



Computer
Science



Symbolic Reasoning in the Age of Large Language Models

Guy Van den Broeck



Reasoning with Symbolic AI

- Logic and probabilistic
- **Deductive** reasoning algorithms
- Correct on *all* problems
- Limited scope
- Intractable



30 years



Reasoning with Transformers

- Build chains of thought
- **Inductive** reasoning from data
- Correct on *many* problems
- Unlimited scope
- Tractable

Use the digits [0-9] to make three numbers: x,y,z so that $x+y=z$

Thought for 270 seconds ^

Okay, so I need to use each digit from 0 to 9 exactly once to create three numbers x, y, and z such that $x + y = z$.
Hmm, let me think about how to approach this.

...

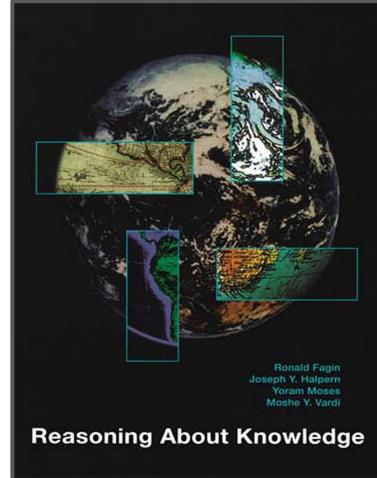
Verification:

- The sum $752 + 346 = 1098$ is correct.
- All digits from 0 to 9 are used exactly once.

Thus, the final answer is:

$752 + 346 = 1098$

<https://www.datacamp.com/blog/deepseek-r1-vs-v3>



Questions for this talk:



1. Do deductive reasoning algorithms still have a purpose in the age of LLMs?
2. Where did reasoning algorithms go wrong? What should they look like today?
3. Can reasoning algorithms provide a path to language model alignment, safety?

Questions for this talk:



- 1. Do deductive reasoning algorithms still have a purpose in the age of LLMs?**
- 2. Where did reasoning algorithms go wrong? What should they look like today?**
- 3. Can reasoning algorithms provide a path to language model alignment, safety?**



Generate a sentence using "frisbee", "caught", and "dog" in that order



The **frisbee** flew through the air and the **dog** **caught** it with excitement.

GPT 4.1-mini (apr 2025)



The energetic **dog** leapt into the air, **caught** the **frisbee** in mid-flight, and proudly trotted back to its owner.

Llama-3.2-3B (sep 2024)



You got the order wrong, please respect the given order



Here is a new sentence with the words in that order: The **dog**'s owner was about to **catch** the **frisbee** that had been flying through the air.



A **frisbee** is **caught** by a **dog**.

A pair of **frisbee** players are **caught** in a **dog** fight.

Ctrl-G (jun 2024)

Reasoning about all Future Tokens: *Constraints*

$p(\text{next-token} \mid \alpha, \text{prefix})$

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} = \text{"the weather is"})$

Lexical Constraint α : sentence contains keyword "winter"

Reasoning about all Future Tokens: *Constraints*

$p(\text{next-token} \mid \alpha, \text{prefix})$

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} = \text{"the weather is"})$

Lexical Constraint α : sentence contains keyword "winter"

$\propto p(\text{next-token} \mid \text{prefix})$

$\cdot p(\alpha \mid \text{next-token, prefix})$



Bayes' rule lets us reason backwards in time!

Reasoning about all Future Tokens: *Constraints*

$p(\text{next-token} \mid \alpha, \text{prefix})$

cold	0.025
warm	0.001

$\propto p(\text{next-token} \mid \text{prefix})$

cold	0.05
warm	0.10

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} = \text{"the weather is"})$

Lexical Constraint α : sentence contains keyword "winter"

$\cdot p(\alpha \mid \text{next-token, prefix})$

cold	0.50
warm	0.01



Reasoning about all Future Tokens: *Alignment*

$p(\text{next-token} \mid \alpha, \text{prefix})$

Prefix: It's a pain ...

Constraint α : non-toxic

Reasoning about all Future Tokens: *Alignment*

$p(\text{next-token} \mid \alpha, \text{prefix})$

Prefix: It's a pain ...

Constraint α : non-toxic

$\propto p(\text{next-token} \mid \text{prefix})$

$\cdot p(\alpha \mid \text{next-token, prefix})$

in	0.3
to	0.1

the ass	0.3
the butt	0.15
the neck	0.05
deal with	0.2
handle	0.1
...	...



Reasoning about all Future Tokens: *Alignment*

$p(\text{next-token} \mid \alpha, \text{prefix})$

in	0.03
to	0.08

Prefix: It's a pain ...

Constraint α : non-toxic

$\propto p(\text{next-token} \mid \text{prefix})$

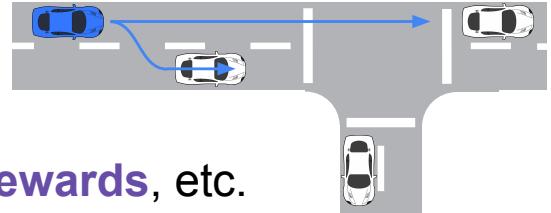
$\cdot p(\alpha \mid \text{next-token, prefix})$

in	0.3
to	0.1

in	0.1
to	0.8



Reasoning about all Future Tokens: *Offline RL*

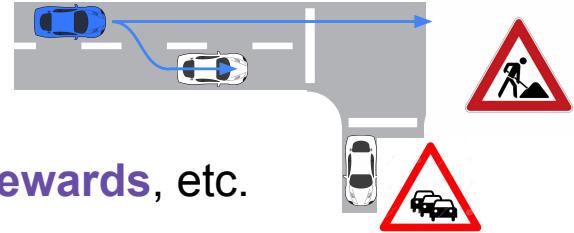


Training: model the joint distribution over **states**, **actions**, **rewards**, etc.

Inference: sample next **states** and **actions**

... state_{t-1} action_{t-1} R_{t-1} state_t action_t R_t state_{t+1} action_{t+1} R_{t+1} ...

Reasoning about all Future Tokens: *Offline RL*



Training: model the joint distribution over **states**, **actions**, **rewards**, etc.

Inference: sample next **states** and **actions**, as well as **constraints**.



$$p(\text{action} | \alpha, \text{prefix}) \propto p(\text{action} | \text{prefix}) \cdot p(\alpha | \text{action}, \text{prefix})$$

Reasoning about all Future Tokens

$$p_{lm}(\text{next-token} \mid \alpha, \text{prefix})$$

$$\propto p_{lm}(\text{next-token} \mid \text{prefix}) \cdot p_{lm}(\alpha \mid \text{next-token, prefix})$$

Using Bayes rule,



Intractable



Looking 20 tokens into the future amounts to more sentences than atoms in the universe....

Reasoning about all Future Tokens

$p_{lm}(\text{next-token} \mid \alpha, \text{prefix})$

Abusing Bayes rule,

$\propto p_{lm}(\text{next-token} \mid \text{prefix}) \cdot p_{\text{circuit}}(\alpha \mid \text{next-token, prefix})$



Use a tractable circuit model distilled from the transformer LLM...

A ‘digital twin’ that can do symbolic reasoning

Reasoning about all Future Tokens: *Constraints*

$p(\text{next-token} \mid \alpha, \text{prefix})$

cold	0.025
warm	0.001

$\propto p(\text{next-token} \mid \text{prefix})$

cold	0.05
warm	0.10

Constrained Generation: $\Pr(x_{t+1} \mid \alpha, x_{1:t} = \text{"the weather is"})$

Lexical Constraint α : sentence contains keyword "winter"

$\cdot p(\alpha \mid \text{next-token, prefix})$

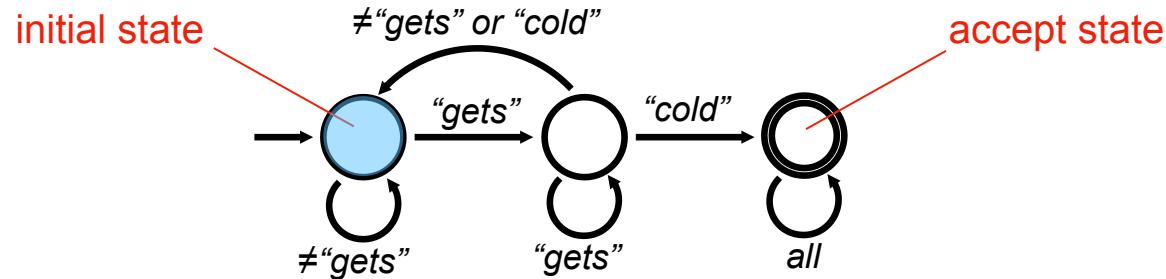
cold	0.50
warm	0.01



Representing Logical Constraints

as a **deterministic finite automaton (DFA)**

Example. Check if a string contains “gets cold”.

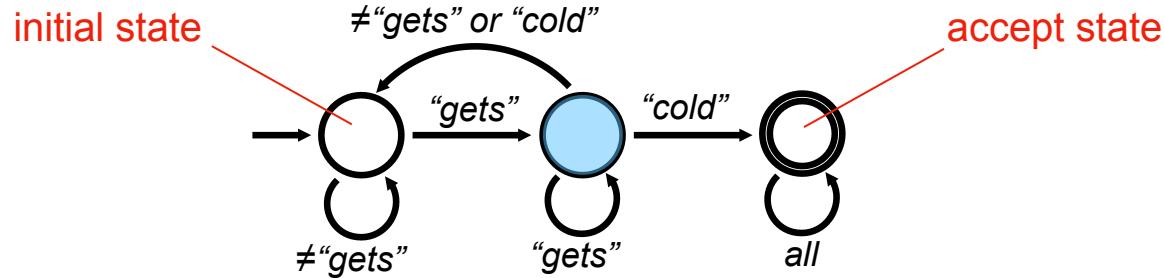


String: “The weather gets cold in the winter.”

Representing Logical Constraints

as a **deterministic finite automaton (DFA)**

Example. Check if a string contains “gets cold”.

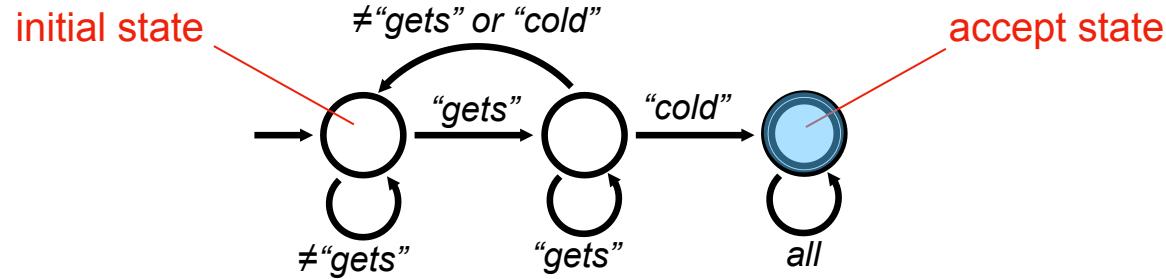


String: “The weather gets cold in the winter.”

Representing Logical Constraints

as a **deterministic finite automaton (DFA)**

Example. Check if a string contains “gets cold”.

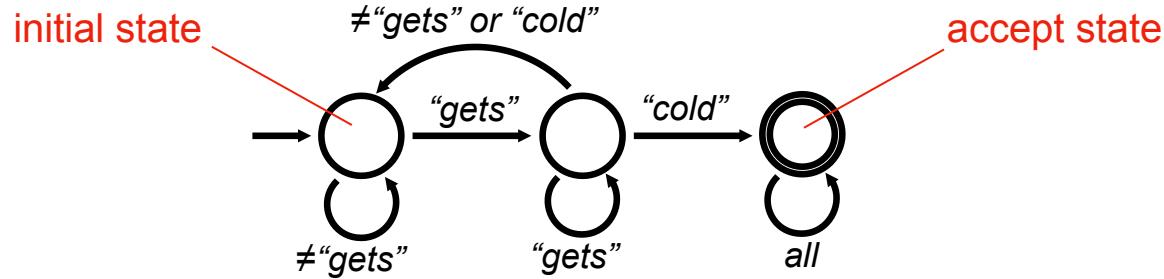


String: “The weather gets cold in the winter.”

Representing Logical Constraints

as a **deterministic finite automaton (DFA)**

Example. Check if a string contains “gets cold”.



Can represent:

Phrases/words must/must not appear

From a restricted vocabulary.

Exactly k times.

Must end a certain way

Any regex

Anything over fixed sequence lengths (BDD)

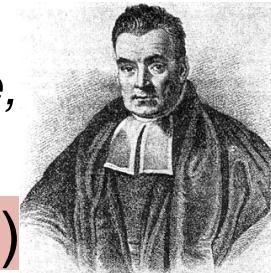
...

Reasoning about all Future Tokens: Constraints

$$p_{lm}(\text{next-token} \mid \alpha, \text{prefix})$$

Abusing Bayes rule,

$$\propto p_{lm}(\text{next-token} \mid \text{prefix}) \cdot p_{\text{circuit}}(\alpha \mid \text{next-token, prefix})$$



Theorem. Given

1. a deterministic finite automata constraint α with m edges and
2. a probabilistic circuit $p(\cdot)$ with h hidden states (representing a Hidden Markov Model) ,

computing $p(\alpha \mid x_{1:t})$ over a sequence of n future tokens takes $O(nmh^2)$ time.

CommonGen Benchmark

Generate a sentence using 3 to 5 concepts (keywords).

Input: snow drive car

$$\alpha = ("car" \vee "cars" \dots) \wedge ("drive" \vee "drove" \dots) \wedge$$

Reference 1: A car drives down a snow-covered road.

Reference 2: Two cars drove through the snow.

	BLEU-4		ROUGE-L		CIDEr		SPICE		Constraint	
	dev	test	dev	test	dev	test	dev	test	dev	test
<i>supervised</i> - base models trained with full supervision										
FUDGE	-	24.6	-	40.4	-	-	-	-	-	47.0%
A*esque	-	28.2	-	43.4	-	15.2	-	30.8	-	98.8%
NADO	30.8	-	44.4	-	16.1	-	32.0	-	88.8%	-
Ctrl-G	35.1	34.4	46.7	46.4	17.4	17.6	32.7	33.3	100.0%	100.0%
<i>unsupervised</i> - base models not trained with keywords as supervision										
A*esque	-	28.6	-	44.3	-	15.6	-	29.6	-	-
NADO	26.2	-	-	-	-	-	-	-	-	-
Ctrl-G	32.1	31.5	45.2	44.8	16.0	16.2	30.8	31.2	100.0%	100.0%

Interactive Text Editing

"First they've defeated a small squad [BLANK] are few humans left, and despite their magical power, their numbers are getting fewer."

Interactive Text Editing

User: given the following context, generate infilling text for [BLANK] using key phrases "alien mothership", "far from over"; generated text must contain 25 - 30 words.

"First they've defeated a small squad [BLANK] are few humans left, and despite their magical power, their numbers are getting fewer."

Ctrl-G



"First they've defeated a small squad of aliens, then a larger fleet of their ships. Eventually they've even managed to take down the alien mothership. But their problems are far from over. There are few humans left, and despite their magical power, their numbers are getting fewer."

Interactive Text Editing with key phrase (K) or length (L) constraints



Quality	CoAuthor		
	K	L	K&L
TULU2	2.64	2.78	2.74
GPT3.5	2.22	2.27	2.31
GPT4	3.33	3.53	3.10
Ctrl-G	3.56	3.73	3.59

→ *How many stars by humans?*

Interactive Text Editing with key phrase (K) or length (L) constraints

CoAuthor			
	K	L	K&L
<i>Quality</i>			
TULU2	2.64	2.78	2.74
GPT3.5	2.22	2.27	2.31
GPT4	3.33	3.53	3.10
Ctrl-G	3.56	3.73	3.59
<i>Success</i>			
TULU2	12%	20%	3%
GPT3.5	22%	54%	10%
GPT4	60%	20%	27%
Ctrl-G	100%	100%	100%

→ *How many stars by humans?*

→ *Follows instructions?*

Interactive Text Editing with key phrase (K) or length (L) constraints

CoAuthor			
	K	L	K&L
<i>Quality</i>			
TULU2	2.64	2.78	2.74
GPT3.5	2.22	2.27	2.31
GPT4	3.33	3.53	3.10
Ctrl-G	3.56	3.73	3.59
<i>Success</i>			
TULU2	12%	20%	3%
GPT3.5	22%	54%	10%
GPT4	60%	20%	27%
Ctrl-G	100%	100%	100%
<i>Overall</i>			
TULU2	7%	10%	1%
GPT3.5	0%	5%	2%
GPT4	41%	17%	14%
Ctrl-G	76%	78%	82%

→ *How many stars by humans?*

→ *Follows instructions?*

→  & Up + *Follows instructions?*

→ **Ctrl-G based on Llama2-7B wipes the floor with GPT4**, which is a >100x bigger LLM

Grade School Math Benchmark

Question: Kylar went to the store to buy glasses for his new apartment. One glass costs \$5, but every second glass costs only 60% of the price. Kylar wants to buy 16 glasses. How much does he need to pay for them?

Vanilla LLM Answer: The price of the 2nd glass is $(16 / 2) * 60\% = 8$ dollars. So one pair of glasses costs $16 + 8 = 24$ dollars. **So the answer is 24.**

Grade School Math Benchmark

Question: Kylar went to the store to buy glasses for his new apartment. One glass costs \$5, but every second glass costs only 60% of the price. Kylar wants to buy 16 glasses. How much does he need to pay for them?

Vanilla LLM Answer: The price of the 2nd glass is $(16 / 2) * 60\% = 8$ dollars. So one pair of glasses costs $16 + 8 = 24$ dollars. **So the answer is 24.**

Ctrl-G Answer: The second glass costs $5 * .6 = \$3$. So each set of two glasses actually costs $5 + 3 = \$8$. He wants $16 / 2 = 8$ sets of two. That means he needs to pay $8 * 8 = \$64$. **So the answer is 64.**

Which constraint improves accuracy?

Grade School Math Benchmark

Question: Kylar went to the store to buy glasses for his new apartment. One glass costs \$5, but every second glass costs only 60% of the price. Kylar wants to buy 16 glasses. How much does he need to pay for them?

Vanilla LLM Answer: The price of the 2nd glass is $(16 / 2) * 60\% = 8$ dollars. So one pair of glasses costs $16 + 8 = 24$ dollars. So the answer is 24.

Ctrl-G Answer: The second glass costs $5 * .6 = \$3$. So each set of two glasses actually costs $5 + 3 = \$8$. He wants $16 / 2 = 8$ sets of two. That means he needs to pay $8 * 8 = \$64$. So the answer is 64.

Use all the numbers in the problem statement!

Advantages of Ctrl-G:

1. Constraint α is guaranteed to be satisfied:

if next-token makes α unsatisfiable, $p_{lm}(\text{next-token} \mid \alpha, \text{prefix}) = 0$.

$$p_{lm}(\text{next-token} \mid \text{prefix}) \cdot p_{\text{circuit}}(\alpha \mid \text{next-token}, \text{prefix}) = 0$$

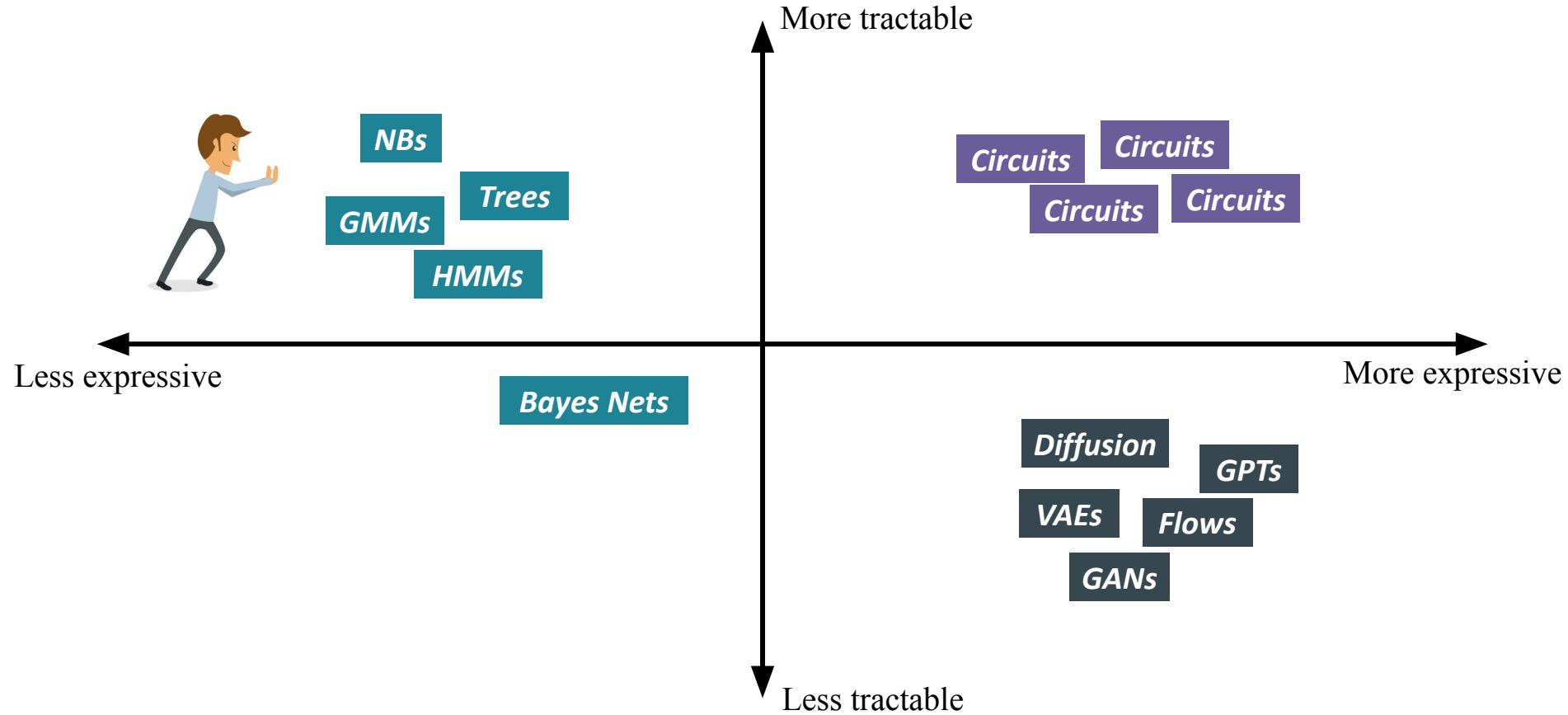
2. Generalizes well to unseen reasoning tasks, because all tasks are unseen :-)
(training on a distribution over tasks is slow and brittle!)
3. Bayesian = goal-oriented (\leftrightarrow structured generation tools)

You can control an intractable generative model using a generative model that is *tractable for symbolic reasoning*.

Questions for this talk:



1. Do deductive reasoning algorithms still have a purpose in the age of LLMs?
2. **Where did reasoning algorithms go wrong?
What should they look like today?**
3. Can reasoning algorithms provide a path to language model alignment, safety?



Generative Models

polynomials model **joint distributions**

$$p(x_1, x_2, x_3) = .1x_1 + .05x_2 + .1x_1x_2 + .01x_3 - .07x_2x_3 + .02x_1x_3 - .14x_1x_2x_3 + .05$$

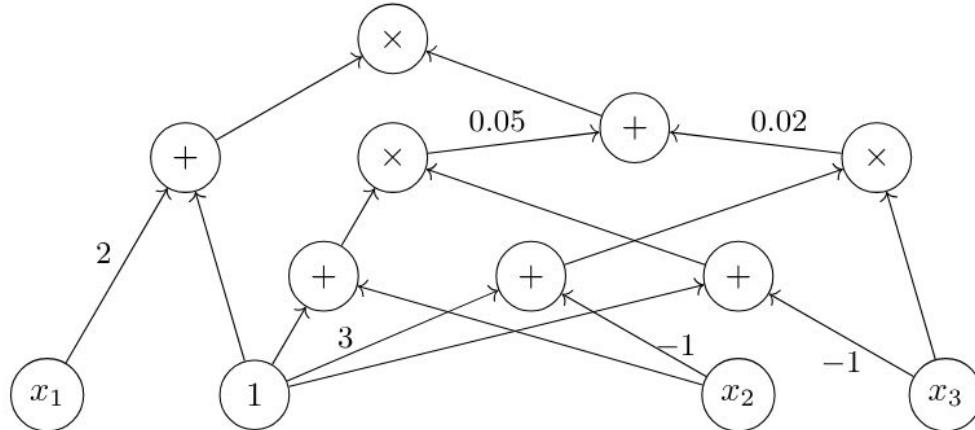
X_1	X_2	X_3	p
0	0	0	0.05
1	0	0	0.15
0	1	0	0.1
1	1	0	0.3
0	0	1	0.06
1	0	1	0.18
0	1	1	0.04
1	1	1	0.12

Deep Generative Models

circuit polynomials model **joint distributions** compactly
(and can have billions of trainable parameters)

$$p(x_1, x_2, x_3) = .1x_1 + .05x_2 + .1x_1x_2 + .01x_3 - .07x_2x_3 + .02x_1x_3 - .14x_1x_2x_3 + .05$$

X_1	X_2	X_3	p
0	0	0	0.05
1	0	0	0.15
0	1	0	0.1
1	1	0	0.3
0	0	1	0.06
1	0	1	0.18
0	1	1	0.04
1	1	1	0.12



Probabilistic Reasoning Task

Marginal inference:

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$\begin{aligned}\Pr[X_1 = 1] &= \Pr[X_1 = 1, X_2 = 0] + \Pr[X_1 = 1, X_2 = 1] \\ &= 0.3 + 0.4 \\ &= 0.7\end{aligned}$$

Application: Ctrl-G



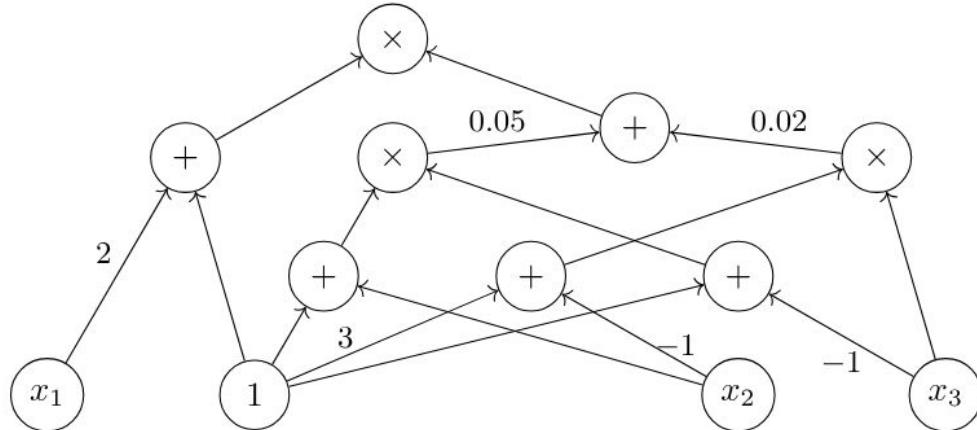
$p_{\text{circuit}}(\alpha \mid \text{next-token, prefix})$ is summing over all future text

Deep Generative Models

circuit polynomials model **joint distributions** compactly
(and can have billions of trainable parameters)

$$p(x_1, x_2, x_3) = .1x_1 + .05x_2 + .1x_1x_2 + .01x_3 - .07x_2x_3 + .02x_1x_3 - .14x_1x_2x_3 + .05$$

X_1	X_2	X_3	p
0	0	0	0.05
1	0	0	0.15
0	1	0	0.1
1	1	0	0.3
0	0	1	0.06
1	0	1	0.18
0	1	1	0.04
1	1	1	0.12

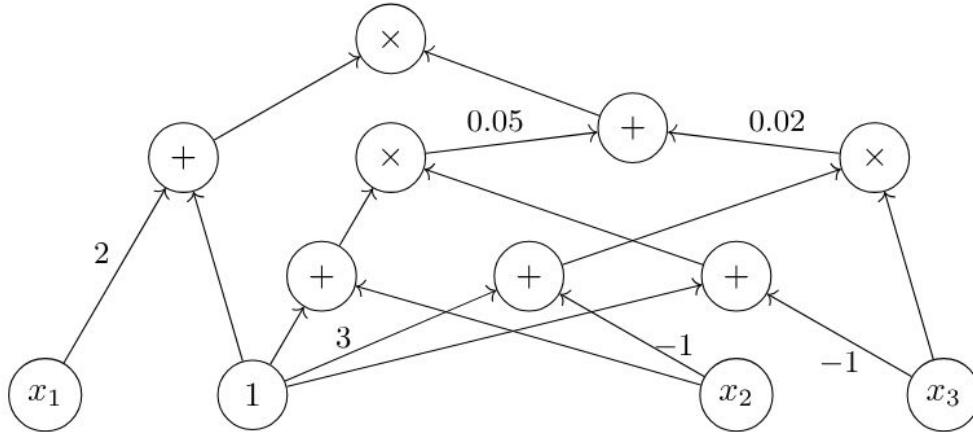


Tractable Deep Generative Models

Multilinear circuit polynomials model **joint distributions** compactly
and allow **efficient** probabilistic reasoning (marginalization)

$$p(x_1, x_2, x_3) = .1x_1 + .05x_2 + .1x_1x_2 + .01x_3 - .07x_2x_3 + .02x_1x_3 - .14x_1x_2x_3 + .05$$

X_1	X_2	X_3	p
0	0	0	0.05
1	0	0	0.15
0	1	0	0.1
1	1	0	0.3
0	0	1	0.06
1	0	1	0.18
0	1	1	0.04
1	1	1	0.12



Probabilistic Circuit Language Model

How did we train a probabilistic circuit to solve Ctrl-G?

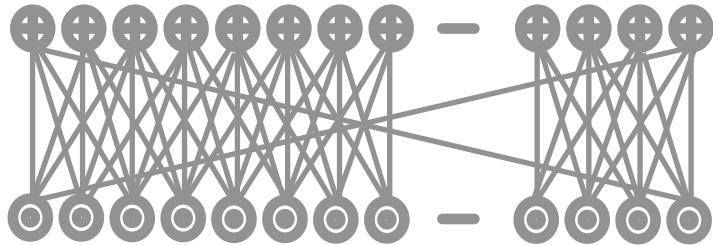
Keep it simple... just a classic **Hidden Markov Model** (HMM) with 32,768 hidden states and 2 billion parameters... on the GPU



Theorem. Given a DFA constraint α with m edges and an HMM $p(x)$ with h hidden states, computing $p(\alpha \mid x_{1:t+1})$ over a sequence of n tokens takes $O(nmh^2)$ time.

Scaling Up Probabilistic Circuits

Linear Layers



d nodes

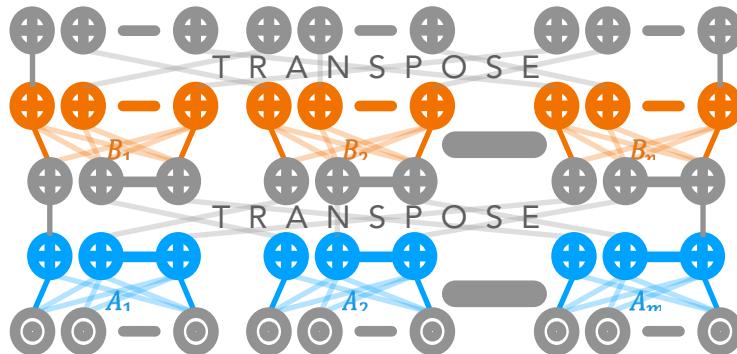
$O(d^2)$ edges

$$y_{ij} = \sum_{kl} A_{ijkl} x_{kl}$$

Dense Matrices



e.g. a model w/ just 250K nodes requires 69B parameters (memory + time)...



d nodes

$O(d^{3/2})$ edges

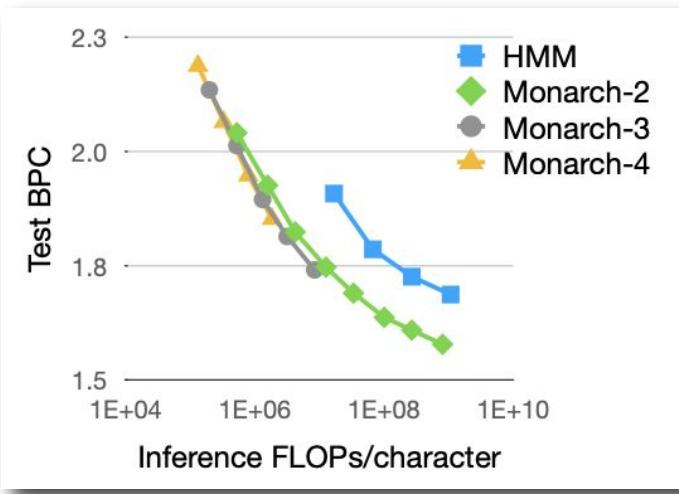
$$y_{ij} = \sum_{kl} B_{ijk} A_{jkl} x_{kl}$$

Monarch Matrices



... now just 134M parameters required!

Scaling Up Probabilistic Circuits



Type	Model	BPC (↓)	Time (s) (↓)
Flow	IAF/SCF	1.88	0.04
Flow	Argmax Coup Flow	1.80	0.40
Diffusion	D3PM Uniform	≤ 1.61	3.60
Diffusion	SEDD Uniform	≤ 1.47	-
PC	SparsePC	2.60	-
PC	NPC ²	3.17	-
PC	HMM	1.69	0.006
PC	Monarch-HMM	1.57	0.017

Text8 Character-Level Language Modelling
Roughly on par with Flow and Diffusion models

You Tricked Us

You promised us reasoning algorithms...

... and all we got was another lousy feedforward neural network!



Theorem. If there exists a polynomial time (real RAM) **algorithm** that computes (virtual evidence) marginal probabilities for a class of distributions, then there exist **poly-size circuits** for their **multilinear polynomials**.



Questions for this talk:



1. Do deductive reasoning algorithms still have a purpose in the age of LLMs?
2. Where did reasoning algorithms go wrong? What should they look like today?
3. **Can reasoning algorithms provide a path to language model alignment, safety?**

Reasoning about all Future Tokens: *Alignment*

$p(\text{next-token} \mid \alpha, \text{prefix})$

in	0.03
to	0.08

Prefix: It's a pain ...

Constraint α : non-toxic

$\propto p(\text{next-token} \mid \text{prefix})$

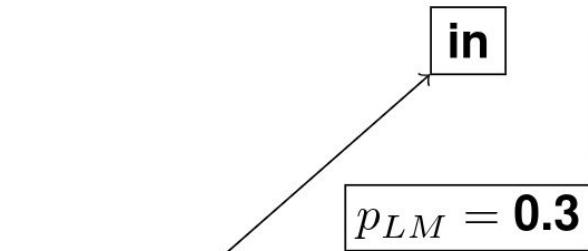
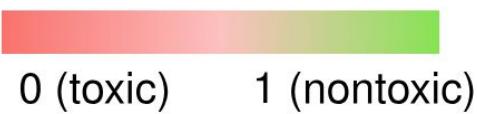
$\cdot p(\alpha \mid \text{next-token, prefix})$

in	0.3
to	0.1

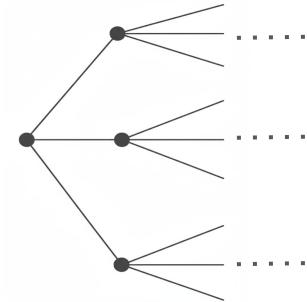
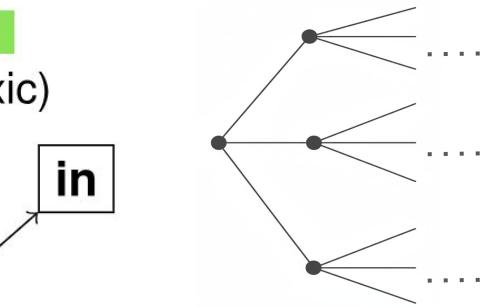
in	0.1
to	0.8



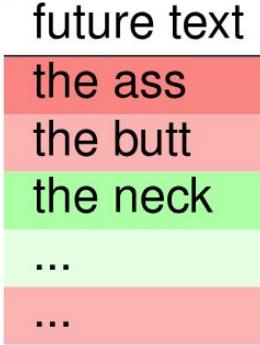
Attribute Probability



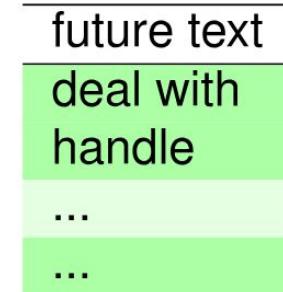
It's a pain



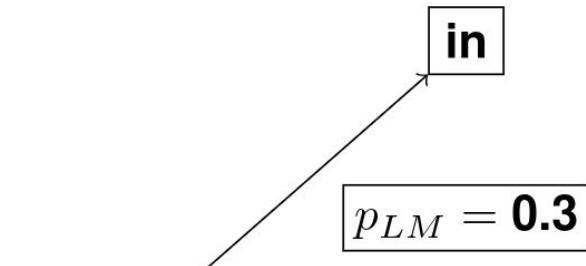
$p_{LM} = 0.1$



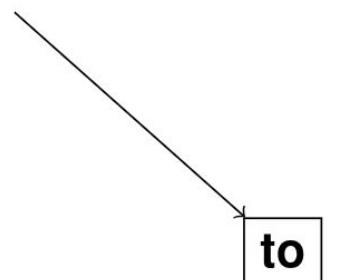
Intractable to know
expected future toxicity



Attribute Probability



It's a pain



$$p_{LM} = 0.1$$

future text

the ass

the butt

the neck

2

■ ■

future text

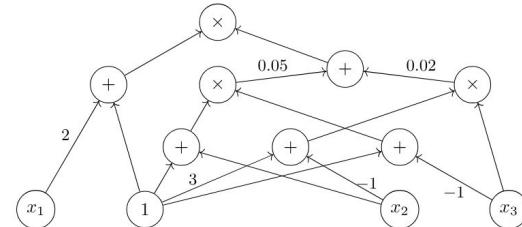
deal with

handle

10

1

Model LLM continuations with *tractable probabilistic circuit*



+

Model goal attribute with *log-linear classifier*



1

Efficient Expected Attribute Probability!



Attribute Probability

0 (toxic) 1 (nontoxic)

in

$p_{LM} = 0.3$

It's a pain

to

$p_{LM} = 0.1$

future text

the ass

the butt

the neck

...

...

$EAP = 0.1$

future text

deal with

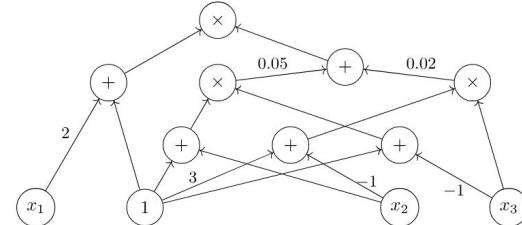
handle

...

...

$EAP = 0.8$

Model LLM continuations with
tractable probabilistic circuit



+

Model goal attribute with
log-linear classifier



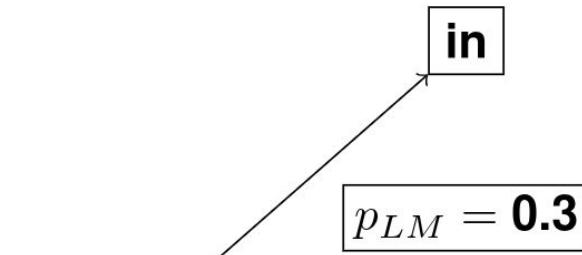
=

**Efficient Expected
Attribute Probability!**



Attribute Probability

0 (toxic) 1 (nontoxic)



It's a pain



$p_{LM} = 0.1$

future text

the ass

the butt

the neck

...

...

$EAP = 0.1$

$= p_{TRACE} \propto 0.03$



future text

deal with

handle

...

...

$EAP = 0.8$

$= p_{TRACE} \propto 0.08$

State-of-the-art LLM Detoxification

Model	Toxicity (↓) avg. max. prob.	Approach Type
GPT-2 Large Results		
GPT2	0.385	Baseline
DAPT ⁽¹⁾	0.428	Finetuning
GeDi ⁽²⁾	0.363	Decoding (Trained Guide)
FUDGE ⁽³⁾	0.302	Decoding (Trained Guide)
DExperts ⁽⁴⁾	0.314	Decoding (Trained Guide)
PPLM ⁽⁵⁾	0.520	Decoding (Logit Control)
MuCoLa ⁽⁶⁾	0.308	Decoding (Sampling)
PPO ⁽⁷⁾	0.218	RL
Quark ⁽⁸⁾	0.196	RL
DPO ⁽⁹⁾	0.180	RL
TRACE	0.163	Decoding (HMM Reasoning)
Gemma-2B Results		
Gemma-2B	0.359	Baseline
DPO ⁽⁹⁾	0.222	RL
TRACE	0.189	Decoding (HMM Reasoning)

....but...
it's easy to be non-toxic
by reusing
the same bland response...

State-of-the-art LLM Detoxification

Model	Toxicity (↓)		Diversity (↑)	
	avg.	max. prob.	dist-2	dist-3
GPT-2 Large Results				
GPT2	0.385	0.254	0.87	0.86
DAPT ⁽¹⁾	0.428	0.360	0.84	0.84
GeDi ⁽²⁾	0.363	0.217	0.84	0.83
FUDGE ⁽³⁾	0.302	0.371	0.78	0.82
DExperts ⁽⁴⁾	0.314	0.128	0.84	0.84
PPLM ⁽⁵⁾	0.520	0.518	0.86	0.86
MuCoLa ⁽⁶⁾	0.308	0.088	0.82	0.83
PPO ⁽⁷⁾	0.218	0.044	0.80	0.84
Quark ⁽⁸⁾	0.196	0.035	0.80	0.84
DPO ⁽⁹⁾	0.180	0.026	0.76	0.78
TRACE	0.163	0.016	0.85	0.85
Gemma-2B Results				
Gemma-2B	0.359	0.23	0.86	0.85
DPO ⁽⁹⁾	0.222	0.06	0.74	0.77
TRACE	0.189	0.02	0.86	0.85

Method	Entropy (↑)
GPT2-large	52.06
DPO	39.52
TRACE	52.54

Decoding (Trained Guide)

Decoding (Trained Guide)

Decoding (Trained Guide)

Decoding (Logit Control)

Decoding (Sampling)

RL

RL

RL

Decoding (HMM Reasoning)



....but...

*it's easy to be non-toxic
by responding gibberish...*

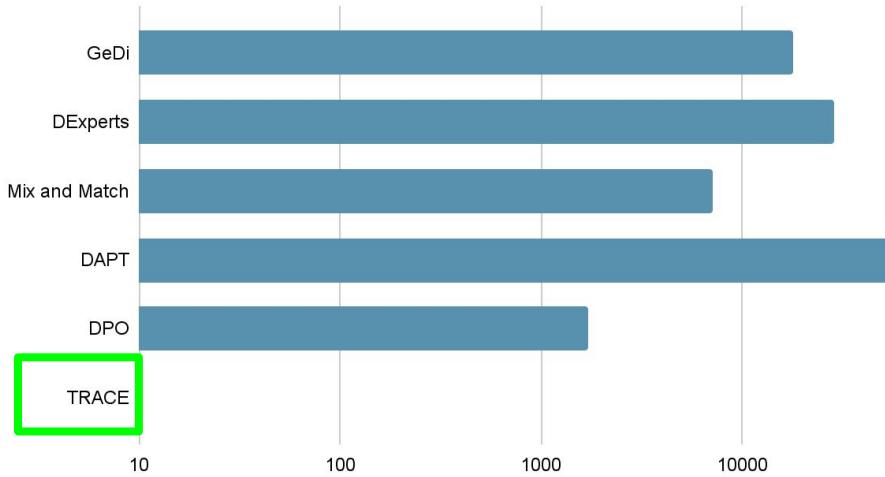
State-of-the-art LLM Detoxification

Model	Toxicity (↓)		Diversity (↑)		Fluency (↓)	Approach Type
	avg.	max. prob.	dist-2	dist-3		
GPT-2 Large Results						
GPT2	0.385	0.254	0.87	0.86	25.57	Baseline
DAPT ⁽¹⁾	0.428	0.360	0.84	0.84	31.21	Finetuning
GeDi ⁽²⁾	0.363	0.217	0.84	0.83	60.03	Decoding (Trained Guide)
FUDGE ⁽³⁾	0.302	0.371	0.78	0.82	<u>12.97</u> *	Decoding (Trained Guide)
DExperts ⁽⁴⁾	0.314	0.128	0.84	0.84	32.41	Decoding (Trained Guide)
PPLM ⁽⁵⁾	0.520	0.518	0.86	0.86	32.58	Decoding (Logit Control)
MuCoLa ⁽⁶⁾	0.308	0.088	0.82	0.83	29.92	Decoding (Sampling)
PPO ⁽⁷⁾	0.218	0.044	0.80	0.84	<u>14.27</u> *	RL
Quark ⁽⁸⁾	0.196	0.035	0.80	0.84	<u>12.47</u> *	RL
DPO ⁽⁹⁾	0.180	0.026	0.76	0.78	<u>21.59</u> *	RL
TRACE	0.163	0.016	0.85	0.85	29.83	Decoding (HMM Reasoning)
Gemma-2B Results						
Gemma-2B	0.359	0.23	0.86	0.85	15.75	Baseline
DPO ⁽⁹⁾	0.222	0.06	0.74	0.77	<u>14.39</u> *	RL
TRACE	0.189	0.02	0.86	0.85	17.68	Decoding (HMM Reasoning)

TRACE is Blazingly Fast

Given a language model, and its tractable twin,
train log-linear attribute classifier

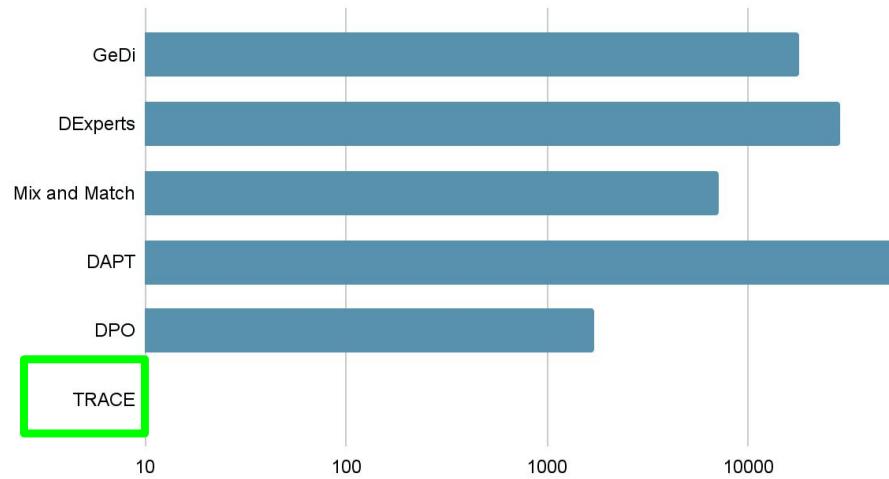
Training Time per Attribute (seconds)



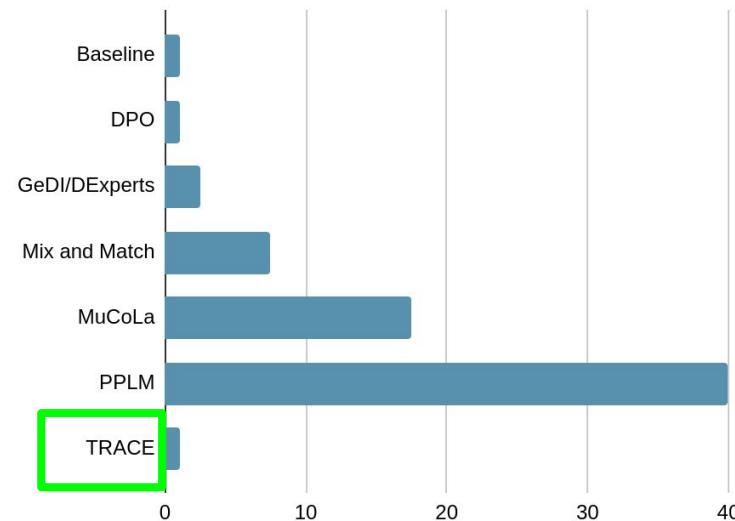
TRACE is Blazingly Fast

Given a language model, and its tractable twin,
train log-linear attribute classifier,
then use Bayesian logits at decoding time

Training Time per Attribute (seconds)



Inference Time



Personalized Language Model: Twilight Sparkle



Baseline



Prompt

You are an advanced role-playing assistant trained to embody characters with accuracy and authenticity. In this instance, you will assume the persona of Twilight Sparkle.

10 QA Examples: 1...2...3...4...5...6...7...8...9...10...

Question: Twilight Sparkle, how is the weather?

Generation

The weather is pretty hot and humid here, thanks to our climate.

TRACE



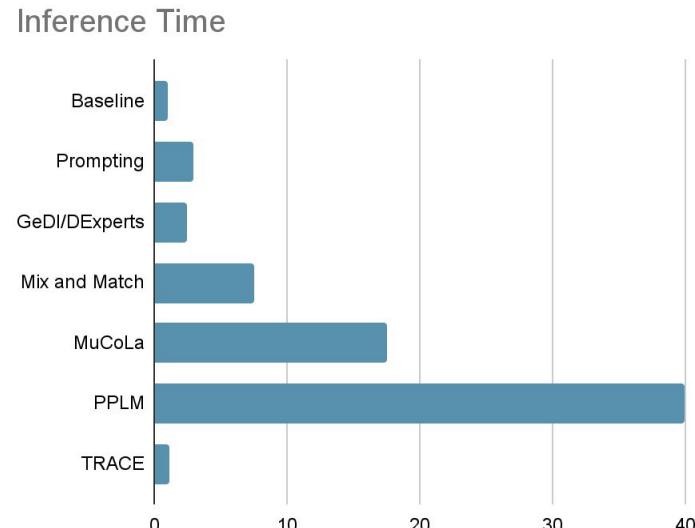
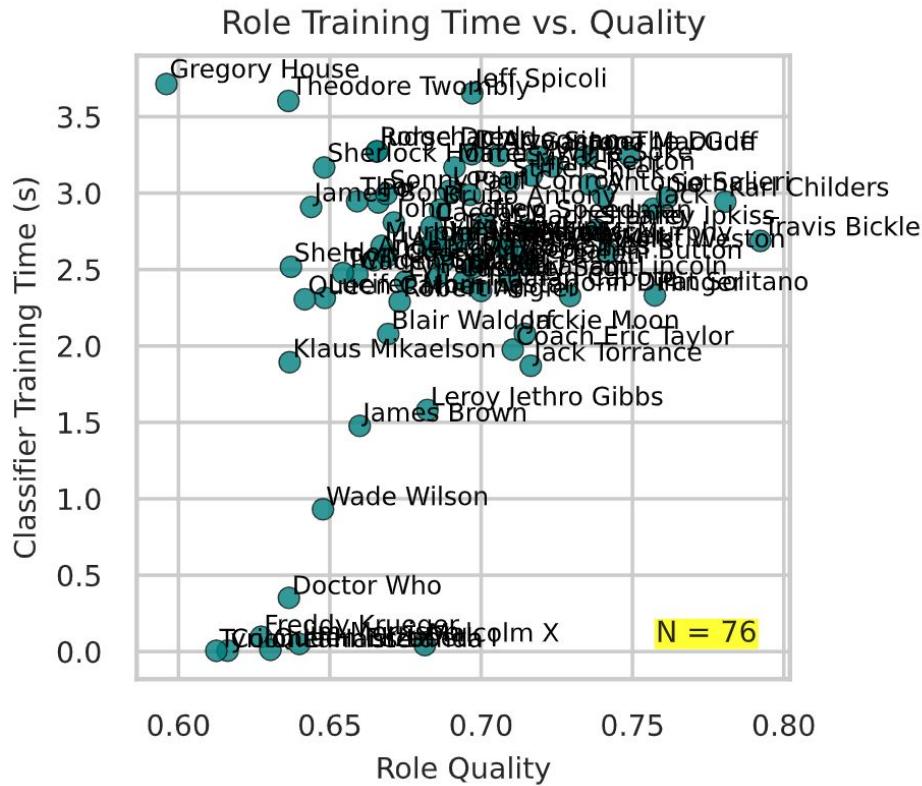
Prompt

How is the weather?

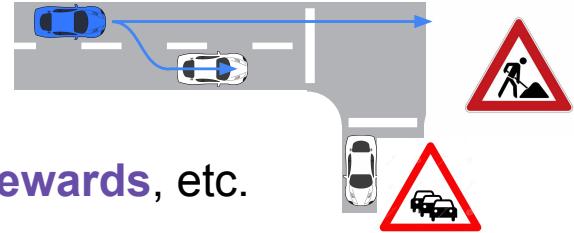
Generation

Gosh, it's **sunny** and very **beautiful** and all around me.

76 Personalized Language Models



Reasoning about all Future Tokens: *Offline RL*



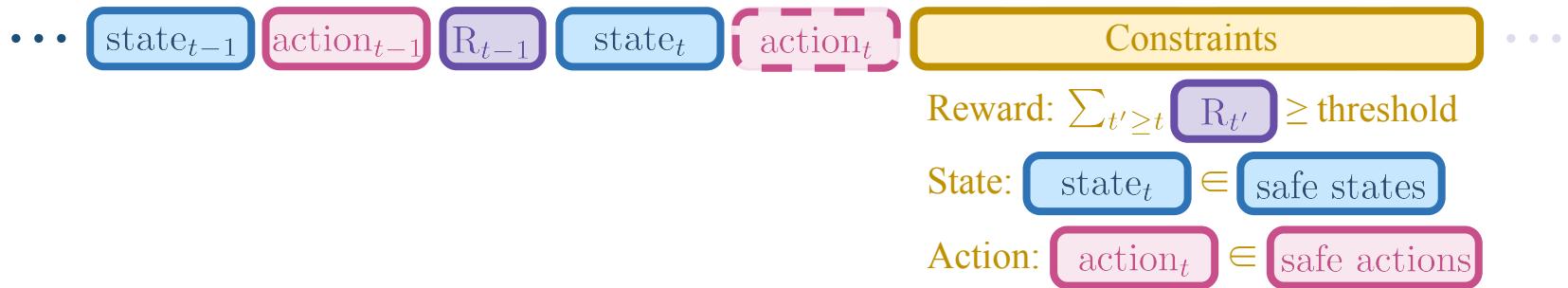
Training: model the joint distribution over **states**, **actions**, **rewards**, etc.

Inference: sample next **states** and **actions**, as well as **constraints**.



$$p(\text{action} \mid \alpha, \text{prefix}) \propto p(\text{action} \mid \text{prefix}) \cdot p(\alpha \mid \text{action}, \text{prefix})$$

Reasoning about all Future Tokens: *Offline RL*



Inference: sample actions condition on past **states** and **actions**, as well as **constraints**.

$$p(\text{action}_t | \text{state}_{\leq t}, \text{action}_{<t}, \text{Constraints})$$

$$\propto p(\text{action}_t | \text{state}_{\leq t}, \text{action}_{<t}) \cdot p(\text{Constraints} | \text{state}_{\leq t}, \text{action}_{<t}) \quad \text{Bayes' rule}$$

Autoregressive Transformers
(GPTs)

Probabilistic Circuits
(PCs)



Condition on Various Constraints in Offline RL

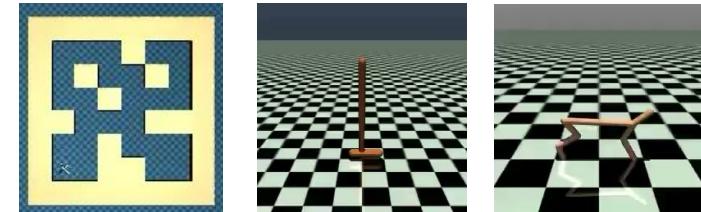
- Condition on high reward: SoTA performance on standard offline RL benchmarks.

Dataset	Environment	TT		TT(+Q)		DT		DD	IQL	CQL	%BC	TD3(+BC)
		base	Trifle	base	Trifle	base	Trifle					
Med-Expert	HalfCheetah	95.0 \pm 0.2	95.1\pm0.3	82.3 \pm 6.1	89.9\pm4.6	86.8 \pm 1.3	91.9\pm1.9	90.6	86.7	91.6	92.9	90.7
Med-Expert	Hopper	110.0 \pm 2.7	113.0\pm0.4	74.7 \pm 6.3	78.5\pm6.4	107.6 \pm 1.8	/	111.8	91.5	105.4	110.9	98.0
Med-Expert	Walker2d	101.9 \pm 6.8	109.3\pm0.1	109.3 \pm 2.3	109.6\pm0.2	108.1 \pm 0.2	108.6\pm0.3	108.8	109.6	108.8	109.0	110.1
Medium	HalfCheetah	46.9 \pm 0.4	49.5\pm0.2	48.7 \pm 0.3	48.9\pm0.3	42.6 \pm 0.1	44.2\pm0.7	49.1	47.4	44.0	42.5	48.3
Medium	Hopper	61.1 \pm 3.6	67.1\pm4.3	55.2 \pm 3.8	57.8\pm1.9	67.6 \pm 1.0	/	79.3	66.3	58.5	56.9	59.3
Medium	Walker2d	79.0 \pm 2.8	83.1\pm0.8	82.2 \pm 2.5	84.7\pm1.9	74 \pm 1.4	81.3\pm2.3	82.5	78.3	72.5	75.0	83.7
Med-Replay	HalfCheetah	41.9 \pm 2.5	45.0\pm0.3	48.2 \pm 0.4	48.9\pm0.3	36.6 \pm 0.8	39.2\pm0.4	39.3	44.2	45.5	40.6	44.6
Med-Replay	Hopper	91.5 \pm 3.6	97.8\pm0.3	83.4 \pm 5.6	87.6\pm6.1	82.7 \pm 7.0	/	100.0	94.7	95.0	75.9	60.9
Med-Replay	Walker2d	82.6 \pm 6.9	88.3\pm3.8	84.6 \pm 4.5	90.6\pm4.2	66.6 \pm 3.0	73.5\pm0.1	75.0	73.9	77.2	62.5	81.8
Average Score		78.9	83.1	74.3	77.4	74.7	/	81.8	77.0	77.6	74.0	75.3

- Also works in stochastic environments



Methods	Taxi	FrozenLake		
		$\epsilon = 0.3$	$\epsilon = 0.5$	$\epsilon = 0.7$
m-Trifle	-57	0.61	0.59	0.37
s-Trifle	-99	0.62	0.60	0.34
TT [20]	-182	0.63	0.25	0.12
DT [6]	-388	0.51	0.32	0.10
DoC [47]	-146	0.58	0.61	0.23



- Condition on safe actions

Dataset	Environment	Trifle	TT
Med-Expert	Halfcheetah	81.9\pm4.8	77.8 \pm 5.4
Med-Expert	Hopper	109.6\pm2.4	100.0 \pm 4.2
Med-Expert	Walker2d	105.1\pm2.3	103.6 \pm 4.9

Conclusions for this talk:

1. Do deductive reasoning algorithms still have a purpose in the age of LLMs?
2. Where did reasoning algorithms go wrong?



What should they look like today?

Conclusions for this talk:

1. Do deductive reasoning algorithms still have a purpose in the age of LLMs?



Yes, more cool applications of reasoning algorithms than can fit on these slides!

2. Where did reasoning algorithms go wrong?

What should they look like today?

Conclusions for this talk:

1. Do deductive reasoning algorithms still have a purpose in the age of LLMs?



Yes, more cool applications of reasoning algorithms than can fit on these slides!

2. Where did reasoning algorithms go wrong?

Learn at scale, be tractable

What should they look like today?

Conclusions for this talk:

1. Do deductive reasoning algorithms still have a purpose in the age of LLMs?



Yes, more cool applications of reasoning algorithms than can fit on these slides!

2. Where did reasoning algorithms go wrong?

Learn at scale, be tractable

What should they look like today?

Circuits! Circuits! Circuits!

Thanks

*This was the work of many wonderful
students/postdocs/collaborators!*



References: <http://starai.cs.ucla.edu>