.0	26.9 ±0.6	58.4
LDAM	71.0 ±1.8	59.3 ±2.3	37.0 ±7.9	69.6 ±1.6	83.6 ±0.4	76.7 ±0.5	26.7 ±1.4	68.2 ±1.4	81.0 ±0.3	50.1 ±0.9	34.5 ±1.5	24.3 ±0.8	56.8
BSoftmax	74.1 ±0.9	83.3 ±0.5	69.4 ±1.2	66.9 ±0.4	82.6 ±0.4	76.1 ±2.0	29.4 ±2.0	67.2 ±0.2	83.7 ±0.3	50.1 ±1.1	74.2 ±1.1	28.6 ±1.4	65.5
DFR	89.0 ±0.2	86.3 ±0.3	66.5 ±0.2	63.8 ±0.0	81.5 ±0.0	74.4 ±1.8	39.3 ±2.4	67.1 ±0.4	80.2 ±0.0	56.1 ±3.5	75.4 ±0.6	26.3 ±0.4	67.2
CRT	76.3 ±0.8	70.4 ±0.4	68.5 ±0.0	65.4 ±0.1	83.1 ±0.0	78.2 ±0.5	33.3 ±0.0	69.1 ±0.2	83.4 ±0.0	56.5 ±3.5	74.0 ±0.2	31.1 ±0.1	65.8
ReWeightCRT	76.3 ±0.2	71.1 ±0.5	68.2 ±0.4	65.3 ±0.1	85.1 ±0.4	77.5 ±0.7	33.3 ±0.0	68.9 ±0.0	83.4 ±0.0	55.4 ±4.9	73.9 ±0.2	33.1 ±0.1	66.0

■ D.4.3 Attributes Unknown in Both Training & Validation

Waterbirds

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	84.1 ±1.7	69.1 ±4.7	77.4 ±2.0	60.7 ±3.2	79.4 ±2.1	69.5 ±2.9	83.1 ±2.0	83.1 ±2.0	91.0 ±1.4	12.9 ±1.7
Mixup	89.2 ±0.6	77.5 ±0.7	83.5 ±0.8	70.6 ±1.9	85.6 ±0.7	78.4 ±0.9	88.9 ±0.2	88.9 ±0.2	94.9 ±0.1	8.0 ±1.1
GroupDRO	86.9 ±0.9	73.1 ±0.4	80.7 ±1.1	66.1 ±2.2	82.8 ±0.9	74.4 ±1.2	86.3 ±0.5	86.3 ±0.5	94.0 ±0.3	10.5 ±0.8
CVaRDRO	89.9 ±0.4	75.5 ±2.2	84.5 ±0.7	73.2 ±1.7	86.2 ±0.3	79.0 ±0.4	88.5 ±0.3	88.5 ±0.3	95.4 ±0.2	8.3 ±0.2
JTT	88.9 ±0.6	71.2 ±0.5	83.2 ±0.8	71.4 ±1.6	84.7 ±0.6	76.8 ±0.8	86.8 ±0.2	86.8 ±0.2	94.2 ±0.1	9.2 ±0.3
LfF	86.6 ±0.5	75.0 ±0.7	80.3 ±0.6	65.1 ±1.1	82.5 ±0.5	74.0 ±0.7	86.3 ±0.3	86.3 ±0.3	93.4 ±0.2	10.0 ±0.8
LISA	89.2 ±0.6	77.5 ±0.7	83.5 ±0.8	70.6 ±1.9	85.6 ±0.7	78.4 ±0.9	88.9 ±0.2	88.9 ±0.2	94.9 ±0.1	8.0 ±1.1
ReSample	86.2 ±0.5	70.0 ±1.0	79.8 ±0.6	64.9 ±1.4	81.7 ±0.5	72.7 ±0.7	85.0 ±0.2	85.0 ±0.2	92.8 ±0.1	11.3 ±0.3
ReWeight	86.2 ±0.6	71.9 ±0.6	79.9 ±0.7	64.3 ±1.6	82.1 ±0.6	73.5 ±0.7	86.2 ±0.1	86.2 ±0.1	94.0 ±0.1	10.8 ±0.4
SqrtReWeight	89.4 ±0.4	71.0 ±1.4	83.9 ±0.6	72.8 ±0.9	85.3 ±0.6	77.7 ±0.9	87.2 ±0.6	87.2 ±0.6	94.4 ±0.5	9.0 ±0.5
CBLoss	86.8 ±0.6	74.4 ±1.2	80.4 ±0.7	65.5 ±1.3	82.6 ±0.7	74.0 ±1.0	86.2 ±0.6	86.2 ±0.6	93.5 ±0.4	11.3 ±0.4
Focal	89.3 ±0.2	71.6 ±0.8	83.7 ±0.3	72.4 ±0.5	85.2 ±0.3	77.5 ±0.4	87.1 ±0.3	87.1 ±0.3	94.2 ±0.2	6.9 ±0.1
LDAM	87.9 ±0.2	70.9 ±1.7	81.9 ±0.3	69.1 ±0.8	83.6 ±0.1	75.2 ±0.1	86.0 ±0.2	86.0 ±0.2	93.5 ±0.1	11.8 ±1.7
BSoftmax	88.4 ±1.2	74.1 ±0.9	82.6 ±1.6	69.9 ±2.9	84.4 ±1.5	76.4 ±2.0	87.0 ±1.0	87.0 ±1.0	94.0 ±0.9	9.9 ±1.2
DFR	92.2 ±0.2	89.0 ±0.2	87.6 ±0.3	78.3 ±0.6	89.2 ±0.2	83.5 ±0.4	91.2 ±0.1	91.2 ±0.1	96.8 ±0.0	6.9 ±0.4
CRT	89.2 ±0.1	76.3 ±0.8	83.5 ±0.1	71.3 ±0.4	85.3 ±0.1	77.8 ±0.1	87.9 ±0.1	87.9 ±0.1	94.8 ±0.0	9.2 ±0.2
ReWeightCRT	89.4 ±0.3	76.3 ±0.2	83.8 ±0.3	71.9 ±0.7	85.6 ±0.3	78.1 ±0.4	88.0 ±0.2	88.0 ±0.2	94.9 ±0.1	8.8 ±0.2

CelebA

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	95.0 ±0.1	57.6 ±0.8	87.4 ±0.2	76.3 ±0.3	89.9 ±0.1	82.7 ±0.2	85.6 ±0.2	93.0 ±0.0	98.4 ±0.0	2.9 ±0.1
Mixup	95.4 ±0.1	57.8 ±0.8	88.4 ±0.3	78.5 ±0.7	90.6 ±0.2	83.8 ±0.3	85.8 ±0.2	93.1 ±0.1	98.4 ±0.1	2.5 ±0.2
GroupDRO	94.4 ±0.0	68.3 ±0.9	85.8 ±0.1	72.7 ±0.1	89.2 ±0.1	81.7 ±0.1	88.4 ±0.2	93.9 ±0.1	98.6 ±0.0	4.1 ±0.0
CVaRDRO	95.1 ±0.1	60.2 ±3.0	87.7 ±0.3	76.9 ±0.6	90.1 ±0.1	83.1 ±0.2	86.3 ±0.7	93.1 ±0.1	98.4 ±0.0	3.1 ±0.1
JTT	95.9 ±0.0	48.3 ±1.5	90.5 ±0.1	82.9 ±0.3	91.4 ±0.1	85.2 ±0.1	83.4 ±0.4	92.4 ±0.2	98.6 ±0.0	1.3 ±0.1
LfF	81.1 ±5.6	53.0 ±4.3	71.8 ±4.1	45.2 ±8.3	73.2 ±5.6	59.0 ±7.3	78.3 ±3.0	85.3 ±2.9	94.1 ±1.2	27.9 ±5.5
LISA	95.4 ±0.1	57.8 ±0.8	88.4 ±0.3	78.5 ±0.7	90.6 ±0.2	83.8 ±0.3	85.8 ±0.2	93.1 ±0.1	98.4 ±0.1	2.5 ±0.2
ReSample	94.1 ±0.1	74.1 ±2.2	85.0 ±0.1	71.1 ±0.2	88.6 ±0.1	80.7 ±0.2	89.5 ±0.5	93.8 ±0.1	98.4 ±0.0	4.8 ±0.1
ReWeight	94.1 ±0.1	69.6 ±0.2	85.1 ±0.1	71.2 ±0.2	88.7 ±0.1	81.0 ±0.2	88.6 ±0.0	94.0 ±0.1	98.5 ±0.0	4.6 ±0.0
SqrtReWeight	94.0 ±0.2	66.9 ±2.2	85.0 ±0.4	71.0 ±0.9	88.6 ±0.3	80.7 ±0.4	87.9 ±0.5	93.9 ±0.1	98.4 ±0.0	4.8 ±0.2
CBLoss	94.4 ±0.0	65.4 ±1.4	85.9 ±0.1	72.9 ±0.2	89.2 ±0.1	81.7 ±0.1	87.6 ±0.3	93.8 ±0.1	98.5 ±0.0	4.4 ±0.1
Focal	94.9 ±0.3	56.9 ±3.4	87.4 ±0.7	76.4 ±1.5	89.7 ±0.4	82.4 ±0.7	85.2 ±0.8	92.6 ±0.3	98.3 ±0.1	3.1 ±0.4
LDAM	94.7 ±0.3	57.0 ±4.1	86.7 ±0.9	74.8 ±2.0	89.5 ±0.5	82.1 ±0.8	85.5 ±0.9	93.2 ±0.2	98.4 ±0.0	30.7 ±0.5
BSoftmax	94.5 ±0.1	69.6 ±1.2	85.9 ±0.2	72.9 ±0.4	89.4 ±0.2	82.0 ±0.3	88.8 ±0.3	94.2 ±0.1	98.6 ±0.0	4.6 ±0.0
DFR	93.6 ±0.0	73.7 ±0.8	84.1 ±0.1	69.4 ±0.2	87.8 ±0.1	79.4 ±0.1	89.0 ±0.2	93.2 ±0.0	98.2 ±0.0	14.8 ±0.5
CRT	94.1 ±0.1	69.6 ±0.7	85.1 ±0.3	71.4 ±0.5	88.6 ±0.2	80.7 ±0.3	88.4 ±0.1	93.6 ±0.0	98.4 ±0.0	4.6 ±0.2
ReWeightCRT	94.2 ±0.1	70.7 ±0.6	85.4 ±0.1	71.9 ±0.3	88.8 ±0.1	81.1 ±0.2	88.7 ±0.1	93.6 ±0.0	98.4 ±0.0	4.7 ±0.1

CivilComments

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	85.4 ±0.2	63.2 ±1.2	75.4 ±0.3	57.6 ±0.7	77.0 ±0.1	63.2 ±0.1	77.7 ±0.1	79.4 ±0.2	89.8 ±0.1	7.8 ±0.4
Mixup	84.6 ±0.1	65.8 ±1.5	74.4 ±0.2	55.3 ±0.3	76.4 ±0.1	62.6 ±0.2	78.0 ±0.2	79.7 ±0.0	89.7 ±0.1	9.3 ±0.4
GroupDRO	81.2 ±0.3	61.5 ±1.8	71.9 ±0.2	48.6 ±0.5	74.2 ±0.3	60.9 ±0.3	78.9 ±0.1	81.3 ±0.1	89.8 ±0.1	15.7 ±0.6
CVaRDRO	81.6 ±0.7	62.9 ±3.8	72.1 ±0.5	49.3 ±1.2	74.4 ±0.6	60.9 ±0.6	78.4 ±0.4	80.8 ±0.1	89.6 ±0.1	31.9 ±0.1
JTT	79.0 ±1.8	51.0 ±4.2	69.7 ±1.3	45.5 ±2.8	71.4 ±1.6	56.6 ±1.8	75.0 ±0.8	77.7 ±0.8	86.5 ±1.0	14.0 ±1.6
LfF	69.1 ±4.3	42.2 ±7.2	62.6 ±3.2	33.9 ±4.5	62.0 ±4.3	45.7 ±5.2	67.2 ±4.0	69.7 ±4.7	75.0 ±6.6	27.9 ±1.6
LISA	84.6 ±0.1	65.8 ±1.5	74.4 ±0.2	55.3 ±0.3	76.4 ±0.1	62.6 ±0.2	78.0 ±0.2	79.7 ±0.0	89.7 ±0.1	9.3 ±0.4
ReSample	80.4 ±0.2	61.0 ±0.6	71.2 ±0.1	47.2 ±0.2	73.4 ±0.1	59.8 ±0.2	78.3 ±0.1	80.7 ±0.1	89.3 ±0.1	17.0 ±0.6
ReWeight	80.6 ±0.3	59.3 ±1.1	71.5 ±0.2	47.6 ±0.5	73.7 ±0.2	60.3 ±0.2	78.7 ±0.1	81.3 ±0.0	89.9 ±0.1	14.9 ±0.5
SqrtReWeight	82.9 ±0.5	68.6 ±1.1	73.0 ±0.4	51.7 ±0.9	75.4 ±0.4	61.8 ±0.4	78.6 ±0.2	80.6 ±0.2	89.8 ±0.1	10.9 ±0.7
CBLoss	84.0 ±0.8	67.3 ±0.2	74.0 ±0.7	54.1 ±1.8	76.2 ±0.6	62.5 ±0.5	78.5 ±0.2	80.3 ±0.3	90.0 ±0.0	9.5 ±1.4
Focal	85.6 ±0.3	61.9 ±1.1	75.6 ±0.4	58.5 ±0.9	77.0 ±0.3	62.9 ±0.5	77.3 ±0.3	78.7 ±0.3	89.4 ±0.4	7.7 ±0.4
LDAM	80.2 ±2.1	28.4 ±7.7	67.8 ±3.3	46.2 ±5.6	68.3 ±3.1	48.9 ±5.0	66.1 ±3.6	69.5 ±3.2	77.4 ±4.0	20.7 ±0.5
BSoftmax	80.3 ±0.2	58.3 ±1.1	71.3 ±0.1	47.2 ±0.3	73.5 ±0.2	60.0 ±0.1	78.4 ±0.1	81.1 ±0.1	89.8 ±0.1	16.5 ±0.8
DFR	80.7 ±0.0	64.4 ±0.1	70.9 ±0.0	47.6 ±0.1	73.1 ±0.0	58.7 ±0.0	76.8 ±0.0	79.0 ±0.0	86.9 ±0.0	20.4 ±0.1
CRT	82.7 ±0.1	67.8 ±0.3	72.8 ±0.1	51.1 ±0.2	75.2 ±0.1	61.6 ±0.1	78.7 ±0.0	80.7 ±0.0	89.5 ±0.1	13.0 ±0.1
ReWeightCRT	82.4 ±0.0	64.7 ±0.2	72.6 ±0.0	50.5 ±0.1	75.0 ±0.0	61.4 ±0.0	78.4 ±0.0	80.7 ±0.0	89.5 ±0.0	12.6 ±0.1

MultiNLI

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	81.0 ±0.3	66.4 ±2.3	81.3 ±0.3	74.2 ±0.4	81.1 ±0.3	77.6 ±0.2	79.4 ±0.8	81.0 ±0.3	93.6 ±0.1	11.1 ±0.6
Mixup	81.7 ±0.1	66.8 ±0.3	81.9 ±0.1	75.8 ±0.1	81.8 ±0.1	78.3 ±0.1	80.1 ±0.2	81.7 ±0.1	93.7 ±0.0	10.9 ±0.2
GroupDRO	81.1 ±0.2	64.1 ±0.8	81.3 ±0.2	74.8 ±0.1	81.1 ±0.2	77.9 ±0.1	79.4 ±0.2	81.1 ±0.2	93.7 ±0.1	9.1 ±0.9
CVaRDRO	75.4 ±0.2	48.2 ±3.4	75.8 ±0.3	68.9 ±0.1	75.5 ±0.3	72.0 ±0.1	72.3 ±0.8	75.4 ±0.2	87.3 ±0.2	41.8 ±0.2
JTT	81.4 ±0.0	65.1 ±1.6	81.7 ±0.0	75.0 ±0.1	81.5 ±0.0	77.8 ±0.1	79.8 ±0.3	81.4 ±0.0	93.9 ±0.0	9.4 ±0.4
LfF	71.4 ±1.6	57.3 ±5.7	71.5 ±1.7	67.1 ±2.0	71.4 ±1.7	68.4 ±2.7	69.3 ±2.9	71.4 ±1.6	86.6 ±1.4	6.2 ±0.5
LISA	81.7 ±0.1	66.8 ±0.3	81.9 ±0.1	75.8 ±0.1	81.8 ±0.1	78.3 ±0.1	80.1 ±0.2	81.7 ±0.1	93.7 ±0.0	10.9 ±0.2
ReSample	81.5 ±0.0	66.8 ±0.5	81.9 ±0.1	74.2 ±0.2	81.6 ±0.0	78.0 ±0.1	79.9 ±0.1	81.5 ±0.0	93.9 ±0.0	12.2 ±0.6
ReWeight	79.4 ±0.2	64.2 ±1.9	79.6 ±0.2	72.9 ±0.3	79.4 ±0.2	75.8 ±0.1	78.0 ±0.2	79.4 ±0.2	92.6 ±0.1	14.2 ±0.6
SqrtReWeight	80.6 ±0.2	63.8 ±2.4	80.8 ±0.2	75.1 ±0.2	80.6 ±0.2	77.5 ±0.2	78.8 ±0.5	80.6 ±0.2	93.5 ±0.1	7.8 ±0.2
CBLoss	80.6 ±0.3	63.6 ±2.4	80.8 ±0.2	75.1 ±0.3	80.6 ±0.3	77.5 ±0.2	78.7 ±0.5	80.6 ±0.3	93.5 ±0.1	7.8 ±0.2
Focal	80.9 ±0.2	62.4 ±2.0	81.2 ±0.2	74.3 ±0.1	81.0 ±0.2	77.4 ±0.2	78.7 ±0.3	80.9 ±0.2	93.7 ±0.1	5.3 ±0.8
LDAM	80.9 ±0.1	65.5 ±0.8	81.1 ±0.1	74.6 ±0.3	80.9 ±0.1	77.4 ±0.0	79.2 ±0.2	80.9 ±0.1	93.5 ±0.0	33.2 ±0.4
BSoftmax	80.6 ±0.2	63.6 ±2.4	80.8 ±0.2	75.1 ±0.2	80.7 ±0.2	77.6 ±0.2	78.7 ±0.5	80.6 ±0.2	93.5 ±0.1	7.8 ±0.2
DFR	80.2 ±0.0	63.8 ±0.0	80.3 ±0.0	75.2 ±0.0	80.3 ±0.0	76.2 ±0.0	78.5 ±0.0	80.2 ±0.0	92.9 ±0.0	5.8 ±0.0
CRT	80.2 ±0.0	65.4 ±0.2	80.3 ±0.0	74.3 ±0.0	80.2 ±0.0	76.4 ±0.0	78.6 ±0.0	80.2 ±0.0	92.8 ±0.0	14.9 ±0.1
ReWeightCRT	80.2 ±0.0	65.2 ±0.2	80.3 ±0.0	74.4 ±0.0	80.2 ±0.0	76.4 ±0.0	78.6 ±0.0	80.2 ±0.0	92.9 ±0.0	14.7 ±0.1

MetaShift

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	91.2 ±0.4	82.1 ±0.8	91.1 ±0.4	90.4 ±0.5	91.1 ±0.4	90.6 ±0.4	89.1 ±0.5	91.1 ±0.4	97.4 ±0.1	6.1 ±0.7
Mixup	91.4 ±0.1	79.0 ±0.8	91.4 ±0.1	90.6 ±0.4	91.4 ±0.1	90.9 ±0.1	88.5 ±0.2	91.4 ±0.1	97.3 ±0.0	1.9 ±0.4
GroupDRO	91.5 ±0.3	83.1 ±0.7	91.4 ±0.3	90.1 ±0.6	91.4 ±0.3	91.0 ±0.3	89.4 ±0.5	91.5 ±0.3	97.6 ±0.1	5.5 ±1.0
CVaRDRO	91.2 ±1.0	83.5 ±0.5	91.2 ±1.0	89.4 ±1.8	91.1 ±1.0	90.7 ±0.9	89.2 ±1.0	91.2 ±0.9	97.5 ±0.2	13.0 ±5.9
JTT	91.2 ±0.1	82.6 ±0.4	91.1 ±0.1	90.6 ±0.2	91.1 ±0.1	90.6 ±0.1	89.2 ±0.1	91.1 ±0.1	97.6 ±0.0	7.2 ±0.2
LfF	80.4 ±0.4	72.3 ±1.3	80.7 ±0.3	76.9 ±1.7	80.4 ±0.4	79.7 ±0.4	80.5 ±0.6	80.6 ±0.3	91.5 ±0.1	8.5 ±1.1
LISA	91.4 ±0.1	79.0 ±0.8	91.4 ±0.1	90.6 ±0.4	91.4 ±0.1	90.9 ±0.1	88.5 ±0.2	91.4 ±0.1	97.3 ±0.0	1.9 ±0.4
ReSample	92.2 ±0.3	81.0 ±1.7	92.1 ±0.3	91.4 ±0.4	92.2 ±0.3	91.7 ±0.3	89.6 ±0.1	92.2 ±0.2	97.5 ±0.1	6.8 ±0.4
ReWeight	91.5 ±0.4	83.1 ±0.7	91.5 ±0.4	90.6 ±0.3	91.5 ±0.4	91.0 ±0.4	89.5 ±0.4	91.5 ±0.4	97.5 ±0.1	5.8 ±0.6
SqrtReWeight	91.3 ±0.1	82.6 ±0.4	91.2 ±0.1	90.3 ±0.2	91.2 ±0.1	90.7 ±0.2	89.2 ±0.2	91.3 ±0.1	97.5 ±0.1	5.6 ±1.0
CBLoss	91.4 ±0.1	83.1 ±0.0	91.3 ±0.1	90.4 ±0.4	91.4 ±0.1	90.9 ±0.2	89.4 ±0.1	91.4 ±0.2	97.4 ±0.1	6.3 ±0.4
Focal	91.6 ±0.2	81.0 ±0.4	91.7 ±0.2	90.9 ±0.6	91.6 ±0.2	91.1 ±0.2	89.4 ±0.2	91.6 ±0.2	97.6 ±0.0	4.9 ±1.2
LDAM	91.6 ±0.1	83.6 ±0.4	91.6 ±0.0	90.9 ±0.3	91.6 ±0.1	91.1 ±0.1	89.9 ±0.1	91.6 ±0.1	97.5 ±0.1	9.5 ±1.0
BSoftmax	91.3 ±0.3	82.6 ±0.4	91.3 ±0.3	89.9 ±0.3	91.3 ±0.3	90.9 ±0.3	89.2 ±0.3	91.4 ±0.3	97.5 ±0.1	5.7 ±0.8
DFR	90.2 ±0.2	81.4 ±0.1	90.2 ±0.2	88.1 ±0.5	90.2 ±0.2	89.8 ±0.2	88.0 ±0.2	90.3 ±0.2	96.7 ±0.0	3.2 ±0.2
CRT	91.5 ±0.0	83.1 ±0.0	91.4 ±0.0	90.6 ±0.1	91.4 ±0.0	90.9 ±0.0	89.5 ±0.0	91.4 ±0.0	97.3 ±0.0	6.8 ±0.0
ReWeightCRT	91.3 ±0.1	85.1 ±0.4	91.2 ±0.1	90.1 ±0.3	91.2 ±0.1	90.8 ±0.1	89.7 ±0.2	91.3 ±0.1	96.8 ±0.1	8.1 ±0.1

ImageNetBG

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	87.7 ±0.1	76.8 ±0.9	87.9 ±0.1	78.3 ±1.3	87.7 ±0.1	80.7 ±0.4	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
Mixup	87.9 ±0.2	76.9 ±0.7	88.4 ±0.1	76.6 ±2.6	88.0 ±0.2	80.5 ±1.0	87.9 ±0.2	87.9 ±0.2	98.7 ±0.0	4.7 ±1.6
GroupDRO	87.7 ±0.1	76.4 ±0.2	87.9 ±0.1	76.2 ±0.5	87.6 ±0.1	81.1 ±0.3	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
CVaRDRO	87.8 ±0.2	74.8 ±0.8	88.0 ±0.1	79.9 ±0.9	87.8 ±0.2	79.5 ±0.3	87.8 ±0.2	87.8 ±0.2	99.0 ±0.0	5.6 ±0.2
JTT	87.6 ±0.4	77.0 ±0.4	87.8 ±0.3	78.3 ±3.0	87.5 ±0.3	80.4 ±0.6	87.6 ±0.4	87.6 ±0.4	99.0 ±0.0	3.7 ±0.2
LfF	84.7 ±0.5	70.1 ±1.4	85.4 ±0.3	72.1 ±3.1	84.7 ±0.5	76.2 ±0.6	84.7 ±0.5	84.7 ±0.5	98.6 ±0.0	1.8 ±0.4
LISA	87.9 ±0.2	76.9 ±0.7	88.4 ±0.1	76.6 ±2.6	88.0 ±0.2	80.5 ±1.0	87.9 ±0.2	87.9 ±0.2	98.7 ±0.0	4.7 ±1.6
ReSample	88.2 ±0.4	77.7 ±1.1	88.4 ±0.4	79.7 ±1.0	88.2 ±0.4	80.6 ±1.0	88.2 ±0.4	88.2 ±0.4	99.0 ±0.0	5.4 ±0.5
ReWeight	87.7 ±0.1	76.8 ±0.9	87.9 ±0.1	78.3 ±1.3	87.7 ±0.1	80.7 ±0.4	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
SqrtReWeight	87.7 ±0.1	76.8 ±0.9	87.9 ±0.1	78.3 ±1.3	87.7 ±0.1	80.7 ±0.4	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
CBLoss	87.7 ±0.1	76.8 ±0.9	87.9 ±0.1	78.3 ±1.3	87.7 ±0.1	80.7 ±0.4	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.1 ±0.1
Focal	86.7 ±0.2	71.9 ±1.2	87.1 ±0.2	74.2 ±1.3	86.6 ±0.2	77.6 ±0.7	86.7 ±0.2	86.7 ±0.2	98.9 ±0.0	2.8 ±0.6
LDAM	88.2 ±0.1	76.7 ±0.5	88.5 ±0.1	77.6 ±1.1	88.1 ±0.1	81.3 ±0.4	88.2 ±0.1	88.2 ±0.1	98.8 ±0.0	45.9 ±0.7
BSoftmax	87.7 ±0.1	76.1 ±2.0	88.0 ±0.1	77.7 ±1.3	87.7 ±0.1	80.4 ±0.8	87.7 ±0.1	87.7 ±0.1	99.0 ±0.0	5.6 ±0.5
DFR	86.8 ±0.5	74.4 ±1.8	86.9 ±0.5	78.9 ±1.6	86.7 ±0.5	78.1 ±1.3	86.8 ±0.5	86.8 ±0.5	98.8 ±0.1	8.9 ±1.4
CRT	88.3 ±0.1	78.2 ±0.5	88.3 ±0.1	82.7 ±0.4	88.3 ±0.1	80.9 ±0.2	88.3 ±0.1	88.3 ±0.1	99.1 ±0.0	5.6 ±0.2
ReWeightCRT	88.4 ±0.1	77.5 ±0.7	88.5 ±0.1	82.1 ±0.4	88.4 ±0.1	81.2 ±0.3	88.4 ±0.1	88.4 ±0.1	99.1 ±0.0	5.4 ±0.2

NICO++

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	85.3 ±0.3	35.0 ±4.1	86.3 ±0.2	56.3 ±4.9	85.4 ±0.3	64.9 ±2.3	84.9 ±0.3	85.2 ±0.3	99.4 ±0.0	9.4 ±0.1
Mixup	85.4 ±0.1	30.0 ±4.1	86.2 ±0.1	62.2 ±0.5	85.5 ±0.1	69.4 ±0.4	85.0 ±0.1	85.3 ±0.1	99.2 ±0.0	2.3 ±0.5
GroupDRO	83.7 ±0.5	31.1 ±0.9	84.6 ±0.4	58.2 ±1.5	83.8 ±0.5	66.7 ±0.9	83.3 ±0.4	83.6 ±0.5	99.3 ±0.0	7.0 ±0.2
CVaRDRO	85.8 ±0.1	27.8 ±2.3	86.5 ±0.1	61.5 ±0.7	85.8 ±0.1	67.3 ±0.5	85.2 ±0.1	85.7 ±0.1	99.4 ±0.0	9.9 ±0.0
JTT	85.7 ±0.1	30.6 ±2.3	86.4 ±0.1	60.9 ±1.2	85.8 ±0.1	66.7 ±0.4	85.2 ±0.1	85.6 ±0.1	99.4 ±0.0	9.5 ±0.2
LfF	78.7 ±0.6	28.8 ±2.0	81.0 ±0.3	45.1 ±1.2	79.0 ±0.6	54.2 ±1.4	78.4 ±0.5	78.6 ±0.6	99.2 ±0.0	1.5 ±0.3
LISA	85.4 ±0.1	30.0 ±4.1	86.2 ±0.1	62.2 ±0.5	85.5 ±0.1	69.4 ±0.4	85.0 ±0.1	85.3 ±0.1	99.2 ±0.0	2.3 ±0.5
ReSample	84.9 ±0.2	30.6 ±2.3	85.6 ±0.1	62.9 ±1.6	84.9 ±0.2	67.1 ±0.3	84.3 ±0.2	84.8 ±0.2	99.3 ±0.0	10.2 ±0.4
ReWeight	85.5 ±0.2	25.0 ±0.0	86.4 ±0.1	59.0 ±1.3	85.6 ±0.2	67.6 ±0.4	84.9 ±0.2	85.4 ±0.2	99.4 ±0.0	9.7 ±0.0
SqrtReWeight	85.5 ±0.1	32.8 ±3.5	86.5 ±0.1	55.9 ±2.1	85.6 ±0.1	66.4 ±1.3	85.0 ±0.1	85.4 ±0.1	99.4 ±0.0	9.4 ±0.1
CBLoss	85.9 ±0.1	31.7 ±3.6	86.6 ±0.1	59.3 ±3.0	86.0 ±0.1	67.2 ±0.7	85.4 ±0.1	85.8 ±0.0	99.4 ±0.0	10.1 ±0.1
Focal	85.7 ±0.1	30.6 ±2.3	86.5 ±0.1	58.5 ±0.6	85.8 ±0.1	66.7 ±0.8	85.2 ±0.1	85.6 ±0.1	99.5 ±0.0	6.3 ±0.3
LDAM	85.4 ±0.4	31.7 ±3.6	86.1 ±0.3	62.4 ±1.0	85.5 ±0.4	68.1 ±1.3	84.9 ±0.5	85.3 ±0.4	99.1 ±0.0	56.9 ±1.4
BSoftmax	85.8 ±0.0	35.6 ±1.8	86.4 ±0.1	60.7 ±1.4	85.8 ±0.0	69.2 ±0.6	85.3 ±0.0	85.7 ±0.0	99.4 ±0.0	9.4 ±0.1
DFR	82.7 ±0.1	38.0 ±3.8	83.2 ±0.1	58.5 ±1.6	82.7 ±0.1	65.1 ±0.2	82.4 ±0.1	82.6 ±0.1	99.2 ±0.0	11.7 ±0.2
CRT	85.8 ±0.1	33.3 ±0.0	86.1 ±0.0	65.3 ±0.6	85.8 ±0.1	69.9 ±0.4	85.3 ±0.1	85.6 ±0.1	99.4 ±0.0	6.0 ±0.6
ReWeightCRT	85.8 ±0.1	33.3 ±0.0	86.1 ±0.1	64.3 ±0.7	85.8 ±0.1	69.7 ±0.4	85.4 ±0.1	85.7 ±0.1	99.4 ±0.0	6.1 ±0.8

MIMIC-CXR

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	78.3 ±0.1	68.6 ±0.2	77.4 ±0.1	71.2 ±0.1	77.6 ±0.1	73.7 ±0.1	77.3 ±0.1	78.0 ±0.1	85.4 ±0.1	3.8 ±0.2
Mixup	77.9 ±0.3	66.8 ±0.6	77.0 ±0.3	70.7 ±0.6	77.2 ±0.3	73.2 ±0.3	76.9 ±0.3	77.5 ±0.3	84.9 ±0.3	3.6 ±0.5
GroupDRO	77.9 ±0.0	67.4 ±0.5	77.1 ±0.0	70.2 ±0.1	77.4 ±0.0	73.7 ±0.0	77.3 ±0.0	77.9 ±0.0	85.2 ±0.1	5.5 ±0.3
CVaRDRO	78.2 ±0.1	68.0 ±0.2	77.3 ±0.1	70.7 ±0.2	77.6 ±0.1	73.8 ±0.1	77.3 ±0.1	78.0 ±0.1	85.1 ±0.0	6.8 ±0.7
JTT	77.4 ±0.3	64.9 ±0.3	76.5 ±0.3	70.1 ±0.7	76.7 ±0.2	72.8 ±0.2	76.4 ±0.2	77.1 ±0.2	84.5 ±0.2	4.2 ±0.4
LfF	73.2 ±0.9	62.2 ±2.4	72.3 ±1.0	65.1 ±0.9	72.5 ±1.0	67.8 ±1.5	72.4 ±1.1	72.9 ±1.1	79.3 ±1.3	11.3 ±0.8
LISA	77.9 ±0.3	66.8 ±0.6	77.0 ±0.3	70.7 ±0.6	77.2 ±0.3	73.2 ±0.3	76.9 ±0.3	77.5 ±0.3	84.9 ±0.3	3.6 ±0.5
ReSample	78.4 ±0.1	67.5 ±0.3	77.6 ±0.1	70.8 ±0.1	77.8 ±0.1	74.2 ±0.1	77.6 ±0.1	78.3 ±0.1	85.4 ±0.1	5.3 ±0.0
ReWeight	77.6 ±0.0	67.0 ±0.4	76.8 ±0.0	69.7 ±0.0	77.1 ±0.0	73.4 ±0.0	76.9 ±0.1	77.6 ±0.0	84.9 ±0.0	5.2 ±0.4
SqrtReWeight	78.3 ±0.0	68.0 ±0.4	77.5 ±0.0	70.6 ±0.0	77.8 ±0.0	74.2 ±0.0	77.6 ±0.0	78.3 ±0.0	85.6 ±0.0	5.2 ±0.3
CBLoss	78.3 ±0.2	67.6 ±0.3	77.4 ±0.2	70.8 ±0.2	77.7 ±0.2	74.0 ±0.2	77.5 ±0.1	78.2 ±0.2	85.5 ±0.1	4.7 ±0.3
Focal	78.3 ±0.1	68.7 ±0.4	77.4 ±0.1	70.8 ±0.2	77.6 ±0.1	73.9 ±0.0	77.4 ±0.1	78.1 ±0.0	85.4 ±0.0	10.1 ±0.6
LDAM	78.0 ±0.1	66.6 ±0.6	77.2 ±0.2	70.4 ±0.2	77.4 ±0.2	73.7 ±0.2	77.2 ±0.2	77.9 ±0.2	85.2 ±0.1	22.5 ±0.2
BSoftmax	78.0 ±0.1	67.6 ±0.6	77.1 ±0.1	70.4 ±0.1	77.3 ±0.1	73.6 ±0.1	77.2 ±0.1	77.8 ±0.1	85.1 ±0.1	5.2 ±0.4
DFR	78.3 ±0.0	67.1 ±0.4	77.4 ±0.0	72.1 ±0.2	77.5 ±0.0	73.2 ±0.1	77.1 ±0.0	77.6 ±0.1	85.1 ±0.0	20.0 ±0.1
CRT	78.0 ±0.0	68.1 ±0.1	77.2 ±0.0	70.1 ±0.0	77.4 ±0.0	73.9 ±0.0	77.3 ±0.0	78.0 ±0.0	85.3 ±0.0	6.2 ±0.0
ReWeightCRT	77.8 ±0.0	67.9 ±0.1	77.0 ±0.0	70.0 ±0.0	77.2 ±0.0	73.6 ±0.0	77.0 ±0.0	77.8 ±0.0	85.0 ±0.0	5.4 ±0.0

MIMICNotes

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	91.1 ±0.1	24.2 ±2.8	76.4 ±0.9	60.3 ±2.1	65.8 ±1.2	36.5 ±2.5	62.2 ±1.3	62.4 ±1.4	85.2 ±0.1	2.2 ±0.1
Mixup	91.1 ±0.0	22.7 ±3.2	76.8 ±0.7	61.2 ±1.6	65.1 ±1.6	35.0 ±3.2	61.5 ±1.6	61.7 ±1.7	85.4 ±0.0	2.0 ±0.8
GroupDRO	83.2 ±2.4	62.6 ±6.3	64.7 ±1.4	33.7 ±3.4	66.4 ±1.3	42.8 ±1.1	74.3 ±1.5	74.4 ±1.5	85.1 ±0.1	13.8 ±3.2
CVaRDRO	90.2 ±0.0	0.0 ±0.0	45.1 ±0.0	0.0 ±0.0	47.4 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	71.6 ±0.4	40.2 ±0.0
JTT	71.3 ±3.7	65.9 ±2.8	60.3 ±0.7	23.4 ±1.9	58.6 ±2.3	36.0 ±1.8	75.5 ±0.4	75.6 ±0.4	84.9 ±0.1	27.5 ±3.9
LfF	84.0 ±1.2	62.7 ±2.1	64.6 ±0.7	33.6 ±1.6	67.1 ±0.8	43.6 ±0.8	74.7 ±0.4	74.7 ±0.5	85.1 ±0.0	12.5 ±1.2
LISA	91.1 ±0.0	22.7 ±3.2	76.8 ±0.7	61.2 ±1.6	65.1 ±1.6	35.0 ±3.2	61.5 ±1.6	61.7 ±1.7	85.4 ±0.0	2.0 ±0.8
ReSample	81.4 ±1.5	67.1 ±2.6	63.3 ±0.6	30.5 ±1.5	65.4 ±1.0	42.0 ±1.0	75.4 ±0.3	75.6 ±0.4	85.1 ±0.0	17.4 ±1.9
ReWeight	82.7 ±0.7	65.5 ±1.3	63.8 ±0.4	31.7 ±0.9	66.3 ±0.5	42.8 ±0.5	75.3 ±0.2	75.4 ±0.3	85.2 ±0.1	15.9 ±0.7
SqrtReWeight	90.3 ±0.2	35.7 ±4.0	72.4 ±0.9	51.3 ±2.1	68.7 ±0.9	42.8 ±2.1	66.7 ±1.6	66.8 ±1.6	85.2 ±0.1	3.7 ±0.7
CBLoss	78.2 ±1.0	72.3 ±1.3	61.9 ±0.3	27.3 ±0.8	63.2 ±0.7	39.8 ±0.6	76.1 ±0.2	76.2 ±0.2	85.0 ±0.0	20.6 ±1.3
Focal	91.0 ±0.0	19.1 ±2.3	77.1 ±0.6	62.1 ±1.4	63.6 ±1.3	31.9 ±2.6	59.9 ±1.1	60.2 ±1.1	85.3 ±0.1	8.1 ±0.7
LDAM	90.6 ±0.1	5.3 ±2.4	84.4 ±0.8	78.1 ±1.7	52.5 ±2.1	10.0 ±4.1	52.7 ±1.2	52.7 ±1.2	84.9 ±0.1	28.9 ±1.0
BSoftmax	76.9 ±0.9	73.1 ±1.0	61.7 ±0.2	26.5 ±0.6	62.5 ±0.5	39.3 ±0.4	76.6 ±0.2	76.7 ±0.2	85.4 ±0.0	23.5 ±1.1
DFR	69.2 ±1.3	67.3 ±1.7	58.8 ±0.2	21.0 ±0.5	56.5 ±0.8	33.1 ±0.5	73.1 ±0.0	73.1 ±0.0	81.0 ±0.0	38.4 ±0.1
CRT	77.8 ±0.0	73.1 ±0.0	61.6 ±0.0	26.7 ±0.0	62.8 ±0.0	39.2 ±0.0	75.9 ±0.0	75.9 ±0.0	84.3 ±0.0	23.0 ±0.1
ReWeightCRT	81.2 ±2.8	63.9 ±7.6	63.6 ±1.6	31.4 ±3.9	64.7 ±1.6	40.7 ±1.3	73.8 ±1.6	73.9 ±1.6	84.3 ±0.0	26.5 ±2.4

CXRMultisite

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	93.1 ±0.1	0.3 ±0.1
Mixup	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	92.9 ±0.1	0.3 ±0.0
GroupDRO	90.2 ±0.1	0.0 ±0.0	56.7 ±0.1	13.6 ±0.1	59.2 ±0.1	23.7 ±0.2	50.2 ±0.1	90.4 ±0.0	92.8 ±0.2	13.5 ±0.7
CVaRDRO	98.3 ±0.0	0.0 ±0.0	61.2 ±4.9	24.0 ±9.8	50.7 ±0.7	2.2 ±1.5	50.2 ±0.2	50.6 ±0.4	93.0 ±0.0	0.9 ±0.3
JTT	94.1 ±0.9	0.0 ±0.0	59.0 ±0.7	18.5 ±1.4	62.9 ±0.8	28.9 ±1.2	55.2 ±0.9	82.2 ±2.4	93.2 ±0.1	6.4 ±0.5
LfF	9.9 ±6.7	5.4 ±4.4	17.4 ±13.5	0.6 ±0.5	8.5 ±5.6	1.2 ±1.0	50.5 ±0.4	51.7 ±1.4	60.6 ±1.6	82.6 ±12.8
LISA	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	92.9 ±0.1	0.3 ±0.0
ReSample	88.3 ±1.6	0.1 ±0.1	55.9 ±0.6	12.0 ±1.2	57.3 ±1.4	20.9 ±1.8	50.3 ±0.5	88.1 ±0.7	92.3 ±0.1	13.0 ±2.8
ReWeight	89.5 ±0.0	0.3 ±0.1	56.4 ±0.0	13.0 ±0.0	58.5 ±0.0	22.7 ±0.0	50.5 ±0.2	90.3 ±0.0	93.2 ±0.1	17.7 ±1.7
SqrtReWeight	94.5 ±0.4	0.0 ±0.0	59.4 ±0.2	19.3 ±0.6	63.7 ±0.3	30.2 ±0.4	56.3 ±0.1	82.3 ±1.6	93.3 ±0.0	6.0 ±0.2
CBLoss	1.7 ±0.0	0.0 ±0.0	0.8 ±0.0	0.0 ±0.0	1.7 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	62.7 ±1.5	98.3 ±0.0
Focal	98.3 ±0.0	0.0 ±0.0	55.4 ±5.1	12.5 ±10.2	49.7 ±0.1	0.3 ±0.2	50.0 ±0.0	50.1 ±0.1	93.2 ±0.0	11.5 ±0.6
LDAM	98.3 ±0.0	0.0 ±0.0	49.2 ±0.0	0.0 ±0.0	49.6 ±0.0	0.0 ±0.0	50.0 ±0.0	50.0 ±0.0	93.1 ±0.1	0.3 ±0.1
BSoftmax	89.1 ±0.2	0.5 ±0.1	56.2 ±0.1	12.5 ±0.2	58.1 ±0.2	22.0 ±0.3	50.4 ±0.0	90.0 ±0.1	92.9 ±0.1	19.9 ±1.3
DFR	89.7 ±0.1	0.6 ±0.1	56.5 ±0.0	13.2 ±0.1	58.7 ±0.1	23.0 ±0.1	50.4 ±0.0	90.4 ±0.0	92.8 ±0.1	47.3 ±0.1
CRT	90.4 ±0.1	1.1 ±0.5	56.9 ±0.0	13.9 ±0.1	59.5 ±0.1	24.2 ±0.1	51.2 ±0.1	90.2 ±0.1	93.3 ±0.0	15.7 ±0.9
ReWeightCRT	89.9 ±0.1	1.4 ±0.6	56.6 ±0.0	13.4 ±0.1	59.0 ±0.1	23.3 ±0.1	51.1 ±0.3	90.4 ±0.0	93.1 ±0.1	15.5 ±0.7

CheXpert

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	88.6 ±0.7	41.7 ±3.4	68.3 ±1.1	42.3 ±2.4	68.7 ±0.2	43.8 ±0.7	69.5 ±1.3	70.0 ±1.5	85.4 ±0.4	5.0 ±1.1
Mixup	81.9 ±6.2	37.4 ±3.5	63.5 ±5.0	33.9 ±9.3	62.5 ±5.9	35.7 ±7.6	63.8 ±4.7	64.1 ±4.6	76.1 ±8.5	16.1 ±9.0
GroupDRO	79.2 ±0.2	74.7 ±0.3	62.9 ±0.1	28.6 ±0.2	64.7 ±0.2	42.0 ±0.2	78.4 ±0.2	79.1 ±0.1	86.2 ±0.0	21.3 ±0.8
CVaRDRO	73.7 ±1.0	50.2 ±1.8	57.3 ±0.1	20.1 ±0.3	56.8 ±0.4	29.9 ±0.2	65.7 ±0.6	67.0 ±0.4	72.9 ±0.4	40.4 ±0.0
JTT	75.2 ±0.8	60.4 ±4.8	59.4 ±1.1	23.0 ±1.5	59.6 ±1.4	34.4 ±2.3	70.7 ±2.6	72.0 ±2.5	79.0 ±2.5	24.4 ±0.5
LfF	22.3 ±10.2	13.7 ±9.8	37.3 ±5.8	9.0 ±0.7	19.5 ±8.3	8.8 ±3.9	46.2 ±2.9	46.2 ±3.1	30.5 ±10.1	65.7 ±10.2
LISA	81.9 ±6.2	37.4 ±3.5	63.5 ±5.0	33.9 ±9.3	62.5 ±5.9	35.7 ±7.6	63.8 ±4.7	64.1 ±4.6	76.1 ±8.5	16.1 ±9.0
ReSample	79.6 ±0.6	74.3 ±0.4	63.1 ±0.3	29.0 ±0.6	65.0 ±0.5	42.3 ±0.6	78.3 ±0.3	79.0 ±0.2	86.3 ±0.2	20.1 ±1.4
ReWeight	79.6 ±0.5	73.7 ±1.0	63.1 ±0.2	29.0 ±0.5	65.0 ±0.4	42.4 ±0.5	78.4 ±0.3	79.1 ±0.2	86.2 ±0.1	21.0 ±0.7
SqrtReWeight	83.5 ±0.3	68.5 ±1.6	65.0 ±0.2	33.3 ±0.4	67.9 ±0.2	45.5 ±0.3	77.8 ±0.4	78.5 ±0.3	86.3 ±0.3	15.6 ±1.1
CBLoss	80.0 ±0.5	74.0 ±0.7	63.3 ±0.2	29.4 ±0.5	65.2 ±0.4	42.6 ±0.5	78.6 ±0.2	79.0 ±0.2	86.1 ±0.3	19.6 ±0.6
Focal	89.3 ±0.3	42.1 ±4.0	69.6 ±0.4	44.7 ±1.1	69.8 ±0.4	45.5 ±1.0	70.4 ±1.1	70.4 ±1.3	86.5 ±0.1	16.1 ±1.7
LDAM	90.0 ±0.0	36.0 ±0.7	70.6 ±0.1	47.3 ±0.2	69.1 ±0.1	43.6 ±0.1	67.8 ±0.2	67.9 ±0.1	86.1 ±0.1	32.3 ±0.2
BSoftmax	79.9 ±0.2	73.8 ±1.0	63.3 ±0.0	29.4 ±0.1	65.3 ±0.1	42.8 ±0.0	78.6 ±0.2	79.5 ±0.1	86.6 ±0.1	21.3 ±0.4
DFR	79.1 ±0.0	75.8 ±0.3	63.0 ±0.0	28.6 ±0.0	64.7 ±0.0	42.1 ±0.0	78.8 ±0.0	79.3 ±0.0	86.0 ±0.0	25.6 ±0.1
CRT	79.3 ±0.1	74.6 ±0.4	62.9 ±0.1	28.7 ±0.1	64.7 ±0.1	42.0 ±0.1	78.5 ±0.1	79.0 ±0.1	86.1 ±0.1	21.7 ±0.1
ReWeightCRT	79.3 ±0.1	75.1 ±0.2	63.0 ±0.0	28.7 ±0.0	64.7 ±0.0	42.0 ±0.0	78.6 ±0.1	79.0 ±0.1	86.2 ±0.0	21.7 ±0.1

Living17

Algorithm	Avg Acc.	Worst Acc.	Avg Prec.	Worst Prec.	Avg F1	Worst F1	Adjusted Acc.	Balanced Acc.	AUROC	ECE
ERM	27.7 ±1.1	5.7 ±1.5	28.2 ±1.0	8.2 ±1.6	27.1 ±1.1	6.9 ±1.7	27.7 ±1.1	27.7 ±1.1	77.3 ±1.3	59.6 ±0.5
Mixup	29.8 ±1.8	8.7 ±1.4	30.9 ±2.2	9.5 ±1.3	29.3 ±1.9	9.7 ±1.5	29.8 ±1.8	29.8 ±1.8	78.2 ±1.2	34.1 ±1.9
GroupDRO	31.1 ±1.0	6.0 ±1.4	32.1 ±0.9	9.6 ±0.2	30.8 ±0.7	7.6 ±0.8	31.1 ±1.0	31.1 ±1.0	80.1 ±0.8	53.5 ±0.8
CVaRDRO	27.3 ±1.6	4.0 ±0.5	29.2 ±1.7	5.1 ±0.8	26.5 ±1.5	4.8 ±0.6	27.3 ±1.6	27.3 ±1.6	81.0 ±0.2	28.8 ±6.8
JTT	28.3 ±1.1	5.7 ±2.2	31.1 ±0.9	8.0 ±1.7	28.3 ±1.3	7.2 ±2.3	28.3 ±1.1	28.3 ±1.1	81.0 ±0.8	36.9 ±3.2
LfF	26.4 ±1.3	7.0 ±1.2	28.3 ±0.8	9.6 ±1.8	26.1 ±1.2	8.7 ±1.7	26.4 ±1.3	26.4 ±1.3	76.6 ±0.6	61.0 ±0.7
LISA	29.8 ±1.8	8.7 ±1.4	30.9 ±2.2	9.5 ±1.3	29.3 ±1.9	9.7 ±1.5	29.8 ±1.8	29.8 ±1.8	78.2 ±1.2	34.1 ±1.9
ReSample	31.4 ±0.6	6.7 ±1.5	33.0 ±0.6	11.0 ±0.6	31.0 ±0.6	8.3 ±1.2	31.4 ±0.6	31.4 ±0.6	81.0 ±0.7	46.6 ±3.1
ReWeight	27.7 ±1.1	5.7 ±1.5	28.2 ±1.0	8.2 ±1.6	27.1 ±1.1	6.9 ±1.7	27.7 ±1.1	27.7 ±1.1	77.3 ±1.3	59.6 ±0.5
SqrtReWeight	27.7 ±1.1	5.7 ±1.5	28.2 ±1.0	8.2 ±1.6	27.1 ±1.1	6.9 ±1.7	27.7 ±1.1	27.7 ±1.1	77.3 ±1.3	59.6 ±0.5
CBLoss	27.7 ±1.1	5.7 ±1.5	28.2 ±1.0	8.2 ±1.6	27.1 ±1.1	6.9 ±1.7	27.7 ±1.1	27.7 ±1.1	77.3 ±1.3	59.6 ±0.5
Focal	26.9 ±0.6	5.3 ±0.3	28.8 ±1.0	7.1 ±1.0	27.0 ±0.7	6.3 ±0.5	26.9 ±0.6	26.9 ±0.6	78.7 ±0.5	49.9 ±1.9
LDAM	24.3 ±0.8	4.0 ±0.8	28.0 ±1.2	6.6 ±1.4	24.0 ±0.8	5.1 ±0.3	24.3 ±0.8	24.3 ±0.8	79.1 ±1.0	12.4 ±0.5
BSoftmax	28.6 ±1.4	6.7 ±0.7	30.7 ±1.0	8.2 ±0.8	28.3 ±1.3	7.3 ±0.5	28.6 ±1.4	28.6 ±1.4	78.0 ±1.0	56.5 ±1.7
DFR	26.3 ±0.4	6.0 ±0.9	27.4 ±0.3	8.6 ±0.8	25.7 ±0.2	7.5 ±1.1	26.3 ±0.4	26.3 ±0.4	79.4 ±0.1	13.8 ±0.4
CRT	31.1 ±0.1	6.3 ±0.3	31.8 ±0.0	7.5 ±0.3	30.5 ±0.1	6.8 ±0.3	31.1 ±0.1	31.1 ±0.1	80.3 ±0.1	49.6 ±1.7
ReWeightCRT	33.1 ±0.1	9.3 ±0.3	33.4 ±0.1	11.3 ±0.3	32.6 ±0.0	10.8 ±0.2	33.1 ±0.1	33.1 ±0.1	82.0 ±0.0	40.0 ±0.4

Overall

Algorithm	Waterbirds	CelebA	CivilComments	MultiNLI	MetaShift	ImageNetBG	NICO++	MIMIC-CXR	MIMICNotes	CXRMultisite	CheXpert	Living17	Avg
ERM	69.1 ±4.7	57.6 ±0.8	63.2 ±1.2	66.4 ±2.3	82.1 ±0.8	76.8 ±0.9	35.0 ±4.1	68.6 ±0.2	80.4 ±0.2	50.1 ±0.9	41.7 ±3.4	27.7 ±1.1	59.9
Mixup	77.5 ±0.7	57.8 ±0.8	65.8 ±1.5	66.8 ±0.3	79.0 ±0.8	76.9 ±0.7	30.0 ±4.1	66.8 ±0.6	81.6 ±0.6	50.1 ±0.9	37.4 ±3.5	29.8 ±1.8	60.0
GroupDRO	73.1 ±0.4	68.3 ±0.9	61.5 ±1.8	64.1 ±0.8	83.1 ±0.7	76.4 ±0.2	31.1 ±0.9	67.4 ±0.5	83.7 ±0.1	59.2 ±0.3	74.7 ±0.3	31.1 ±1.0	64.5
CVaRDRO	75.5 ±2.2	60.2 ±3.0	62.9 ±3.8	48.2 ±3.4	83.5 ±0.5	74.8 ±0.8	27.8 ±2.3	68.0 ±0.2	65.6 ±1.5	50.2 ±0.9	50.2 ±1.8	27.3 ±1.6	57.8
JTT	71.2 ±0.5	48.3 ±1.5	51.0 ±4.2	65.1 ±1.6	82.6 ±0.4	77.0 ±0.4	30.6 ±2.3	64.9 ±0.3	83.8 ±0.1	57.9 ±2.1	60.4 ±4.8	28.3 ±1.1	60.1
LfF	75.0 ±0.7	53.0 ±4.3	42.2 ±7.2	57.3 ±5.7	72.3 ±1.3	70.1 ±1.4	28.8 ±2.0	62.2 ±2.4	84.0 ±0.1	50.1 ±0.9	13.7 ±9.8	26.4 ±1.3	52.9
LISA	77.5 ±0.7	57.8 ±0.8	65.8 ±1.5	66.8 ±0.3	79.0 ±0.8	76.9 ±0.7	30.0 ±4.1	66.8 ±0.6	81.6 ±0.6	50.1 ±0.9	37.4 ±3.5	29.8 ±1.8	60.0
ReSample	70.0 ±1.0	74.1 ±2.2	61.0 ±0.6	66.8 ±0.5	81.0 ±1.7	77.7 ±1.1	30.6 ±2.3	67.5 ±0.3	82.6 ±0.6	55.0 ±0.2	74.3 ±0.4	31.4 ±0.6	64.3
ReWeight	71.9 ±0.6	69.6 ±0.2	59.3 ±1.1	64.2 ±1.9	83.1 ±0.7	76.8 ±0.9	25.0 ±0.0	67.0 ±0.4	84.0 ±0.1	61.4 ±1.3	73.7 ±1.0	27.7 ±1.1	63.6
SqrtReWeight	71.0 ±1.4	66.9 ±2.2	68.6 ±1.1	63.8 ±2.4	82.6 ±0.4	76.8 ±0.9	32.8 ±3.5	68.0 ±0.4	83.1 ±0.2	61.2 ±0.6	68.5 ±1.6	27.7 ±1.1	64.2
CBLoss	74.4 ±1.2	65.4 ±1.4	67.3 ±0.2	63.6 ±2.4	83.1 ±0.0	76.8 ±0.9	31.7 ±3.6	67.6 ±0.3	84.0 ±0.1	50.2 ±0.9	74.0 ±0.7	27.7 ±1.1	63.8
Focal	71.6 ±0.8	56.9 ±3.4	61.9 ±1.1	62.4 ±2.0	81.0 ±0.4	71.9 ±1.2	30.6 ±2.3	68.7 ±0.4	70.9 ±9.8	50.0 ±0.9	42.1 ±4.0	26.9 ±0.6	57.9
LDAM	70.9 ±1.7	57.0 ±4.1	28.4 ±7.7	65.5 ±0.8	83.6 ±0.4	76.7 ±0.5	31.7 ±3.6	66.6 ±0.6	81.0 ±0.3	50.1 ±0.9	36.0 ±0.7	24.3 ±0.8	56.0
BSoftmax	74.1 ±0.9	69.6 ±1.2	58.3 ±1.1	63.6 ±2.4	82.6 ±0.4	76.1 ±2.0	35.6 ±1.8	67.6 ±0.6	83.8 ±0.3	58.6 ±1.8	73.8 ±1.0	28.6 ±1.4	64.4
DFR	89.0 ±0.2	73.7 ±0.8	64.4 ±0.1	63.8 ±0.0	81.4 ±0.1	74.4 ±1.8	38.0 ±3.8	67.1 ±0.4	80.2 ±0.0	60.8 ±0.4	75.8 ±0.3	26.3 ±0.4	66.2
CRT	76.3 ±0.8	69.6 ±0.7	67.8 ±0.3	65.4 ±0.2	83.1 ±0.0	78.2 ±0.5	33.3 ±0.0	68.1 ±0.1	83.4 ±0.0	61.8 ±0.1	74.6 ±0.4	31.1 ±0.1	66.1
ReWeightCRT	76.3 ±0.2	70.7 ±0.6	64.7 ±0.2	65.2 ±0.2	85.1 ±0.4	77.5 ±0.7	33.3 ±0.0	67.9 ±0.1	83.4 ±0.0	53.1 ±2.3	75.1 ±0.2	33.1 ±0.1	65.4

References

[1] Yuzhe Yang, Xin Liu, Jiang Wu, Silviu Borac, Dina Katabi, Ming-Zher Poh, and Daniel McDuff. Simper: Simple self-supervised learning of periodic targets. In *International Conference on Learning Representations*, 2023.

[2] Yuzhe Yang, Kaiwen Zha, Ying-Cong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *International Conference on Machine Learning (ICML)*, 2021.

[3] Yuzhe Yang, Hao Wang, and Dina Katabi. On multi-domain long-tailed recognition, imbalanced domain generalization and beyond. In *European Conference on Computer Vision (ECCV)*, 2022.

[4] Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. In *International Conference on Machine Learning*, 2023.

[5] Yuzhe Yang, Yuan Yuan, Guo Zhang, Hao Wang, Ying-Cong Chen, Yingcheng Liu, Christopher G Tarolli, Daniel Crepeau, Jan Bukartyk, Mithri R Junna, et al. Artificial intelligence-enabled detection and assessment of parkinson's disease using nocturnal breathing signals. *Nature Medicine*, 28(10):2207–2215, 2022.

[6] Shichao Yue, Yuzhe Yang, Hao Wang, Hariharan Rahul, and Dina Katabi. Bodycompass: Monitoring sleep posture with wireless signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2), 2020.

[7] Yuzhe Yang, Haoran Zhang, Judy W Gichoya, Dina Katabi, and Marzyeh Ghassemi. The limits of fair medical imaging ai in the wild. *arXiv preprint arXiv:2312.10083*, 2023.

[8] Maksut Senbekov, Timur Saliev, Zhanar Bukeyeva, Aigul Almabayeva, Marina Zhanaliyeva, Nazym Aitenova, Yerzhan Toishibekov, Ildar Fakhradiyev, et al. The recent progress and applications of digital technologies in healthcare: a review. *International journal of telemedicine and applications*, 2020.

[9] Food and Drug Administration. Artificial intelligence and machine learning (ai/ml) medical devices, 2024.

[10] Michael J. Armstrong and Michael S. Okun. Diagnosis and treatment of parkinson disease: a review. *JAMA*, 323:548–560, 2020.

[11] Robert Kloster and Torstein Engelskjøn. Sudden unexpected death in epilepsy (sudep): a clinical perspective and a search for risk factors. *Journal of Neurology, Neurosurgery & Psychiatry*, 67(4):439–444, 1999.

[12] Woo Suk Hong, Adrian Daniel Haimovich, and R Andrew Taylor. Predicting hospital admission at emergency department triage using machine learning. *PloS one*, 13(7):e0201016, 2018.

[13] Girish Narayanswamy, Yujia Liu, Yuzhe Yang, Chengqian Ma, Xin Liu, Daniel McDuff, and Shwetak Patel. Bigsmall: Efficient multi-task learning for disparate spatial and temporal physiological measurements. *Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[14] H. Brunner et al. Microstructure of the non-rapid eye movement sleep electroencephalogram in patients with newly diagnosed parkinson's disease: effects of dopaminergic treatment. *Mov Disord.*, 17:928–933, 2002.

[15] Brandon Philip Theodorou, Cao Xiao, and Jimeng Sun. Treement: Interpretable patient-trial matching via personalized dynamic tree-based memory network. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1–9, 2023.

[16] Yingcheng Liu, Guo Zhang, Christopher G Tarolli, Rumen Hristov, Stella Jensen-Roberts, Emma M Waddell, Taylor L Myers, Meghan E Pawlik, Julia M Soto, Renee M Wilson, et al. Monitoring gait at home with radio waves in parkinsonâ€™s disease: A marker of severity, progression, and medication response. *Science Translational Medicine*, 14(663):eadc9669, 2022.

[17] Yiwen Gu, Shreya Pandit, Elham Saraee, Timothy Nordahl, Terry Ellis, and Margrit Betke. Home-based physical therapy with an interactive computer vision system. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[18] Timothy P Hanna, Will D King, Stephane Thibodeau, Matthew Jalink, Gregory A Paulin, Elizabeth Harvey-Jones, Dylan E O'Sullivan, Christopher M Booth, Richard Sullivan, and Ajay Aggarwal. Mortality due to cancer treatment delay: systematic review and meta-analysis. *bmj*, 371, 2020.

[19] Albert Haque, Arnold Milstein, and Li Fei-Fei. Illuminating the dark spaces of healthcare with ambient intelligence. *Nature*, 585(7824):193–202, 2020.

[20] Bruce Leff. Defining and disseminating the hospital-at-home model. *Cmaj*, 180(2):156–157, 2009.

[21] Andrew Wong, Erkin Otles, John P Donnelly, Andrew Krumm, Jeffrey McCullough, Olivia DeTroyer-Cooley, Justin Pestrue, Marie Phillips, Judy Konye, Carleen Penoza, et al. External validation of a widely implemented proprietary sepsis prediction model in hospitalized patients. *JAMA internal medicine*, 181(8):1065–1070, 2021.

[22] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*, 2020.

[23] Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: Learning continuous representations for regression. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[24] Guo Zhang, Ipsit V Vahia, Yingcheng Liu, Yuzhe Yang, Rose May, Hailey V Cray, William McGrory, and Dina Katabi. Contactless in-home monitoring of the long-term respiratory and behavioral phenotypes in older adults with covid-19: A case series. *Frontiers in psychiatry*, 12, 2021.

[25] Yuzhe Yang, Yujia Liu, Xin Liu, Avanti Gulhane, Domenico Mastrodicasa, Wei Wu, Edward J Wang, Dushyant W Sahani, and Shwetak Patel. Demographic bias of expert-level vision-language foundation models in medical imaging. *arXiv preprint arXiv:2402.14815*, 2024.

[26] Anurag Vaidya, Richard J Chen, Drew FK Williamson, Andrew H Song, Guillaume Jaume, Yuzhe Yang, Thomas Hartvigsen, Emma C Dyer, Ming Y Lu, Jana Lipkova, et al. Demographic bias in misdiagnosis by computational pathology models. *Nature Medicine*, 30(4):1174–1190, 2024.

[27] E. Ray Dorsey, Todd Sherer, Michael S. Okun, and Bastiaan R. Bloem. The emerging evidence of the parkinson pandemic. *J. Parkinsons Dis.*, 8:S3–S8, 2018.

[28] Connie Marras, J. C. Beck, J. H. Bower, E. Roberts, B. Ritz, et al. Prevalence of parkinson's disease across north america. *npj Parkinson's Disease*, 4:21, 2018.

[29] W. Yang, J.L. Hamilton, C. Kopil, et al. Current and projected future economic burden of parkinson's disease in the u.s. *npj Parkinsons Dis.*, 6:15, 2020.

[30] Stuart F. Quan, Barbara V. Howard, Conrad Iber, James P. Kiley, F. Javier Nieto, George T. O'Connor, David M. Rapoport, Susan Redline, John Robbins, Jonathan M. Samet, and Patricia W. Wahl. The sleep heart health study: design, rationale, and methods. *Sleep*, 20:1077–1085, 1997.

[31] Lasse Espeholt, Shreya Agrawal, Casper Sønderby, Manoj Kumar, Jonathan Heek, Carla Bromberg, Cenk Gazen, Jason Hickey, Aaron Bell, and Nal Kalchbrenner. Skillful twelve hour precipitation forecasts using large context neural networks. *arXiv preprint arXiv:2111.07470*, 2021.

[32] Hong Luo, Deye Yang, Andrew Barszczyk, Naresh Vempala, Jing Wei, Si Jia Wu, Paul Pu Zheng, Genyue Fu, Kang Lee, and Zhong-Ping Feng. Smartphone-based blood pressure measurement using transdermal optical imaging technology. *Circulation: Cardiovascular Imaging*, 12(8):e008857, 2019.

[33] Bryan P Yan, William HS Lai, Christy KY Chan, Stephen Chun-Hin Chan, Lok-Hei Chan, Ka-Ming Lam, Ho-Wang Lau, Chak-Ming Ng, Lok-Yin Tai, Kin-Wai Yip, et al. Contact-free screening of atrial fibrillation by a smartphone using facial pulsatile photoplethysmographic signals. *Journal of the American Heart Association*, 7(8):e008585, 2018.

[34] Robert Amelard, Kaylen J Pfisterer, Shubh Jagani, David A Clausi, and Alexander Wong. Non-contact assessment of obstructive sleep apnea cardiovascular biomarkers using photoplethysmography imaging. In *Optical Diagnostics and Sensing XVIII: Toward Point-of-Care Diagnostics*, volume 10501, pages 225–229. SPIE, 2018.

[35] Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. Metnet: A neural weather model for precipitation forecasting. *arXiv preprint arXiv:2003.12140*, 2020.

[36] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[37] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[38] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10539–10548, 2021.

[39] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6964–6974, 2021.

[40] Hanzhe Hu, Jinshi Cui, and Liwei Wang. Region-aware contrastive learning for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16291–16301, 2021.

[41] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[42] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM transactions on graphics (TOG)*, 31(4):1–8, 2012.

[43] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10387–10396, 2020.

[44] Mohammad Rafayet Ali, Javier Hernandez, E Ray Dorsey, Ehsan Hoque, and Daniel McDuff. Spatio-temporal attention and magnification for classification of parkinson's disease from videos collected via the internet. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 207–214. IEEE, 2020.

[45] Xin Liu, Josh Fromm, Shwetak Patel, and Daniel McDuff. Multi-task temporal shift attention networks for on-device contactless vitals measurement. *Advances in Neural Information Processing Systems*, 33:19400–19411, 2020.

[46] Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.

[47] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *Asian conference on computer vision*, pages 669–684. Springer, 2018.

[48] Xin Liu, Ziheng Jiang, Josh Fromm, Xuhai Xu, Shwetak Patel, and Daniel McDuff. Metaphys: few-shot adaptation for non-contact physiological measurement. In *Proceedings of the conference on health, inference, and learning*, pages 154–163, 2021.

[49] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, pages 766–774, 2014.

[50] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[51] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017.

[52] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[53] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[54] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[55] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv:1906.05849*, 2019.

[56] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *European Conference on Computer Vision*, pages 425–442. Springer, 2020.

[57] Lloyd Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information theory*, 20(3):397–399, 1974.

[58] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[59] Daniel McDuff, Miah Wander, Xin Liu, Brian L Hill, Javier Hernandez, Jonathan Lester, and Tadas Baltrusaitis. Scamps: Synthetics for camera measurement of physiological signals. *arXiv preprint arXiv:2206.04197*, 2022.

[60] Serge Bobbia, Richard Macwan, Yannick Benezeth, Alamin Mansouri, and Julien Dubois. Unsupervised skin tissue segmentation for remote photoplethysmography. *Pattern Recognition Letters*, 124:82–90, 2019.

[61] Ronny Stricker, Steffen Müller, and Horst-Michael Gross. Non-contact video-based pulse rate measurement on a mobile service robot. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 1056–1062. IEEE, 2014.

[62] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[63] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

[64] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[65] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[66] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[67] Danish Contractor, Daniel McDuff, Julia Katherine Haines, Jenny Lee, Christopher Hines, Brent Hecht, Nicholas Vincent, and Hanlin Li. Behavioral use licensing for responsible ai. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 778–788, 2022.

[68] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[69] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

[70] Chen Huang, Yining Li, Change Loy Chen, and Xiaoou Tang. Deep imbalanced learning for face recognition and attribute prediction. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[71] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019.

[72] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

[73] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

[74] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[75] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE international joint conference on neural networks*, pages 1322–1328, 2008.

[76] Salvador García and Francisco Herrera. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary computation*, 17(3):275–306, 2009.

[77] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, 2016.

[78] Qi Dong, Shaogang Gong, and Xiatian Zhu. Imbalanced deep learning by minority class incremental rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6):1367–1381, 2019.

[79] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *CVPR*, 2019.

[80] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *ICCV*, 2017.

[81] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *arXiv preprint arXiv:1902.07379*, 2019.

[82] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *ICLR*, 2020.

[83] Luís Torgo, Rita P Ribeiro, Bernhard Pfahringer, and Paula Branco. Smote for regression. In *Portuguese conference on artificial intelligence*, pages 378–389. Springer, 2013.

[84] Paula Branco, Luís Torgo, and Rita P Ribeiro. Smogn: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, pages 36–50. PMLR, 2017.

[85] Paula Branco, Luis Torgo, and Rita P Ribeiro. Rebagg: Resampled bagging for imbalanced regression. In *Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 67–81. PMLR, 2018.

[86] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

[87] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2-4):144–157, 2018.

[88] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[89] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, volume 30, 2016.

[90] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: The first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, volume 2, page 5, 2017.

[91] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 1–14, 2017.

[92] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *EMNLP 2018*, page 353, 2018.

[93] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[94] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *WACV*, 2019.

[95] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. *arXiv preprint arXiv:2011.14141*, 2020.

[96] Stuart F Quan, Barbara V Howard, Conrad Iber, James P Kiley, F Javier Nieto, George T O'Connor, David M Rapoport, Susan Redline, John Robbins, Jonathan M Samet, et al. The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12):1077–1085, 1997.

[97] John E Ware Jr and Cathy Donald Sherbourne. The mos 36-item short-form health survey (sf-36): I. conceptual framework and item selection. *Medical care*, pages 473–483, 1992.

[98] Hao Wang, Chengzhi Mao, Hao He, Mingmin Zhao, Tommi S Jaakkola, and Dina Katabi. Bidirectional inference networks: A class of deep bayesian networks for health profiling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 766–773, 2019.

[99] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[100] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. In *NeurIPS*, 2020.

[101] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021.

[102] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *ICLR*, 2021.

[103] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. *arXiv preprint arXiv:2107.09249*, 2021.

[104] Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogerio Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. *arXiv preprint arXiv:2111.13998*, 2021.

[105] Mark Dredze, Alex Kulesza, and Koby Crammer. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(1):123–149, 2010.

[106] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[107] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[108] Alice Schoenauer-Sebag, Louise Heinrich, Marc Schoenauer, Michele Sebag, Lani F Wu, and Steve J Altschuler. Multi-domain adversarial learning. In *ICLR*, 2019.

[109] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016.

[110] Yongxin Yang and Timothy M Hospedales. A unified perspective on multi-domain and multi-task learning. In *ICLR*, 2015.

[111] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

[112] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *AAAI*, 2018.

[113] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(1):2096–2030, 2016.

[114] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization in vision: A survey. *arXiv preprint arXiv:2103.02503*, 2021.

[115] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013.

[116] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018.

[117] Marvin Zhang, Henrik Marklund, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. *arXiv preprint arXiv:2007.02931*, 2020.

[118] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021.

[119] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019.

[120] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020.

[121] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[122] J Douglas Carroll and Phipps Arabie. Multidimensional scaling. *Measurement, judgment and decision making*, pages 179–250, 1998.

[123] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[124] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. In *NeurIPS*, 2004.

[125] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016.

[126] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. *CVPR*, 2020.

[127] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[128] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *ICCV*, 2013.

[129] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

[130] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[131] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018.

[132] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[133] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021.

[134] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[135] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *AAAI*, 2020.

[136] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *CVPR*, 2021.

[137] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018.

[138] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *Journal of Machine Learning Research*, 22(2):1–55, 2021.

[139] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

[140] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[141] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[142] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[143] Haoran Zhang, Amy X Lu, Mohamed Abdalla, Matthew McDermott, and Marzyeh Ghassemi. Hurtful words: quantifying biases in clinical contextual word embeddings. In *proceedings of the ACM Conference on Health, Inference, and Learning*, pages 110–120, 2020.

[144] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

[145] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[146] Herbert A Simon. Spurious correlation: A causal interpretation. *Journal of the American statistical Association*, 49(267):467–479, 1954.

[147] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[148] Alex J DeGrave, Joseph D Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

[149] Natalia L Martinez, Martin A Bertran, Afroditi Papadaki, Miguel Rodrigues, and Guillermo Sapiro. Blind pareto fairness and subgroup robustness. In *International Conference on Machine Learning*, pages 7492–7501. PMLR, 2021.

[150] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*, 2020.

[151] Tianle Cai, Ruiqi Gao, Jason Lee, and Qi Lei. A theory of label propagation for subpopulation shift. In *International Conference on Machine Learning*, pages 1170–1182. PMLR, 2021.

[152] Nitish Joshi, Xiang Pan, and He He. Are all spurious features in natural language alike? an analysis through a causal lens. *arXiv preprint arXiv:2210.14011*, 2022.

[153] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. *arXiv preprint arXiv:2201.00299*, 2022.

[154] Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew Gordon Wilson. On feature learning in the presence of spurious correlations. *arXiv preprint arXiv:2210.11369*, 2022.

[155] Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. *arXiv preprint arXiv:2204.02070*, 2022.

[156] Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*, 2020.

[157] Sindhu Gowda, Shalmali Joshi, Haoran Zhang, and Marzyeh Ghassemi. Pulling up by the causal bootstraps: Causal data augmentation for pre-training debiasing. *arXiv preprint arXiv:2108.12510*, 2021.

[158] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[159] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

[160] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pages 336–351. PMLR, 2022.

[161] Zongbo Han, Zhipeng Liang, Fan Yang, Liu Liu, Lanqing Li, Yatao Bian, Peilin Zhao, Bingzhe Wu, Changqing Zhang, and Jianhua Yao. Umix: Improving importance weighting for subpopulation shift via uncertainty-aware mixup. *arXiv preprint arXiv:2209.08928*, 2022.

[162] Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. Harnessing structures for value-based planning and reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[163] Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. When NAS meets robustness: In search of robust architectures against adversarial attacks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[164] Gregory Holste, Song Wang, Ajay Jaiswal, Yuzhe Yang, Mingquan Lin, Yifan Peng, and Atlas Wang. Cxr-lt: Multi-label long-tailed classification on chest x-rays. *PhysioNet*, 2023.

[165] Stephen R Pfohl, Haoran Zhang, Yizhe Xu, Agata Foryciarz, Marzyeh Ghassemi, and Nigam H Shah. A comparison of approaches to improve worst-case predictive model performance over patient subpopulations. *Scientific reports*, 12(1):1–13, 2022.

[166] Yongshuo Zong, Yongxin Yang, and Timothy Hospedales. Medfair: Benchmarking fairness for medical imaging. *arXiv preprint arXiv:2210.01725*, 2022.

[167] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33:728–740, 2020.

[168] Natalia Martinez, Martin Bertran, and Guillermo Sapiro. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning*, pages 6755–6764. PMLR, 2020.

[169] Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. Nico++: Towards better benchmarking for domain generalization. *arXiv preprint arXiv:2204.08040*, 2022.

[170] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[171] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[172] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Krishnamurthy Dvijotham, and Taylan Cemgil. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

[173] Tan Wang, Zhongqi Yue, Jianqiang Huang, Qianru Sun, and Hanwang Zhang. Self-supervised learning disentangled group representation as feature. *Advances in Neural Information Processing Systems*, 34:18225–18240, 2021.

[174] Kaihua Tang, Mingyuan Tao, Jiaxin Qi, Zhenguang Liu, and Hanwang Zhang. Invariant feature learning for generalized long-tailed classification. In *ECCV*, pages 709–726. Springer, 2022.

[175] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*, 2011.

[176] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[177] Weixin Liang and James Zou. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. *arXiv preprint arXiv:2202.06523*, 2022.

[178] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

[179] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

[180] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[181] Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

[182] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

[183] Irene Y Chen, Peter Szolovits, and Marzyeh Ghassemi. Can ai help reduce disparities in general medical and mental health care? *AMA journal of ethics*, 21(2):167–179, 2019.

[184] Aahlad Manas Puli, Lily H Zhang, Eric Karl Oermann, and Rajesh Ranganath. Out-of-distribution generalization in the presence of nuisance-induced spurious correlations. In *International Conference on Learning Representations*, 2021.

[185] John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

[186] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.

[187] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[188] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on Artificial Intelligence*, volume 56, pages 111–117. Citeseer, 2000.

[189] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[190] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*, 2022.

[191] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pages 7721–7735. PMLR, 2021.

[192] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[193] Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens Van Der Maaten. Revisiting weakly supervised pre-training of visual perception models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 804–814, 2022.

[194] Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigatingythu one amplifies others. *arXiv preprint arXiv:2212.04825*, 2022.

[195] Lasse F Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint arXiv:2007.03051*, 2020.

[196] Farhad Maleki, Nikesh Muthukrishnan, Katie Ovens, Caroline Reinhold, and Reza Forghani. Machine learning algorithm validation: from essentials to advanced applications and implications for regulatory certification and deployment. *Neuroimaging Clinics*, 30(4):433–445, 2020.

[197] Jenna Wiens, Suchi Saria, Mark Sendak, Marzyeh Ghassemi, Vincent X Liu, Finale Doshi-Velez, Kenneth Jung, Katherine Heller, David Kale, Mohammed Saeed, et al. Do no harm: a roadmap for responsible machine learning for health care. *Nature medicine*, 25(9):1337–1340, 2019.

[198] Marion Delenclos, Darren R. Jones, Pamela J. McLean, and Ryan J. Uitti. Biomarkers in parkinson's disease: advances and strategies. *Parkinsonism Relat Disord.*, 22:S106–S110, 2016.

[199] Joseph Jankovic. Parkinson's disease: Clinical features and diagnosis. *J Neurol Neurosurg Psychiatry*, 79:368–376, 2008.

[200] Robert A. Hauser et al. A home diary to assess functional status in patients with parkinson's disease with motor fluctuations and dyskinesia. *Clin Neuropharmacol.*, 23:75–81, 2000.

[201] Christopher G. Goetz et al. Movement disorder society-sponsored revision of the unified parkinson's disease rating scale (mds-updrs): Scale presentation and clinimetric testing results. *Mov Disord.*, 23:2129–2170, 2008.

[202] L. J. W. Evers, J. H. Krijthe, M. J. Meinders, Bastiaan R. Bloem, and T. M. Heskes. Measuring parkinson's disease over time: The real-world within-subject reliability of the mds-updrs. *Mov. Disord.*, 34:1480–1487, 2019.

[203] A. Regnault et al. Does the mds-updrs provide the precision to assess progression in early parkinson's disease? learnings from the parkinson's progression marker initiative cohort. *J. Neurol.*, 266:1927–1936, 2019.

[204] Terry D. Ellis et al. Identifying clinical measures that most accurately reflect the progression of disability in parkinson disease. *Parkinsonism Relat. Disord.*, 25:65–71, 2016.

[205] Dilan Athauda and Thomas Foltynie. The ongoing pursuit of neuroprotective therapies in parkinson disease. *Nat Rev Neurol.*, 11:25–40, 2015.

[206] Karl Kieburtz, Richard Katz, and C. Warren Olanow. New drugs for parkinson's disease: The regulatory and clinical development pathways in the united states. *Mov Disord.*, 33:920–927, 2018.

[207] Jun Zhang et al. Longitudinal assessment of tau and amyloid beta in cerebrospinal fluid of parkinson disease. *Acta Neuropathol.*, 126:671–682, 2013.

[208] Lucilla Parnetti et al. Cerebrospinal fluid glucocerebrosidase activity is reduced in parkinson's disease patients. *Mov Disord.*, 32:1423–1431, 2017.

[209] Lucilla Parnetti et al. Csf and blood biomarkers for parkinson's disease. *Lancet Neurol.*, 18:573–586, 2019.

[210] Yan Tang et al. Identifying the presence of parkinson's disease using low-frequency fluctuations in bold signals. *Neurosci Lett.*, 645:1–6, 2017.

[211] James Parkinson. An essay on the shaking palsy. *J. Neuropsychiatry Clin. Neurosci.*, 14, 2002. Originally published in 1817.

[212] Eduardo E. Benarroch, Ann M. Schmeichel, Phillip A. Low, and Joseph E. Parisi. Depletion of ventromedullary nk-1 receptor-immunoreactive neurons in multiple system atrophy. *Brain.*, 126:2183–2190, 2003.

[213] Gerald Baille et al. Early occurrence of inspiratory muscle weakness in parkinson's disease. *PloS one*, 13:e0190400, 2018.

[214] Yao Wang, Wei-bo Shao, Li Gao, Jie Lu, Hao Gu, Li-hua Sun, Yan Tan, and Ying-dong Zhang. Abnormal pulmonary function and respiratory muscle strength findings in chinese patients with parkinson's disease and multiple system atrophy–comparison with normal elderly. *PloS one*, 9:e116123, 2014.

[215] K. M. Torsney and D. Forsyth. Respiratory dysfunction in parkinson's disease. *J R Coll Physicians Edinb.*, 47:35–39, 2017.

[216] Michal Pokusa, Dagmar Hajduchova, Tomas Buday, and Anna Kralova Trancikova. Respiratory function and dysfunction in parkinson-type neurodegeneration. *Physiol Res.*, 69:S69–S79, 2020.

[217] Gerald Baille et al. Ventilatory dysfunction in parkinson's disease. *J Parkinsons Dis.*, 6:463–471, 2016.

[218] Leigh M. Seccombe et al. Abnormal ventilatory control in parkinson's disease–further evidence for non-motor dysfunction. *Respir Physiol Neurobiol.*, 179:300–304, 2011.

[219] Fadel Adib, Hongzi Mao, Zachary Kabelac, Dina Katabi, and Robert C Miller. Smart homes that monitor breathing and heart rate. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 837–846. ACM, 2015.

[220] Shichao Yue, Hao He, Hao Wang, Hariharan Rahul, and Dina Katabi. Extracting multi-person respiration from entangled rf signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2):86, 2018.

[221] Terri Blackwell et al. Associations of sleep architecture and sleep-disordered breathing and cognition in older community-dwelling men: the osteoporotic fractures in men sleep study. *J Am Geriatr Soc*, 59:2217–2225, 2011.

[222] Margaret M. Hoehn and Melvin D. Yahr. Parkinsonism: onset, progression and mortality. *Neurology*, 17:427–442, 1967.

[223] Tao Lei, Yu Zhang, Sida I Wang, Hui Dai, and Yoav Artzi. Simple recurrent units for highly parallelizable recurrence. In *EMNLP*, pages 4470–4481, 2018.

[224] R. Soikkeli, J. Partanen, H. Soininen, A. Pääkkönen, and P. Sr. Riekkinen. Slowing of eeg in parkinson's disease. *Electroencephalography and clinical neurophysiology*, 79:159–165, 1991.

[225] B. T. Klassen et al. Quantitative eeg as a predictive biomarker for parkinson disease dementia. *Neurology*, 77:118–124, 2011.

[226] D. H. Heck et al. Breathing as a fundamental rhythm of brain function. *Frontiers in neural circuits*, 10:115, 2017.

[227] Peter Welch. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15:70–73, 1967.

[228] H. Railo et al. Resting state eeg as a biomarker of parkinson's disease: Influence of measurement conditions. *Preprint at https://doi.org/10.1101/2020.05.08.084343*, 2020.

[229] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, pages 234–241, 2015.

[230] Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S Jaakkola, and Matt T Bianchi. Learning sleep stages from radio signals: A conditional adversarial architecture. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4100–4109. JMLR. org, 2017.

[231] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, volume 10, pages 61–74, 1999.

[232] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[233] Robert G. Newcombe. Two-sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in Medicine*, 17:857–872, 1998.

[234] Louis Guttman. A basis for analyzing test-retest reliability. *Psychometrika*, 10:255–282, 1945.

[235] H. Zahed et al. The neurophysiology of sleep in parkinson's disease. *Mov Disord.*, 36:1526–1542, 2021.

[236] José Enrique González-Naranjo et al. Analysis of sleep macrostructure in patients diagnosed with parkinson's disease. *Behav Sci. (Basel)*, 9, 2019.

[237] Allison W. Willis, Mario Schootman, Bradley A. Evanoff, Joel S. Perlmutter, and Brad A. Racette. Neurologist care in parkinson disease: a utilization, outcomes, and survival study. *Neurology*, 77:851–857, 2011.

[238] H. Braak et al. Staging of brain pathology related to sporadic parkinson's disease. *Neurobiol. Aging*, 24:197–211, 2003.

[239] T. A. Mestre et al. Parkinson's disease subtypes: Critical appraisal and recommendations. *J. Parkinsons. Dis.*, 11:395–404, 2021.

[240] Kenji Uchino, Makoto Shiraishi, Keita Tanaka, Masashi Akamatsu, and Yasuhiro Hasegawa. Impact of inability to turn in bed assessed by a wearable three-axis accelerometer on patients with parkinson's disease. *PloS one*, 12(11):e0187616, 2017.

[241] Claudia Gorecki, S José Closs, Jane Nixon, and Michelle Briggs. Patient-reported pressure ulcer pain: a mixed-methods systematic review. *Journal of pain and symptom management*, 42(3):443–459, 2011.

[242] Gustavo Desouzart, Rui Matos, Filipe Melo, and Ernesto Filgueiras. Effects of sleeping position on back pain in physically active seniors: A controlled pilot study. *Work*, 53(2):235–240, 2016.

[243] Akshay Menon and Manoj Kumar. Influence of body position on severity of obstructive sleep apnea: a systematic review. *ISRN otolaryngology*, 2013, 2013.

[244] Alister Mckenzie Neill, Susan Michelle Angus, Dimitar Sajkov, and Ronald Douglas Mcevoy. Effects of sleep posture on upper airway stability in patients with obstructive sleep apnea. *American journal of respiratory and critical care medicine*, 155(1):199–204, 1997.

[245] Terence Dwyer, A-LB Ponsonby, Neville M Newman, and Laura E Gibbons. Prospective cohort study of prone sleeping position and sudden infant death syndrome. *The Lancet*, 337(8752):1244–1247, 1991.

[246] Jennifer A Liebenthal, Shasha Wu, Sandra Rose, John S Ebersole, and James X Tao. Association of prone position with sudden unexpected death in epilepsy. *Neurology*, 84(7):703–709, 2015.

[247] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.

[248] Fadel Adib, Zachary Kabelac, and Dina Katabi. Multi-person localization via {RF} body reflections. In *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, pages 279–292, 2015.

[249] Hong Ji Lee, Su Hwan Hwang, Seung Min Lee, Yong Gyu Lim, and Kwang Suk Park. Estimation of body postures on bed using unconstrained ecg measurements. *IEEE journal of biomedical and health informatics*, 17(6):985–993, 2013.

[250] Heenam Yoon, Suhwan Hwang, Dawoon Jung, Sangho Choi, Kwangmin Joo, Jaewon Choi, Yujin Lee, Do-Un Jeong, and Kwangsuk Park. Estimation of sleep posture using a patch-type accelerometer based device. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4942–4945. IEEE, 2015.

[251] Kang-Ming Chang and Shin-Hong Liu. Wireless portable electrocardiogram and a tri-axis accelerometer implementation and application on sleep activity monitoring. *Telemedicine and e-Health*, 17(3):177–184, 2011.

[252] Jason J Liu, Wenyao Xu, Ming-Chun Huang, Nabil Alshurafa, Majid Sarrafzadeh, Nitin Raut, and Behrooz Yadegar. Sleep posture analysis using a dense pressure sensitive bedsheet. *Pervasive and Mobile Computing*, 10:34–50, 2014.

[253] Sina Akbarian, Ghazaleh Delfi, Kaiyin Zhu, Azadeh Yadollahi, and Babak Taati. Automated non-contact detection of head and body positions during sleep. *IEEE Access*, 2019.

[254] Shuangjun Liu and Sarah Ostadabbas. A vision-based system for in-bed posture tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1373–1382, 2017.

[255] Timo Grimm, Manuel Martinez, Andreas Benz, and Rainer Stiefelhagen. Sleep position classification from a depth camera using bed aligned maps. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 319–324. IEEE, 2016.

[256] Chi-Chun Hsia, Yu-Wei Hung, Yu-Hsien Chiu, and Chia-Hao Kang. Bayesian classification for bed posture detection based on kurtosis and skewness estimation. In *HealthCom 2008-10th International Conference on e-health Networking, Applications and Services*, pages 165–168. IEEE, 2008.

[257] Xiaowei Xu, Feng Lin, Aosen Wang, Chen Song, Yu Hu, and Wenyao Xu. On-bed sleep posture recognition based on body-earth mover's distance. In *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2015.

[258] M Baran Pouyan, Sarah Ostadabbas, Masoud Farshbaf, Rasoul Yousefi, Mehrdad Nourani, and MDM Pompeo. Continuous eight-posture classification for bed-bound patients. In *2013 6th International Conference on Biomedical Engineering and Informatics*, pages 121–126. IEEE, 2013.

[259] Sarah Ostadabbas, Maziyar Baran Pouyan, Mehrdad Nourani, and Nasser Kehtarnavaz. In-bed posture classification and limb identification. In *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, pages 133–136. IEEE, 2014.

[260] Jia Liu, Xingyu Chen, Shigang Chen, Xiulong Liu, Yanyan Wang, and Lijun Chen. Tagsheet: Sleeping posture recognition with an unobtrusive passive tag matrix. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 874–882. IEEE, 2019.

[261] Enamul Hoque, Robert F Dickerson, and John A Stankovic. Monitoring body positions and movements during sleep using wisps. In *Wireless Health 2010*, pages 44–53. ACM, 2010.

[262] Xuefeng Liu, Jiannong Cao, Shaojie Tang, and Jiaqi Wen. Wi-sleep: Contactless sleep monitoring via wifi signals. In *2014 IEEE Real-Time Systems Symposium*, pages 346–355. IEEE, 2014.

[263] Paolo Barsocchi. Position recognition to support bedsores prevention. *IEEE journal of biomedical and health informatics*, 17(1):53–59, 2012.

[264] Jian Liu, Yingying Chen, Yan Wang, Xu Chen, Jerry Cheng, and Jie Yang. Monitoring vital signs and postures during sleep using wifi signals. *IEEE Internet of Things Journal*, 5(3):2071–2084, 2018.

[265] Chen-Yu Hsu, Aayush Ahuja, Shichao Yue, Rumen Hristov, Zachary Kabelac, and Dina Katabi. Zero-effort in-home sleep and insomnia monitoring using radio signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1, 2017.

[266] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. Rf-based 3d skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 267–281, 2018.

[267] Chenshu Wu, Zheng Yang, Zimu Zhou, Xuefeng Liu, Yunhao Liu, and Jiannong Cao. Non-invasive detection of moving and stationary human with wifi. *IEEE Journal on Selected Areas in Communications*, 33(11):2329–2342, 2015.

[268] Teng Wei and Xinyu Zhang. mtrack: High-precision passive tracking using millimeter wave radios. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 117–129. ACM, 2015.

[269] Chen-Yu Hsu, Yuchen Liu, Zachary Kabelac, Rumen Hristov, Dina Katabi, and Christine Liu. Extracting gait velocity and stride length from surrounding radio signals. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2116–2126. ACM, 2017.

[270] Wei Wang, Alex X Liu, and Muhammad Shahzad. Gait recognition using wifi signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 363–373. ACM, 2016.

[271] Xuyu Wang, Chao Yang, and Shiwen Mao. Tensorbeat: Tensor decomposition for monitoring multiperson breathing beats with commodity wifi. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(1):8, 2017.

[272] Phuc Nguyen, Xinyu Zhang, Ann Halbower, and Tam Vu. Continuous and fine-grained breathing volume monitoring from afar using wireless signals. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.

[273] Xuefeng Liu, Jiannong Cao, Shaojie Tang, Jiaqi Wen, and Peng Guo. Contactless respiration monitoring via off-the-shelf wifi devices. *IEEE Transactions on Mobile Computing*, 15(10):2466–2479, 2015.

[274] Xuyu Wang, Chao Yang, and Shiwen Mao. Phasebeat: Exploiting csi phase data for vital sign monitoring with commodity wifi devices. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1230–1239. IEEE, 2017.

[275] Yonglong Tian, Guang-He Lee, Hao He, Chen-Yu Hsu, and Dina Katabi. Rf-based fall monitoring using convolutional neural networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):137, 2018.

[276] Yuxi Wang, Kaishun Wu, and Lionel M Ni. Wifall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing*, 16(2):581–594, 2016.

[277] Hao Wang, Daqing Zhang, Yasha Wang, Junyi Ma, Yuxiang Wang, and Shengjie Li. Rt-fall: A real-time and contactless fall detection system with commodity wifi devices. *IEEE Transactions on Mobile Computing*, 16(2):511–526, 2016.

[278] Tauhidur Rahman, Alexander T Adams, Ruth Vinisha Ravichandran, Mi Zhang, Shwetak N Patel, Julie A Kientz, and Tanzeem Choudhury. Dopplesleep: A contactless unobtrusive sleep sensing system using short-range doppler radar. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 39–50. ACM, 2015.

[279] Alexander Tataraidze, Lyudmila Korostovtseva, Lesya Anishchenko, Mikhail Bochkarev, Yurii Sviryaev, and Sergey Ivashov. Bioradiolocation-based sleep stage classification. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2839–2842. IEEE, 2016.

[280] Bassem R Mahafza. *Radar systems analysis and design using MATLAB*. Chapman and Hall/CRC, USA, 2005.

[281] Grace Gita Redhyka, Dika Setiawan, and Demi Soetraprawata. Embedded sensor fusion and moving-average filter for inertial measurement unit (imu) on the microcontroller-based stabilized platform. In *2015 International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*, pages 72–77. IEEE, 2015.

[282] Tianben Wang, Daqing Zhang, Yuanqing Zheng, Tao Gu, Xingshe Zhou, and Bernadette Dorizzi. C-fmcw based contactless respiration detection using acoustic signal. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):170, 2018.

[283] Zhenyuan Zhang, Zengshan Tian, and Mu Zhou. Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor. *IEEE Sensors Journal*, 18(8):3278–3289, 2018.

[284] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[285] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[286] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[287] Eivind Schjelderup Skarpsno, Paul Jarle Mork, Tom Ivar Lund Nilsen, and Andreas Holtermann. Sleep positions and nocturnal body movements based on free-living accelerometer recordings: association with demographics, lifestyle, and insomnia symptoms. *Nature and science of sleep*, 9:267, 2017.

[288] Joseph De Koninck, Dominique Lorrain, and Pierre Gagnon. Sleep positions and position shifts in five age groups: an ontogenetic picture. *Sleep*, 15(2):143–149, 1992.

[289] M. A. Ahmad, A. Patel, C. Eckert, V. Kumar, and A. Teredesai. Fairness in machine learning for healthcare. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3529–3530, 2020.

[290] Scott Mayer McKinney et al. International evaluation of an AI system for breast cancer screening. *Nature*, 577(7788):89–94, 2020.

[291] Laleh Seyyed-Kalantari, Haoran Zhang, Matthew BA McDermott, Irene Y Chen, and Marzyeh Ghassemi. Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations. *Nature medicine*, 27(12):2176–2182, 2021.

[292] A. S. Adamson and A. Smith. Machine learning and health care disparities in dermatology. *JAMA Dermatol.*, 154(11):1247–1248, 2018.

[293] J. Adleberg et al. Predicting patient demographics from chest radiographs with deep learning. *J. Am. Coll. Radiol.*, 19(10):1151–1161, 2022.

[294] Robert Geirhos et al. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673, 2020.

[295] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Med.*, 15(11):e1002683, 2018.

[296] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[297] A. Zawacki et al. Siim-acr pneumothorax segmentation, 2019. Available online: https://www.kaggle.com/c/siim-acr-pneumothorax-segmentation/.

[298] A. Bustos, A. Pertusa, J.-M. Salinas, and M. De La Iglesia-Vaya. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Med. Image Anal.*, 66:101797, 2020.

[299] H. Q. Nguyen et al. Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations. *Sci. Data*, 9(1):429, 2022.

[300] A. J. Larrazabal, N. Nieto, V. Peterson, D. H. Milone, and E. Ferrante. Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis. *Proc. Natl. Acad. Sci.*, 117(23):12592–12594, 2020.

[301] V. Rotemberg et al. A patient-centric dataset of images and metadata for identifying melanomas using clinical context. *Sci. Data*, 8(1), Jan 2021.

[302] Ocular disease recognition. Available online. Accessed: Nov. 02, 2023.

[303] Alistair E. W. Johnson et al. MIMIC-IV, a freely accessible electronic health record dataset. *Sci. Data*, 10(1), Jan 2023.

[304] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[305] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

[306] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 624–639, 2018.

[307] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[308] S. Barocas, M. Hardt, and A. Narayanan. *Fairness and machine learning: Limitations and opportunities*. fairmlbook.org, 2019.

[309] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[310] S. Pfohl, Y. Xu, A. Foryciarz, N. Ignatiadis, J. Genkins, and N. Shah. Net benefit, calibration, threshold selection, and training objectives for algorithmic fairness in healthcare. In *ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[311] F. Kuppers, J. Kronenberger, A. Shantia, and A. Haselhoff. Multivariate confidence calibration for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[312] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

[313] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(2), 2012.

[314] F. Pedregosa et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

[315] B. Glocker, C. Jones, M. Bernhardt, and S. Winzeck. Algorithmic encoding of protected characteristics in chest x-ray disease detection models. *Ebiomedicine*, 89, 2023.

[316] Susan Wei and Marc Niethammer. The fairness-accuracy pareto front. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 15(3):287–302, 2022.

[317] J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Leibniz International Proceedings in Informatics (LIPIcs)*, volume 67, pages 1–23, 2017.

[318] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[319] B. An, Z. Che, M. Ding, and F. Huang. Transferring fairness under distribution shifts via fair consistency regularization. In *Advances in Neural Information Processing Systems*, volume 35, pages 32582–32597, Dec 2022.

[320] Darshali A Vyas, Leo G Eisenstein, and David S Jones. Hidden in plain sight – reconsidering the use of race correction in clinical algorithms. *New England Journal of Medicine*, 383(9):874–882, 2020.

[321] Abhinav Kumar, Amit Deshpande, and Amit Sharma. Causal effect regularization: Automated detection and removal of spurious attributes. *arXiv preprint arXiv:2306.11072*, 2023.

[322] Anirban Basu. Use of race in clinical algorithms. *Science Advances*, 9(21):eadd2704, 2023.

[323] M. D. McCradden, S. Joshi, M. Mazwi, and J. A. Anderson. Ethical limitations of algorithmic fairness solutions in health care machine learning. *Lancet Digit. Health*, 2(5):e221–e223, 2020.

[324] H. Singh, R. Singh, V. Mhasawade, and R. Chunara. Fairness violations and mitigation under covariate shift. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 3–13, Mar 2021.

[325] The White House. Executive order on the safe, secure, and trustworthy development and use of artificial intelligence. Available online. Accessed: Nov. 15, 2023.

[326] M. Mitchell et al. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229, Atlanta GA USA, Jan 2019. ACM.

[327] Jack Gallifant, Emmett Alexander Kistler, Luis Filipe Nakayama, Chloe Zera, Sunil Kripalani, Adelline Ntatin, Leonor Fernandez, David Bates, Irene Dankwa-Mullan, and Leo Anthony Celi. Disparity dashboards: an evaluation of the literature and framework for health equity improvement. *The Lancet Digital Health*, 5(11):e831–e839, 2023.

[328] Haoran Zhang, Natalie Dullerud, Karsten Roth, Lauren Oakden-Rayner, Stephen Pfohl, and Marzyeh Ghassemi. Improving the fairness of chest x-ray classifiers. In *Conference on health, inference, and learning*, pages 204–233. PMLR, 2022.

[329] N. Martinez, M. Bertran, and G. Sapiro. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning*, pages 6755–6764. PMLR, 2020.

[330] Eike Petersen, Sune Holm, Melanie Ganz, and Aasa Feragen. The path toward equal performance in medical machine learning. *Patterns*, 4(7), 2023.

[331] J. K. Paulus and D. M. Kent. Predictably unequal: understanding and addressing concerns that algorithmic clinical prediction may increase health disparities. *Npj Digit. Med.*, 3(1), Jul 2020.

[332] Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. ME-Net: Towards effective adversarial robustness with matrix estimation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[333] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3681–3691, 2021.

[334] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[335] Xin Liu, Brian L Hill, Ziheng Jiang, Shwetak Patel, and Daniel McDuff. Efficientphys: Enabling simple, fast and accurate camera-based vitals measurement. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[336] John Gideon and Simon Stent. The way to my heart is through contrastive learning: Remote photoplethysmography from unlabelled video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3995–4004, 2021.

[337] Hao Wang, Euijoon Ahn, and Jinman Kim. Self-supervised representation learning framework for remote physiological measurement using spatiotemporal augmentation loss. *AAAI*, 2022.

[338] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[339] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[340] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv:1803.07640*, 2017.

[341] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014.

[342] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, 2019.

[343] Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean embedding of co-occurrence data. In *NeurIPS*, 2004.

[344] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[345] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[346] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[347] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[348] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Connecting language and vision using crowdsourced dense image annotations. *Visual genome*, 2016.

[349] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[350] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[351] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[352] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[353] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[354] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[355] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[356] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

[357] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019.

[358] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[359] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[360] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[361] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[362] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[363] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022.

[364] Raghav Mehta, Vítor Albiero, Li Chen, Ivan Evtimov, Tamar Glaser, Zhiheng Li, and Tal Hassner. You only need a good embeddings extractor to fix spurious correlations. *arXiv preprint arXiv:2212.06254*, 2022.