Abstract—By adhering to the dictum, “No causation without manipulation (treatment, intervention)”, cause and effect data analysis represents changes in observed data in terms of changes in the causal factors. When causal factors are not amenable for active manipulation in the real world due to current technological limitations or ethical considerations, a counterfactual approach performs an intervention on the model of data formation. In the case of object representation or activity (temporal object) representation, varying object parts is generally unfeasible whether they be spatial and/or temporal. Multilinear algebra, the algebra of higher order tensors, is a suitable and transparent framework for disentangling the causal factors of data formation. Learning a part-based intrinsic causal factor representations in a multilinear framework requires applying a set of interventions on a part-based multilinear model. We propose a unified multilinear model of wholes and parts. We derive a hierarchical block multilinear factorization, the $M$-mode Block SVD, that computes a disentangled representation of the causal factors by optimizing simultaneously across the entire object hierarchy. Given computational efficiency considerations, we introduce an incremental bottom-up computational alternative, the Incremental $M$-mode Block SVD, that employs the lower level abstractions, the part representations, to represent the higher level of abstractions, the parent wholes. This incremental computational approach may also be employed to update the causal model parameters when data becomes available incrementally. The resulting object representation is an interpretable combinatorial choice of intrinsic causal factor representations related to an objects recursive hierarchy of wholes and parts that renders object recognition robust to occlusion and reduces training data requirements.

Index Terms—causality, counterfactuals, explanatory variables, latent representation, factor analysis, multilinear algebra, $M$-mode SVD, block tensor factorization, hierarchical tensor hierarchical representation, object recognition, image analysis

I. INTRODUCTION: PROBLEM DEFINITION

Developing causal explanations for correct results or for failures from mathematical equations and data is important in developing a trustworthy artificial intelligence, and retaining public trust. Causal explanations are germane to the “right to an explanation” statute [15], [13] i.e., to data driven decisions, such as those that rely on images. Computer graphics and computer vision problems, also known as forward and inverse imaging problems, have been cast as causal inference questions [40], [42] consistent with Donald Rubin’s quantitative definition of causality, where “A causes B” means “the effect of A is B”, a measurable and experimentally repeatable quantity [14], [17]. Computer graphics may be viewed as addressing analogous questions to forward causal inferencing that addresses the “what if” question, and estimates the change in effects given a delta change in a causal factor. Computer vision may be viewed as addressing analogous questions to inverse causal inferencing that addresses the “why” question [12]. We define inverse causal inference as the estimation of causes given an estimated forward causal model and a set of observations that constrain the solution set.

Natural images are the composite consequence of multiple factors related to scene structure, illumination conditions, and imaging conditions. Multilinear algebra, the algebra of higher-order tensors, offers a potent mathematical framework for analyzing the multifactor structure of image ensembles and for addressing the difficult problem of disentangling the constituent factors, Fig. 2. (Vasilescu and Terzopoulos: TensorFaces 2002 [43], [44], MPCA and MICA 2005 [46], kernel variants [48], Multilinear Projection 2007/2011 [47], [41])

![Fig. 1: Data tensor, $D$, expressed in terms of a hierarchical data tensor, $D_h$, a mathematical instantiation of a tree data structure where $D = D_h \times L_1 \times \cdots \times L_m$, versus an independent bag of parts/sub-parts, or a data tensor with a reparameterized measurement mode in terms of regions and sub-regions. An object hierarchy may be based on adaptive quad/triangle based subdivision of various depths [38], or a set of perceptual parts of arbitrary shape, size and location. Images of non-articulated objects are best expressed with hierarchical data tensors that have a partially compositional form, where all the parts share the same extrinsic causal factor representations, Fig. 3b. Images of objects with articulated parts are best expressed in terms of hierarchical data tensors that are fully compositional in the causal factors, Fig. 3c. Images of non-articulated objects may also be represented by a fully compositional hierarchical data tensor, as depicted by the TensorTrinity example above.](image-url)
Scene structure is composed from a set of objects that appear to be formed from a recursive hierarchy of perceptual wholes and parts whose properties, such as shape, reflectance, and color, constitute a hierarchy of intrinsic causal factors of object appearance. Object appearance is the compositional consequence of both an object’s intrinsic causal factors, and extrinsic causal factors with the latter related to illumination (i.e. the location and types of light sources), and imaging (i.e. viewing direction, camera lens, rendering style etc.). Intrinsic and extrinsic causal factors confound each other’s contributions hindering recognition [42].

“Intrinsic properties are by virtue of the thing itself and nothing else” (David Lewis, 1983 [22]); whereas extrinsic properties are not entirely about that thing, but as a result of the way the thing interacts with the world. Unlike global intrinsic properties, local intrinsic properties are intrinsic to a part of the thing, and it may be said that a local intrinsic property is in an “intrinsic fashion”, or “intrinsically” about the thing, rather than “is intrinsic” to the thing [19].

Cause and effect analysis models the mechanisms of data formation, unlike conventional statistical analysis and conventional machine learning that model the distribution of the data [29]. Causal modeling from observational studies are suspect of bias and confounding with some exceptions [8], [34], unlike experimental studies [31], [32] in which a set of active interventions are applied, and their effect on response variables are measured and modeled. The differences between experimental studies, denoted symbolically with Judea Pearl’s do-operator [29], and observational studies are best exemplified by the following expectation and probability expressions

\[
E(d|c) \neq E(d|do(c)) \quad P(d|c) \neq P(d|do(c)),
\]

where \(d\) is a multivariate observation, and \(c\) is a hypothesized or actual causal factor. Pearl and Bareinboim [30], [2] have delineated the challenges of generalizing results from experimental studies to observational studies by parameterizing the error based on the possible error inducing sources.

The multilinear (tensor) structural equation approach is a suitable and transparent framework for disentangling the factors of data formation that has been employed in psychometrics [37], [16], [6], [3] econometrics [26], chemometrics [5], [1], signal processing [9], [10], [24], [27], computer vision [44], [11], [49], [50], computer graphics [39], [45], [48], [18], [25], [28], and machine learning [40], [46], [7], [36].

Adhering to the dictum, “No causation without manipulation (treatment, intervention)” [32], [20] each causal factor is varied one at a time while holding the rest fixed, and their effects on the response variables are measured and modeled by a data tensor model. The best evidence comes from randomized comparative studies. However, when causal factors are not amenable for manipulation due to current technological limitations or ethical considerations, a counterfactual approach is required. Rather than performing a manipulation in the real world, a counterfactual approach performs an intervention on the model.

In the case of object representation or activity (temporal object) representation, varying object parts is generally unfeasible whether they be spatial or temporal. Learning a hierarchy of causal factor representations requires applying a set of interventions on the structural model, hence it requires a part-based multilinear model, Fig 1.

This paper proposes a unified multilinear model of wholes and parts that defines a data tensor in terms of a hierarchical data tensor, \(D_w\), a mathematical instantiation of a tree data structure. Our hierarchical data tensor is a mathematical conceptual device that enables us to derive a hierarchical block multilinear factorization, an \(M\)-mode Block SVD, that optimizes simultaneously across the entire object hierarchy and allows for different tree parameterizations for the intrinsic versus the extrinsic causal factors. Given computational considerations, we develop an incremental computational alternative that employs the lower level abstractions, the part representations, to represent the higher level of abstractions, the parent wholes.

Our hierarchical block multilinear factorization, \(M\)-mode Block SVD, disentangles the causal structure by computing statistically invariant intrinsic and extrinsic representations. The factorization learns a hierarchy of low-level, mid-level and high-level features. Our hybrid approach mitigates the shortcomings of local features that are sensitive to local deformations and noise, and the shortcomings of global features that are sensitive to occlusions. The resulting object representation is a combinatorial choice of part representations, that renders object recognition robust to occlusion and reduces large training data requirements. This approach was employed for face verification by computing a set of causal explanations (causalX) [42].

II. RELEVANT TENSOR ALGEBRA

We will use standard textbook notation, denoting scalars by lower case italic letters \((a, b, \ldots)\), vectors by bold lower case letters \((\mathbf{a}, \mathbf{b}, \ldots)\), matrices by bold uppercase letters \((\mathbf{A}, \mathbf{B}, \ldots)\), and higher-order tensors by bold uppercase calligraphic letters \((\mathcal{A}, \mathcal{B}, \ldots)\). Index upper bounds are denoted by italic uppercase \((i.e., \; 1 \leq i \leq I)\). The zero matrix is denoted by \(0\), and the identity matrix is denoted by \(I\). References [21], [33] provide a quick tutorial, but references [40], [46], [41] are an indepth treatment of tensor based factor analysis.

Briefly, the natural generalization of matrices (i.e., linear operators defined over a vector space), tensors define multilinear operators over a set of vector spaces. A “data tensor” denotes an \(M\)-way data array.

**Definition 1 (Tensor):** Tensors are multilinear mappings over a set of vector spaces, \(\mathbb{R}^{I_1}, 1 \leq c \leq C\), to a range vector space \(\mathbb{R}^{I_0}\):

\[
\mathcal{A} : \{ \mathbb{R}^{I_1} \times \mathbb{R}^{I_2} \times \cdots \times \mathbb{R}^{I_C} \} \rightarrow \mathbb{R}^{I_0}. \tag{1}
\]

The order of tensor \(\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_C}\) is \(M = C + 1\). An element of \(\mathcal{A}\) is denoted as \(A_{i_1 \ldots i_C}\) or \(a_{i_1 \ldots i_C}\), where \(1 \leq i_0 \leq I_0\), and \(1 \leq i_c \leq I_c\).

The mode-\(m\) vectors of an \(M\)-order tensor \(\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}\) are the \(I_m\)-dimensional vectors obtained from \(\mathcal{A}\) by varying index \(i_m\) while keeping the other indices fixed. In tensor terminology, column vectors are the mode-1 vectors and row vectors are mode-2 vectors. The mode-\(m\) vectors of a tensor are also known as *fibers*. The mode-\(m\) vectors are the column vectors of matrix \(\mathbf{A}^{(m)}\) that results from *matrixizing* (a.k.a. *flattening*) the tensor \(\mathcal{A}\).
Definition 2 (Mode-m Matrizing): The mode-\( m \) matrizing of a tensor \( A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M} \) is defined as the matrix \( A_{[m]} \in \mathbb{R}^{I_m \times (I_1 \cdot I_2 \cdot \cdots \cdot I_{m-1} \cdot I_{m+1} \cdot \cdots \cdot I_M)} \). As the parentheses ordering indicates, the mode-\( m \) column vectors are arranged by sweeping all the other mode indices through their ranges, with smaller mode indices varying more rapidly than larger ones; thus,

\[
[A_{[m]}]_{jk} = a_{i_1 \cdots i_m \cdots i_M}, \quad \text{where} \quad j = i_m \quad \text{and} \quad k = 1 + \sum_{n \neq m}^{M} (i_n - 1) \prod_{l \neq m}^{M} I_l.
\]

A generalization of the product of two matrices is the product of a tensor and a matrix [9].

Definition 3 (Mode-\( m \) Product, \( \times m \)): The mode-\( m \) product of a tensor \( A \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M} \) and a matrix \( B \in \mathbb{R}^{I_m \times I} \), denoted by \( A \times_m B \), is a tensor of dimensionality \( \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{m-1} \times I_{m+1} \times \cdots \times I_M} \) whose entries are computed by

\[
[A \times_m B]_{i_1 \cdots i_m \cdots i_M} = \sum_{i_m} a_{i_1 \cdots i_m \cdots i_M} b_m, \quad C = A \times_m B, \quad C_{[m]} = BA_{[m]},
\]

The \( M \)-mode SVD (aka., the Tucker decomposition) is a “generalization” of the conventional matrix (i.e., 2-mode) SVD which may be written in tensor notation as

\[
D = U_1 S_{U_2^T} U_2 \Leftrightarrow D = S \times_1 U_1 \times_2 U_2.
\]

The \( M \)-mode SVD orthogonalizes the \( M \) spaces and decomposes the tensor as the mode-\( m \) product, denoted \( \times_m \), of \( M \)-orthonormal mode matrices, and a core tensor \( Z \)

\[
D = Z \times_1 U_1 \times_2 U_2 \cdots \times_m U_m \cdots \times_M U_M.
\]

III. Hierarchical Block Tensor Factorizations of \( D \)

Within the tensor mathematical framework, a \( M \)-way array or “data-tensor”, \( D \in \mathbb{R}^{I_0 \times I_1 \times \cdots \times I_M} \) contains a collection of vectorized and centered observations, \( d_{i_1 \cdots i_c \cdots i_M} \in \mathbb{R}^{I_0} \) that are the result of \( C \) causal factors. The \( c \) causal factor (\( 1 \leq c \leq C \)) takes one of \( I_c \) values that are indexed by \( i_c \), \( 1 \leq i_c \leq I_c \). An observation that is result of the confluence \( C \) causal factors is modeled by a multilinear structural equation with multimode latent variables, \( r_i \), that represent the causal factors

\[
d_{i_1 \cdots i_c \cdots i_M} = \sum_{c=1}^{C} d_{c(i_1 \cdots i_c \cdots i_M)} = r_{i_1} \cdots r_{i_c} \cdots r_{i_M} + \epsilon_{i_1 \cdots i_c \cdots i_M},
\]

where \( T = Z \times_1 U_1 \times_2 U_2 \cdots \times_m U_m \cdots \times_M U_M \).

A. Hierarchical Data Tensor, \( D_{H} \)

We identify a general base case object and two special cases. A base case object may be composed of (i) two partially overlapping children-parts and parent-whole that has data not contained in any of the children-parts, (ii) a set of non-overlapping parts, or (iii) a set of fully overlapping parts. The tensor representation of an object with fully overlapping parts, Fig. 3(e), resembles the rank-(\( L, M, N \)) or a rank-(\( L, M, \cdot \)) block tensor decomposition [10].

The data wholes and parts are extracted by employing a filter bank \( \{H_s \in \mathbb{R}^{H_0 \times I_0} \mid \sum_{s=1}^{S} H_s = I \} \) where a 1D (2D or 3D) convolutional filter, \( H_s \), is written as a circulant matrix (dubly or triply circulant matrix), \( H_s \), and \( s \) is a segment index. The convolution may be written as a matrix-vector multiplication or the mode-\( m \) product, \( \times_m \), between a circulant matrix, \( H_s \), and a vectorized observation. For example, if an observation is returned by the capture device as a 2-way

\[1\]Reference [40, Appendix A] evaluates some of the arguments found in highly cited publications in favor of treating an image as a matrix rather than vectorizing it. While technically speaking it is not incorrect to treat an image as a matrix, the evaluation concludes that most arguments do not stand up to analytical scrutiny, and it is preferable to vectorize images.

\[2\]The block tensor decomposition [10] goal is to find the best fitting \( K \) fully overlapping tensor blocks that are all multilinearly decomposable into the same multilinear rank-(\( R_1, R_2, R_3 \)). This is analogous to finding the best fitting K rank-1 terms (also known as rank-(1, 1, 1)) computed by the CP-algorithm.
Fig. 3: The data tensor, \( \mathcal{D} \), written in terms of a hierarchical data tensor, \( \mathcal{D}_H \). (a) When \( \mathcal{D}_H \) contains the data tensor segments, \( \mathcal{D}_s \), along its super-diagonal then \( \mathcal{D}_H \) has a fully compositional form, and every mode matrix has a compositional representation. (b) A general base case object written in a partially compositional form with a compositional representation for only one mode matrix (causal factor). (c) A general base case object where all the causal factors have a compositional representation. The tensor \( \mathcal{D}_s \) is fully compositional in the causal factors. (d) A base case object with non-overlapping parts. All the causal factors have a compositional representation. Multilinear factorizations are block independent. (e) Base case object with completely overlapping parts where all the causal factors have compositional representation. Objects with non-overlapping or completely overlapping parts may also be written using a partially compositional form analogously to (b).

Multivariate array, \( \mathbf{D} \in \mathbb{R}^{I_{x} \times I_{c}} \), with \( I_{x} \) rows and \( I_{c} \) columns, the convolution is written as

\[
\mathbf{D} = \mathbf{D} \ast \mathbf{h}(x, y)
\]

where the measurement mode is mode 0. In practice, a convolution is efficiently implemented using a DFFT. The segment data tensor, \( \mathcal{D}_s = \mathcal{D} \times_x \mathbf{H}_s \), is the result of multiplying (convolving) every observation, \( \mathbf{d} \), with the block circulant matrix (filter), \( \mathbf{H}_s \). A filter \( \mathbf{H}_s \) may be of any type, and have any spatial scope. When a filter matrix is a block identity matrix, \( \mathbf{H}_s = \mathbf{I}_s \), the filter matrix multiplication with a vectorized observation has the effect of segmenting a portion of the data. Measurements associated with perceptual parts may not be tightly packed into a block apriori, as in the case of vectorized images, but chunking is achieved by a trivial permutation. A data tensor is expressed as a recursive hierarchy of wholes and parts by defining and employing a hierarchical data tensor, \( \mathcal{D}_H \). When a data tensor contains along its super-diagonal the data tensor segments, \( \mathcal{D}_s \), then \( \mathcal{D}_H \) has a fully compositional form, and all the data tensor modes have a compositional representation, Fig. 3(a). The data tensor segments, \( \mathcal{D}_s \), may be sparse and represent local parts, or may be full and correspond to a filtered version of a parent-whole, as in the case of a Laplacian pyramid. Mathematically writing \( \mathcal{D} \) in terms of \( \mathcal{D}_H \) is expressed with
data tensor, where a single factor has a compositional form, the intrinsic object representation, Fig. 3(b). Thus, the $\mathcal{D}_N$ is multiplied through by all the $I_n$ except one. Each multiplied $I_n$ is replaced by a single place holder identity matrix in the model.

The three different ways of rewriting $\mathcal{D}$ in terms of a hierarchy of wholes and parts, eq. 7-9, results in three mathematically equivalent representations,\(^3\) based on factorizing $\mathcal{D}$, $\mathcal{D}^\perp$, and $\mathcal{D}^\parallel$:

\[
\mathcal{D} = \sum_{s=1}^{S} (Z_{xs} U_{0x} \times U_{1x} \cdots \times U_{cx} X_c) x_s H_s \tag{10}
\]

\[
= \sum_{s=1}^{S} (Z_{xs} U_{0x} \times U_{1x} \cdots \times U_{cx} \cdots \times U_{cx} X_c) \tag{11}
\]

\[
= (Z_{x0} U_{0x} x_0 \times U_{1x}(1) \cdots \times U_{cx}(1) \cdots \times U_{cx}(1)) \mathcal{D}_N \times I_{nx} x_1 \cdots x_c I_{nx} \tag{12}
\]

Despite the prior mathematical equivalence, equations 7,10, and equations 8,12 are not flexible enough to explicitly indicate if the parts are organized in a partially compositional form, or a fully compositional form.

The expression of $\mathcal{D}$ in terms of a hierarchical data tensor is a mathematical conceptual device, that enables a unified mathematical model of wholes and parts that can be expressed completely as a mode-m product (tensor-matrix multiplication) and whose factorization can be optimized in a principled manner.

Dimensionality reduction of the compositional representation is performed by optimizing

\[
e = ||\mathcal{D} - (Z_{x0} U_{0x} x_0 \times U_{1x}(1) \cdots \times U_{cx}(1)) x_s I_{nx} \cdots x_c I_{nx}||^2 + \sum_{l=0}^{C} \lambda_l ||U_{lx}^T U_{lx} - I||^2 \tag{13}
\]

where $U_{lx}$ is the composite representation of the $l$th mode, and $Z_{x0}$ governs the interaction between causal factors. Our optimization may be initialized by setting $Z_{x0}$ and $U_{lx}$ to the M-mode SVD of $\mathcal{D}_N$,\(^5\) and performing dimensionality reduction through truncation, where $U_{lx} \in \mathbb{R}^{Sl \times J_x}$, $Z_{x0} \in \mathbb{R}^{J_0 \cdots J_L \cdots J_C}$ and $\hat{J}_l \leq S_l$.

B. Derivation

For notational simplicity, we re-write the loss function as,

\[
e := ||\mathcal{D} - \tilde{Z}_{x0} U_{0x} x_0 \times U_{1x}(1) \cdots \times U_{cx}(1)||^2 + \sum_{c=0}^{C} \lambda_c ||U_{cx}^T U_{cx} - I|| \tag{14}
\]

whose $U_{cx}$ sub-matrices are then subject to orthonormality constraints.

Solving for the optimal core tensor, $\tilde{Z}_{x0}$, the data tensor, $\mathcal{D}$, approximation is expressed in vector form as,

\[
e = ||\text{vec}(\mathcal{D}) - (\tilde{U}_{cx} \cdots \times \tilde{U}_{cx} \cdots \times \tilde{U}_{cx}) \text{vec}(\tilde{Z}_{x0})|| \tag{19}
\]

\[
\text{vec}(\mathcal{D}) = \sum_{c=0}^{C} \lambda_c ||U_{cx}^T U_{cx} - I|| \tag{20}
\]

6 The Kronecker product of $U \in \mathbb{R}^{J \times J}$ and $V \in \mathbb{R}^{K \times L}$ is the $IK \times JL$ matrix defined as $[U \otimes V]_{ij,kl}=u_{ij}v_{kl}$.

7 The Khatri-Rao product of $\{U_1, \ldots, U_n\} \odot \{V_1, \ldots, V_n\}$ with $U_i \in \mathbb{R}^{J_i \times N}$ and $V_j \in \mathbb{R}^{K_j \times N}$ is a block-matrix Kronecker product; therefore, it can be expressed as $U \odot V = \{U_1 \odot V_1, \ldots, (U_i \odot V_j), \ldots, (U_n \odot V_n)\}$.\(^3\)

\footnotesize
\(^3\)Equivalent representations can be transformed into one another by post-multiplying mode matrices with nonsingular matrices, $G$, $\mathcal{D}' = \{Z_{xs} G_x \times G_x \times \cdots \times G_x \times G_x G_c\} x_0 I_{nx} x_0 \times G_x G_c I_{nx} x_1 \cdots x_c I_{nx}$.\(^4\)

\(^4\)Note that eq.13 does not reduce to a multilinear subspace decomposition of $\mathcal{D}_N$ since $\mathcal{D} \times I_{nx} x_1 \times I_{nx} x_2 \cdots \times \times I_{nx} x_c \mathcal{D}_N \neq \mathcal{D}_N$.

\(^5\)For computational efficiency, we may perform M-mode SVD on each data tensor segment $\mathcal{D}_N$ and concatenate terms along the diagonal of $\mathcal{Z}_{x0}$ and $U_{lx}$.\(^6\)
Algorithm 1 M-mode Block SVD.

Input: Data tensor, \( \mathcal{D} \in \mathbb{R}^{k_1 \times k_2 \times \cdots k_c} \), filters \( H_i \), and desired dimensionality reduction \( f_1, \ldots, f_c \).

1. Initialization.
   1a. Decompose each data tensor segment, \( \mathcal{D}_i = \mathcal{D} \times H_i \), by employing the M-mode SVD.
   \[ \mathcal{D}_i = \mathcal{Z} \times_1 u_{1i} \times_2 u_{2i} \times_3 u_{3i} \times_4 u_{4i} \]
   1b. For \( \epsilon = 0, 1, \ldots, c \), set \( \mathcal{U}_i = [u_{1i}, \ldots, u_{4i}, \ldots, u_{ci}] \), and truncate \( f_i \) columns by sorting all the eigenvalues from all data segments and deleting the columns corresponding to the lowest eigenvalues from various \( u_{4i} \) and the rows from \( Z_{\delta_{0i}} \).

2. Optimization via alternating least squares.
   Iterate for \( n = 1, \ldots, N \)
   For \( \epsilon = 0, 1, \ldots, c \),
   2a. Compute mode matrix \( U_{\epsilon i} \) while holding the rest fixed.
   \[ Z_{\delta_{0i}} \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \ddots & \vdots \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} \]
   \[ U_{\epsilon i} = D_{\epsilon i}[U_{\epsilon 1i} \cdots U_{\epsilon 4i}]^{T} \]

Set \( \hat{U}_{\epsilon i} \) to the leading left-singular vectors of the SVD of \( \mathcal{Z}_{\delta_{0i}} \), a subset of the columns in \( U_{\epsilon i} \). Update \( U_{\epsilon i} \) in \( U_{\epsilon i} \) with \( \hat{U}_{\epsilon i} \).

2b. Set the non-zero(nz) terms of \( \mathcal{X}_n \) based on:
   \[ \text{vec}(Z_{\delta_{0i}}) = (\text{vec}(U_{\epsilon 1i} \cdots U_{\epsilon 4i} \cdots U_{\epsilon ni})\text{vec}(\mathcal{D}) \]

until convergence.\(^6\)

Output converged matrices \( U_{1i}, \ldots, U_{ci} \) and tensor \( Z_{\delta_{0i}} \).

The complexity of computing the SVD of an \( m \times n \) matrix \( A \) is \( O(mn \min(m,n)) \), which is costly when both \( m \) and \( n \) are large. However, we can efficiently compute the \( \hat{R} \) leading left-singular vectors of \( A \) by first computing the rank-\( \hat{R} \) modified Gram Schmidt (MGS) orthogonal decomposition \( A = QR \), where \( R \) is \( \hat{R} \times \hat{R} \) and \( R \) is \( \hat{R} \times n \), and then computing the SVD of \( R \) and multiplying it as follows:

\[ A \approx QR = (Q \Sigma) \Sigma \Gamma = \text{SVD} \]

Note that the \( \hat{R} \) is a pre-specified maximum number of iterations. A possible convergence criterion is to compute at each iteration the approximation error \( \epsilon = \|D - \hat{D}\|^{2} \) and test if \( \epsilon < \epsilon_{T} \) for sufficiently small tolerance \( \epsilon_{T} \).

Solve for the non-zero(nz) terms of \( Z_{\delta_{0i}} \) in the equation \( \partial e / \partial (Z_{\delta_{0i}}) = 0 \), by removing the corresponding zero columns of the first rank matrix on the right side of the equation below, performing the pseudo-inverse, and setting

\[ \text{vec}(Z_{\delta_{0i}}) = (U_{\epsilon 1i} \cdots U_{\epsilon 4i} \cdots U_{\epsilon ni})\text{vec}(\mathcal{D}). \]

Repeat all steps until convergence. This optimization is the basis of the M-mode Block SVD Algorithm 1.

\[ D_{\epsilon i} = [D_{1i} \cdots D_{ki} \cdots D_{ki}] \]

\[ = \begin{bmatrix} U_{1,1} \Sigma_{1,1} T_{1,1} (U_{1,1} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4})^{T} & \ldots & U_{1,4} \Sigma_{1,k} T_{1,i} (U_{1,1} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4})^{T} \end{bmatrix} \]

\[ = \begin{bmatrix} U_{1,1} \Sigma_{1,1} \ldots U_{1,4} \Sigma_{1,k} \ldots U_{1,1} \Sigma_{1,k} \end{bmatrix} \]

\[ \text{QR} + \text{SVD of R} \]

\[ T_{1,i} (U_{1,1} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4})^{T} \]

\[ = U_{1,1} \Sigma_{1,1} \begin{bmatrix} V_{1,1}^{T} \ldots V_{1,4}^{T} \ldots V_{1,4}^{T} \end{bmatrix} \]

When the data tensor is a collection of observations made up of non-overlapping parts, Fig. 3d, the data tensor decomposition reduces to the concatenation of an M-mode SVD of individual parts and when the data tensor is a collection of overlapping parts that have the same multilinear-rank reduction, Fig. 3e, see[42] for additional specific optimizations.

IV. REPRESENTING LEVELS OF ABSTRACTION BOTTOM-UP

An incremental hierarchical block multilinear factorization that represents levels of abstractions bottom-up is developed analogously to the incremental SVD for matrices [4]. The precomputed multilinear factorizations of the children parts are employed to determine the parent whole multilinear factorization. The derived algorithm may also be employed to update the overall model when the data becomes available sequentially [23]. We first address the computation of the mode matrices and the extended core of the parent whole when the children parts are non-overlapping. Next, we consider the overlapping children case, and the case where the parent-wholes and children-parts contain differently filtered data.

Computing parent causal mode matrices, \( U_{c,w} \):

Note that the parent whole, \( D_{c,w} \), is a concatenation of the data contained in its \( K' \) children segments that are part of the hierarchy, \( D_{k} \), where \( 1 \leq k \leq K' \). New data that is not contained by any of the children is denoted as the \( K = K' + 1 \) child, \( D_{K} \), eq. 21.

We initialize the hierarchical block multilinear factorization by performing an \( M \)-mode SVD on each leaf.

The \( c \) mode matrix, \( U_{c,w} \), of the \( w \) parent whole, \( D_{c,w} \), is the left singular matrix of \( \begin{bmatrix} U_{c,1} \Sigma_{c,1} & \ldots & U_{c,4} \Sigma_{c,4} \end{bmatrix} \]

which is based on the following derivation, that writes SVD of the flattened parent whole in terms of the SVDs of its flattened children parts, followed by a collection terms such that \( V_{c,all} \) is a block diagonal matrix of \( V_{c,k} \):

\[ D_{c,w} = [D_{c,1} \cdots D_{c,k} \cdots D_{c,K}] \]

\[ = \begin{bmatrix} U_{c,1} \Sigma_{c,1} T_{c,1} (U_{1,1} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4})^{T} & \ldots & U_{c,4} \Sigma_{c,4} T_{c,4} (U_{1,1} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4})^{T} \end{bmatrix} \]

\[ = \begin{bmatrix} U_{c,1} \Sigma_{c,1} \ldots U_{c,4} \Sigma_{c,4} \end{bmatrix} \]

\[ \text{QR} + \text{SVD of R} \]

\[ T_{c,1} (U_{1,1} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4}, U_{1,4} \otimes \ldots U_{1,4})^{T} \]

\[ = U_{c,1} \Sigma_{c,1} \begin{bmatrix} V_{1,1}^{T} \ldots V_{1,4}^{T} \ldots V_{1,4}^{T} \end{bmatrix} \]

\[ V_{c,k} \]

Computing the parent extended core, \( T_{c} \):

Computation of the extended core associated with the parent whole, \( T_{c} \), is performed by considering the following derivation.
Algorithm 2 Incremental M-Mode Block SVD.

Input the data tensor $\mathcal{D} \in \mathbb{R}^{I \times J \times K}$ and data tensor tree parameterization.

1. For $c = 0, \ldots, C$, in a bottom-up fashion
   a. For each data tensor leaf nodes:
      Set mode matrix $U_{c}$ to the left matrix of the SVD of $\mathcal{D}_{Ic}^e \mathcal{D}_{Ic}^e = U_{c} \Sigma_{c}^e U_{c}^T$.
   b. In a bottom-up fashion, for each data tensor that is a parent-whole:
      Set the mode matrix $U_{c}$ of the parent-whole by computing the SVD
      $U_{c}, \Sigma_{c}, \Sigma_{c}^e = \text{SVD}(\mathcal{D}_{Ic}^e, \mathcal{D}_{Ic}^e, \ldots, \mathcal{U}_{c}, \Sigma_{c}, \Sigma_{c}^e, \mathcal{U}_{c}, \Sigma_{c}, \Sigma_{c}^e)$.

2. Compute the normalized core of the $k$ child $T_k = T_k \times \Sigma_{1}^{-1} \times \Sigma_{2}^{-1} \times \ldots \times \Sigma_{c}^{-1}$.

3. Let $T_{ext}$ be a tensor that contains the children normalized cores $\tilde{Z}_{k}$ along its super-diagonal.

4. Set the core of the parent-whole as $T_c = T_{ext} \times \Sigma_{1}^{-1} \times \Sigma_{2}^{-1} \times \ldots \times \Sigma_{c}^{-1}$.

Output mode matrices $U_{1}, \ldots, U_{C}$ and the core tensor $T$ of the final parent-whole.

where $\Sigma_{k} = \Sigma_{k} \Sigma_{k}^e$ is the normalized extended core of the $k$th child, and $T_{ext}$ contains along the diagonal the children normalized extended cores $\tilde{T}_k$. Thus, the extended core of the parent whole is

$$T_c = T_{ext} \times \Sigma_{c}^{-1} \times \ldots \times \Sigma_{1}^{-1} \times \Sigma_{c}^{-1}$$

### Overlapping children:

This case may be reduced to the non-overlapping case by introducing another level in the hierarchy. Overlapping children are now treated as parents with one non-overlapping child sub-part and child sub-parts that correspond to every possible combination of overlaps that are shared by siblings. The original parent whole representation is computed in terms of the grandchildren representations.

### Parent-whole and children-parts with differently filtered data:

This is the case when a parent-whole and the child parts contain differently filtered information, as in the case when a parent-whole and the children parts sample information from different layers of a Laplacian pyramid. This case may be reduced to a non-overlapping case by writing the filters as the product between a segmentation filter, $S$, and a general filter that post multiplies the segmentation, $H = F.S$, and $D_c = (D \times S)^c \times F$. The general filters, $F$, may be applied after the cores are computed.

### Computational Cost Analysis:

Let an $M$-order data tensor, $\mathcal{D} \in \mathbb{R}^{I \times J \times K}$, where $M = C + 1$, be recursively subdivided into $K = 2^M$ children of the same order, but with each mode half in size. There are a total of $\log_{2}K N + 1$ levels, where $N = \prod_{i=0}^{C} L_i$. Recursive subdivision results in $S = N \log_{2}N \log_{2}N + 1$ segments. The total computational cost is the amortized M-mode SVD cost per data tensor segment, $T$, times the number of segments, $O(TN \log_{2}N)$. Since siblings at each level can be computed independently, on a distributed system the cost is $O(T \log_{2}N)$.

### V. CAUSALX EXPERIMENTS

CausalX visual recognition system computes a set of causal explanations based on a counterfactual causal model that takes advantage of the assets of multilinear (tensor) algebra. The M-mode Block SVD and the Incremental M-mode Block SVD algorithms estimate the model parameters. In the context of face image verification, we compute a compositional hierarchical person representation [42]. Our system is trained on a set of observations that are the result of combinatorially manipulating the scene structure, the viewing and illumination conditions.

We rendered in Maya images of 100 people from 15 different viewpoints with 15 different illuminations. The collection of vectorized images with 10,414 pixels is organized in a data tensor, $\mathcal{D} \in \mathbb{R}^{10,414 \times 15 \times 15 \times 100}$. The counterfactual model is employed by estimating $\mathcal{D}_a$, a hierarchical tensor of part-based Laplacian pyramids. We report encouraging face verification results on two test data sets the Freiburg, and the Labeled Faces in the Wild (LFW) datasets. We have currently achieved verification rates just shy of 80% on LFW [42], by employing less than one percent (1%) of the total images employed by DeepFace [35]. When data is limited, convolutional neural networks (CNNs) do not converge or generalize. More importantly, CNNs are predictive rather than causal models.

### Conclusion

This paper deepens the definition of causality in a multilinear (tensor) framework by addressing the distinctions between intrinsic versus extrinsic causality, and local versus global causality. It proposes a unified multilinear model of wholes and parts that reconceptualizes a data tensor in terms of a hierarchical data tensor. Our hierarchical data tensor is a mathematical instantiation of a tree data structure that enables a single elegant model of wholes and parts and allows for different tree parameterizations for the intrinsic versus extrinsic causal factors. The derived tensor factorization is a hierarchical block multilinear factorization that disentangles the causal structure of data formation. Given computational efficiency considerations, we present an incremental computational alternative that employs the part representations from the lower levels of abstraction to compute the parent whole representations from the higher levels of abstraction in an iterative bottom-up way. This computational approach may be employed to update causal representations in scenarios when data is available incrementally. The resulting object representation is a combinatorial choice of part representations, that renders object recognition robust to occlusion and reduces large training data requirements. We have demonstrated our work in the context of face verification by extending the TensorFaces method with promising results. TensorFaces is a component of CausalX, a counterfactual causal based visual recognition system, and an explainable AI.

### Acknowledgement

The authors are thankful to Ernest Davis for feedback provided during the writing of this document, and to Donald Rubin and Andrew Gelman for helpful discussions.