



Computer  
Science



# Weighted Model Counting Problems in the Context of LLMs

Guy Van den Broeck

Shonan Meeting - Model Counting: Theory meets Practice - Feb 1 2026

# Questions for this talk:

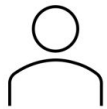


1. What is the purpose of weighted model counting in the age of LLMs?
2. Ensuing theory questions
  - a. When can we multiply circuits?
  - b. How to capture all tractable distributions?

# Questions for this talk:



1. **What is the purpose of weighted model counting in the age of LLMs?**
2. Ensuing theory questions
  - a. When can we multiply circuits?
  - b. How to capture all tractable distributions?



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



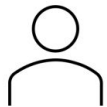
The frisbee flew through the air and the dog caught it with excitement.

*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**  $\alpha$ : 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**  $\alpha$ : sentence contains keyword "winter"

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



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**  $\alpha$ : sentence contains keyword "winter"

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

cold	0.50
warm	0.01



# Reasoning about all Future Tokens

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

*Using Bayes rule,*

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

*Intractable*





# Reasoning about all Future Tokens

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

*Abusing Bayes rule,*

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



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

***A 'tractable digital twin'***

# Reasoning about all Future Tokens: Constraints

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

*Abusing Bayes rule,*

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



**Theorem.** Given

1. a deterministic finite automata constraint  $\alpha$  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.

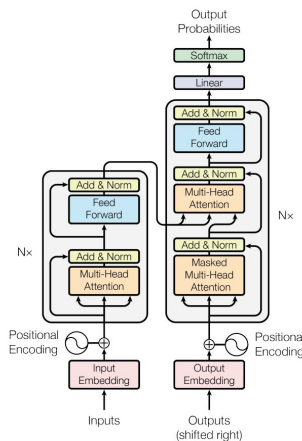
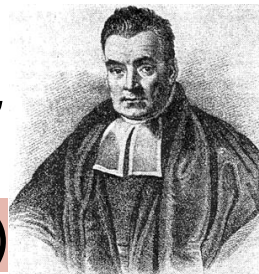
# Reasoning about all Future Tokens: Constraints

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

*Abusing Bayes rule,*

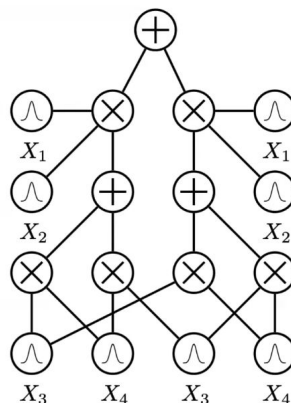
$$\propto p_{lm}(\text{next-token} \mid \text{prefix})$$

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

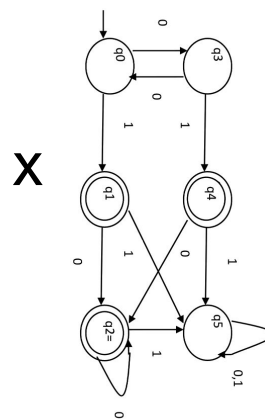


transformer

$\cdot \sum_{\text{future}}$



weights/circuit



models/dfa

# 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



	<i>K</i>	<i>L</i>	<i>K&amp;L</i>
<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	<b>3.56</b>	<b>3.73</b>	<b>3.59</b>

→ *How many stars by humans?*

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



	<i>K</i>	<i>L</i>	<i>K&amp;L</i>
<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	<b>3.56</b>	<b>3.73</b>	<b>3.59</b>
<i>Success</i>			
TULU2	12%	20%	3%
GPT3.5	22%	54%	10%
GPT4	60%	20%	27%
Ctrl-G	<b>100%</b>	<b>100%</b>	<b>100%</b>

→ *How many stars by humans?*

→ *Follows instructions?*

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



	<i>K</i>	<i>L</i>	<i>K&amp;L</i>
<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	<b>3.56</b>	<b>3.73</b>	<b>3.59</b>
<i>Success</i>			
TULU2	12%	20%	3%
GPT3.5	22%	54%	10%
GPT4	60%	20%	27%
Ctrl-G	<b>100%</b>	<b>100%</b>	<b>100%</b>
<i>Overall</i>			
TULU2	7%	10%	1%
GPT3.5	0%	5%	2%
GPT4	41%	17%	14%
Ctrl-G	<b>76%</b>	<b>78%</b>	<b>82%</b>

→ *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**



# 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



# 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  $\alpha$  is guaranteed to be satisfied:

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

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

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. What is the purpose of weighted model counting in the age of LLMs?
2. Ensuing theory questions
  - a. **When can we multiply circuits?**
  - b. How to capture all tractable distributions?

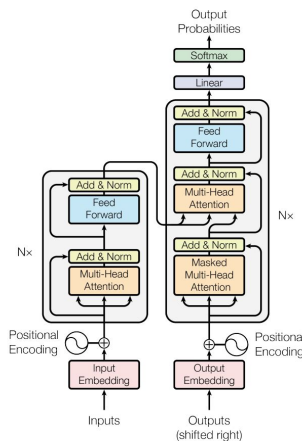
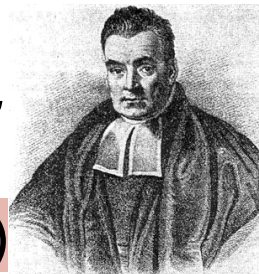
# Reasoning about all Future Tokens: Constraints

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

*Abusing Bayes rule,*

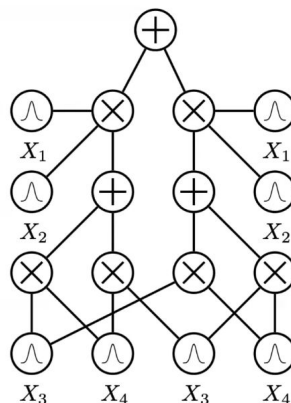
$$\propto p_{lm}(\text{next-token} \mid \text{prefix})$$

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

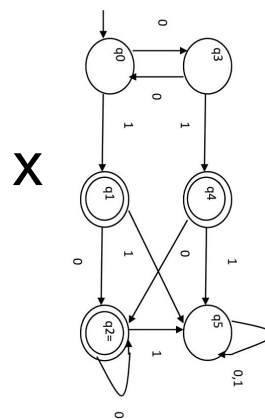


transformer

$\cdot \sum_{\text{future}}$



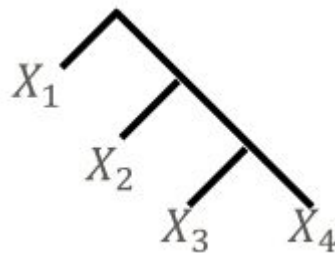
weights/circuit



models/dfa

# When can we multiply circuits efficiently?

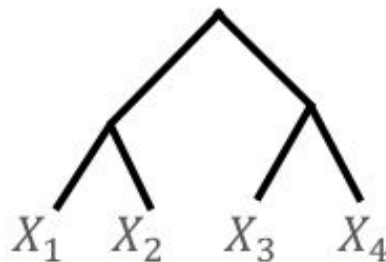
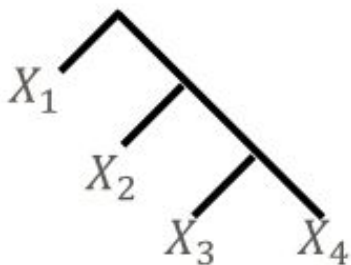
- Obviously when they
  - Are structured decomposable and
  - have the **same vtree**
    - E.g., two OBDDs with the same variable order
    - E.g., Hidden Markov model with DFA-turned-into-OBDD
- Otherwise easy to prove intractable
- *That's it?*





# When can we multiply circuits efficiently?

- We could “**restructure**” the circuits to give same vtree
  - Boring answer: modify constant-size sub-circuits
  - Otherwise ... **no global differences allowed?**



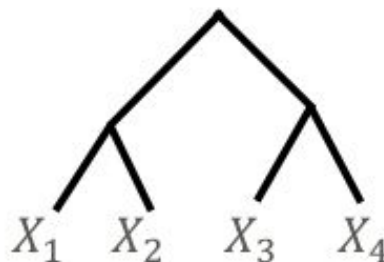
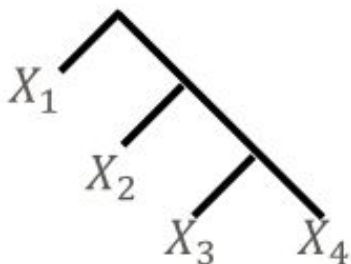
# Contiguous Circuits

We say that a circuit is compatible with a total order  $<$  on its variables if for every product node  $g$ , its children  $g_1, \dots, g_m$  (in some order) satisfy:

[ABJM'17]

$$\max(g_i) < \min(g_{i+1}) \forall i = 1, \dots, m - 1$$

We say that a circuit is contiguous if it is compatible with some order.



# Multiplying Contiguous Circuits, Deep case

Thm: Let A and B be **contiguous structured** circuits.

If A has a **linear** vtree, then A and B can be multiplied in PTIME and the size of the product circuit is  $O(|A|^2|B|)$ .

Example: A is a Hidden Markov Model,

B is a (Prob./Unambiguous) Context-Free Grammar

Example: A is an OBDD, B is an SDD, both contiguous

Thm: Theorem still works if B is **unstructured** but contiguous.

*You can efficiently multiply a structured and unstructured circuit!!!!*

# Multiplying Contiguous Circuits, Shallow case

Thm: Let A and B be **contiguous structured** circuits.

Let d be the **depth** of the vtree for A,

then A and B can be multiplied in time  $O(|A|^{12d}|B|)$ .

Example: A is log-depth, then multiplication is quasi-polynomial.

Cor: Every structured circuit can be made **structured log-depth**.

Thus, contiguous multiplication is quasipolynomial for some order.

Everything else is an open problem...

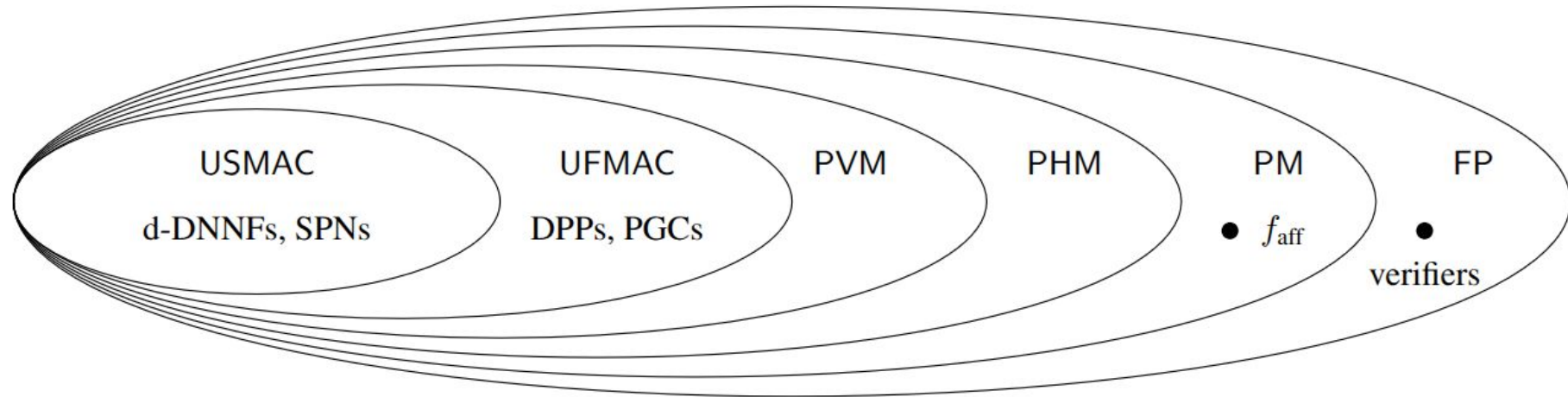
(lower bounds, reachable vtrees, what if vtree is deep but non-linear)

# Questions for this talk:



1. What is the purpose of weighted model counting in the age of LLMs?
2. Ensuing theory questions
  - a. When can we multiply circuits?
  - b. How to capture all tractable distributions?**

# What are the Limits of Tractable Marginalization?



USMAC: uniform syntactically multilinear circuits (decomposable circuits)

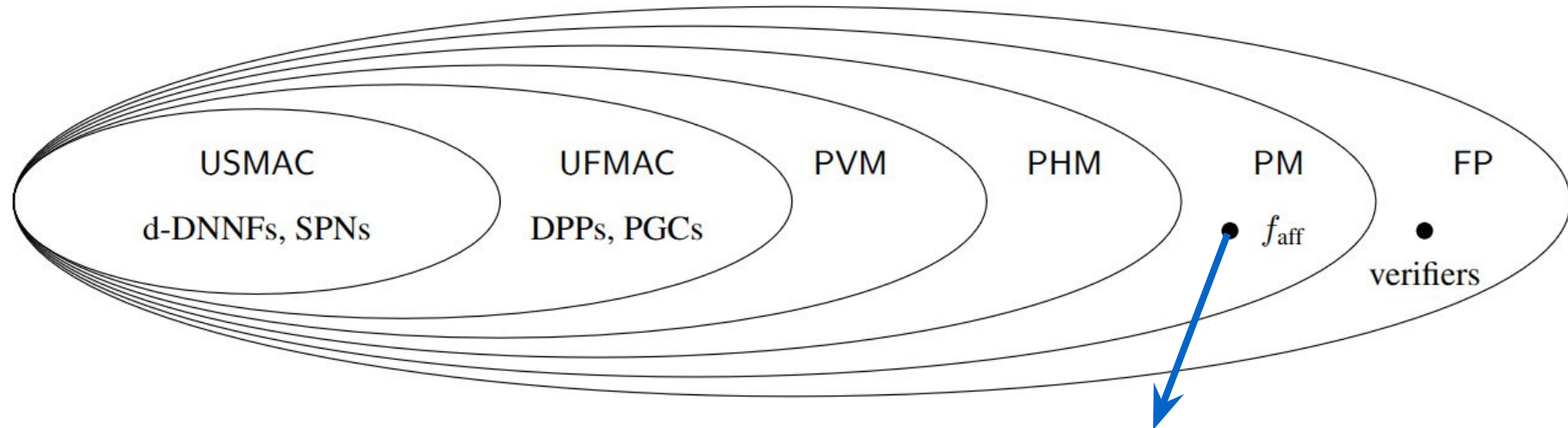
UFMACs: uniform finally multilinear circuits

PVM: efficient virtual evidence marginalization (weighted model counting)

PHM: efficient hamming weight marginalization

PM: efficient marginalization (model counting)

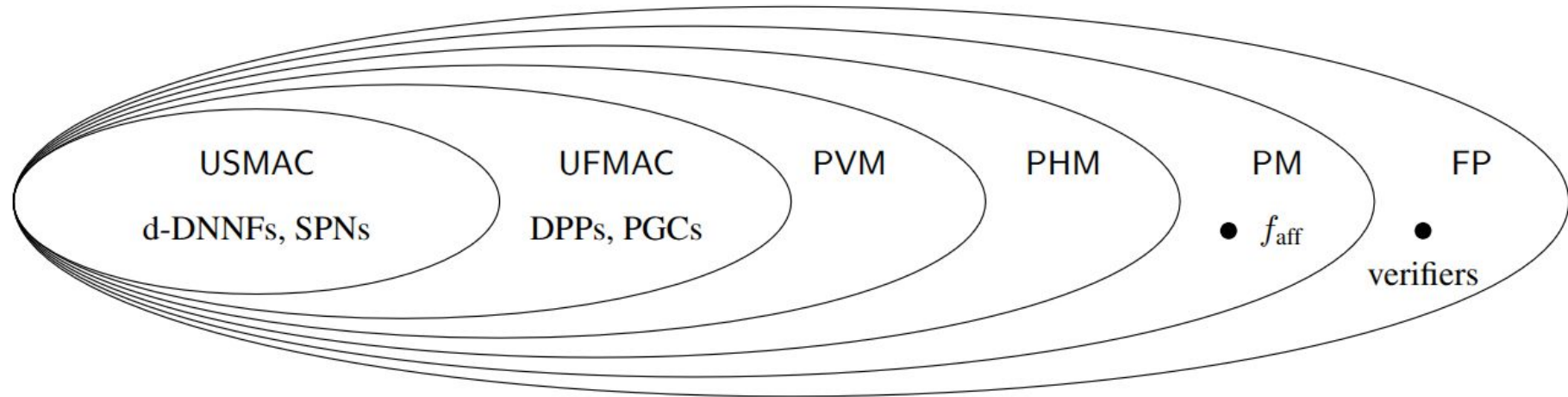
# What are the Limits of Tractable Marginalization?



**Theorem (Creignou and Hermann [1996]).** *If  $\Gamma$  contains only affine relations, then  $\#\text{CSP}(\Gamma)$  is in  $\text{PTIME}$ . Otherwise,  $\#\text{CSP}(\Gamma)$  is  $\#\text{P}$ -complete.*

**Theorem (Creignou et al. [2010]).** *If  $\Gamma$  contains only width-2 affine relations, then  $\#k\text{-ONES}(\Gamma)$  is in  $\text{PTIME}$ . Otherwise,  $\#k\text{-ONES}(\Gamma)$  is  $\#\text{P}$ -complete.*

# What are the Limits of Tractable Marginalization?



USMAC: uniform syntactically multilinear circuits (decomposable circuits)

UFMACs: uniform finally multilinear circuits

PVM: efficient virtual evidence marginalization (weighted model counting)

PHM: efficient hamming weight marginalization

PM: efficient marginalization (model counting)

FP: polynomial time computable functions



# On real RAM things are much simpler...

**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.*



## Open problems:

- Are multilinear circuits (UFMACs) ‘complete’ for virtual evidence marginalization/weighted model counting?
- Is the inclusion  $\text{PVM} \subseteq \text{PHM}$  strict?
- Are there non-parallelizable (P-hard) marginalization algorithms?

# Many ways to encode functions into tractable circuit polynomials

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

Network  
polynomial

$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2$$

Likelihood  
polynomial

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

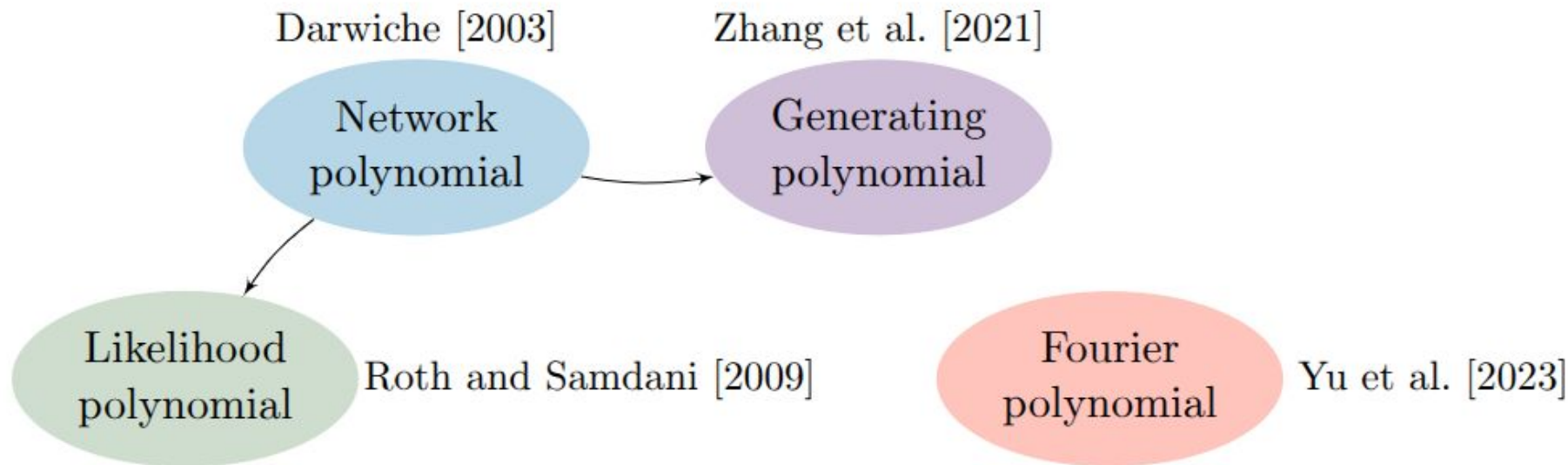
$\approx$  A neural net that for an input vector outputs its probability

Generating  
polynomial

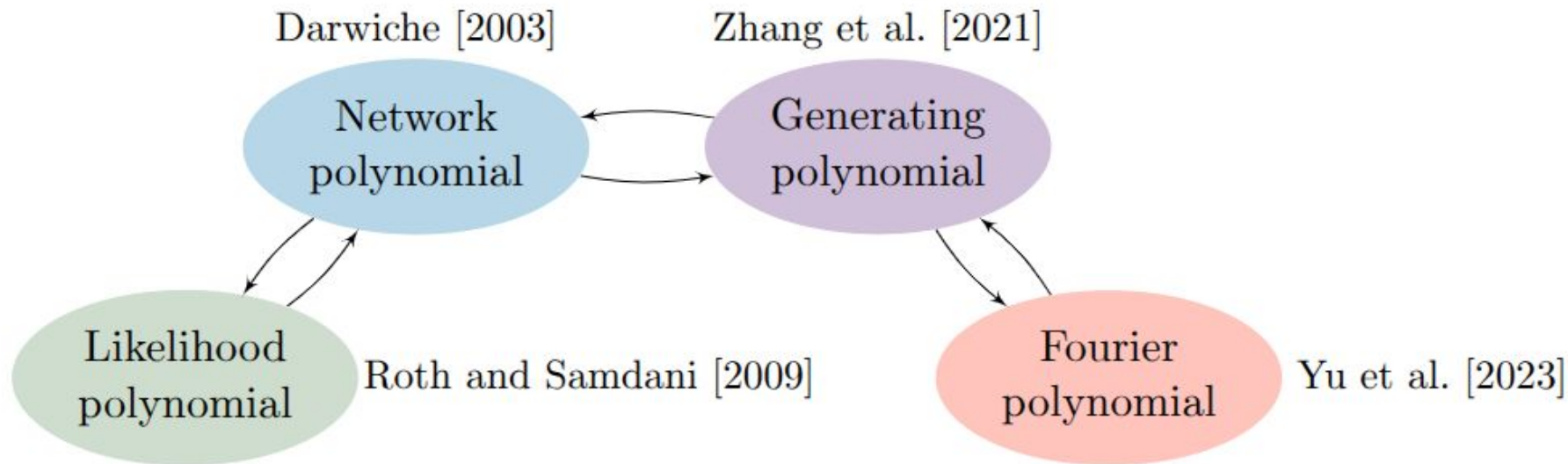
$$g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$$

Fourier  
polynomial

# Many ways to encode functions into circuit polynomials



# Many ways to encode functions into circuit polynomials



# Questions for this talk:



1. **What is the purpose of weighted model counting in the age of LLMs?**
2. Ensuing theory questions
  - a. When can we multiply circuits?
  - b. How to capture all tractable distributions?

# Reasoning about all Future Tokens: *Alignment*

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

**Prefix:** It's a pain ...

**Constraint  $\alpha$ :** non-toxic

# Reasoning about all Future Tokens: *Alignment*

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

**Prefix:** It's a pain ...

**Constraint  $\alpha$ :** non-toxic

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

in	0.3	the ass	0.3
to	0.1	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  $\alpha$ :** non-toxic

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

in	0.3
to	0.1

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

in	0.1
to	0.8





## Attribute Probability



0 (toxic)

1 (nontoxic)

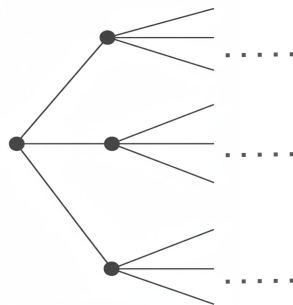
It's a pain

in

$p_{LM} = 0.3$

to

$p_{LM} = 0.1$



future text

the ass

the butt

the neck

...

...

**Intractable** to know  
expected future toxicity



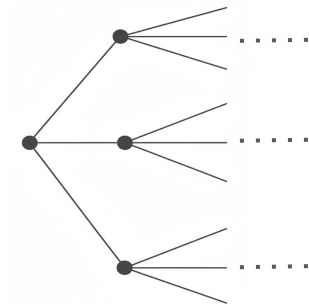
future text

deal with

handle

...

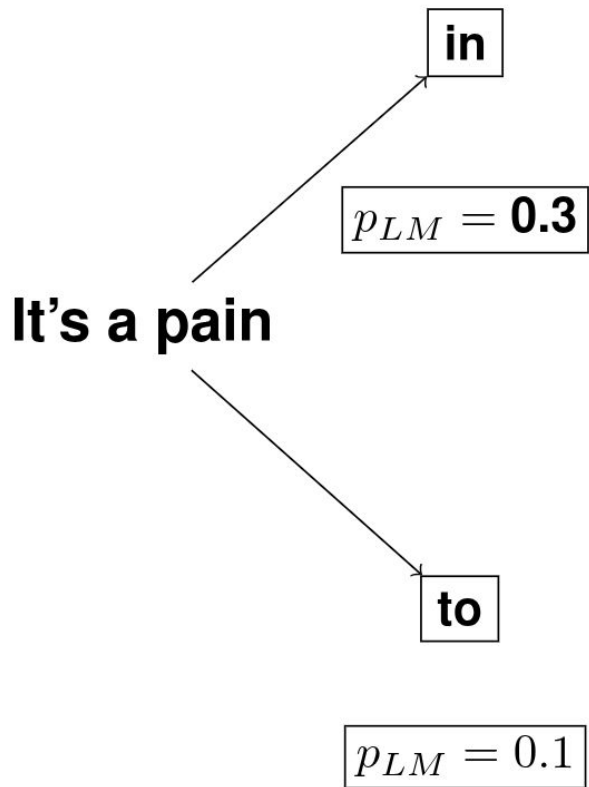
...



## Attribute Probability



0 (toxic)      1 (nontoxic)



future text

the ass

the butt

the neck

...

...

future text

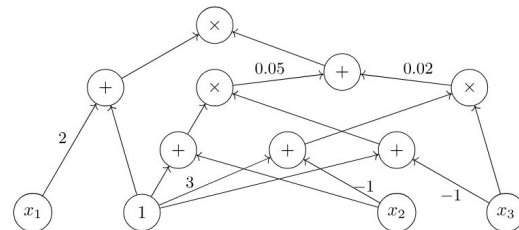
deal with

handle

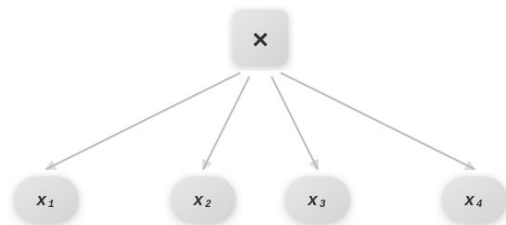
...

...

## LLM continuations with *tractable probabilistic circuit*



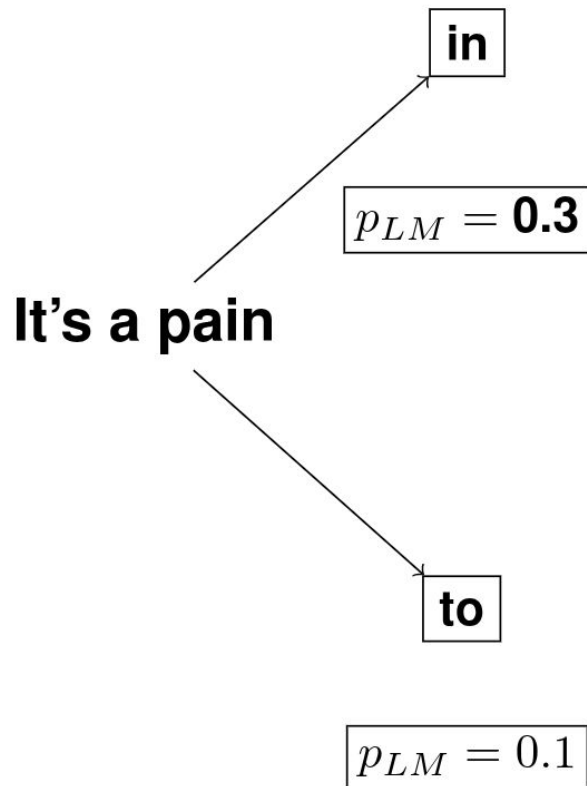
## Goal attribute with *log-linear classifier* (also a circuit)



## Attribute Probability



0 (toxic)      1 (nontoxic)



future text

the ass

the butt

the neck

...

...

$EAP = 0.1$

future text

deal with

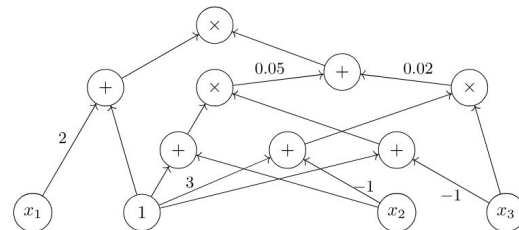
handle

...

...

$EAP = 0.8$

LLM continuations with  
*tractable probabilistic circuit*



+

Goal attribute with  
*log-linear classifier*



=

**Efficient Expected  
Attribute Probability!**



## Attribute Probability



0 (toxic)

1 (nontoxic)

It's a pain

in

$$p_{LM} = \mathbf{0.3}$$

to

$$p_{LM} = 0.1$$

future text

the ass

the butt

the neck

...

...

$$EAP = 0.1$$

$$= p_{TRACE} \propto 0.03$$

future text

deal with

handle

...

...

$$EAP = 0.8$$

$$= p_{TRACE} \propto \mathbf{0.08}$$



# State-of-the-art LLM Detoxification

Model	Toxicity (↓)		Approach Type
	avg.	max. prob.	
GPT-2 Large Results			
GPT2	0.385	0.254	Baseline
DAPT <sup>(1)</sup>	0.428	0.360	Finetuning
GeDi <sup>(2)</sup>	0.363	0.217	Decoding (Trained Guide)
FUDGE <sup>(3)</sup>	0.302	0.371	Decoding (Trained Guide)
DExperts <sup>(4)</sup>	0.314	0.128	Decoding (Trained Guide)
PPLM <sup>(5)</sup>	0.520	0.518	Decoding (Logit Control)
MuCoLa <sup>(6)</sup>	0.308	0.088	Decoding (Sampling)
PPO <sup>(7)</sup>	0.218	0.044	RL
Quark <sup>(8)</sup>	0.196	0.035	RL
DPO <sup>(9)</sup>	0.180	0.026	RL
TRACE	<b>0.163</b>	<b>0.016</b>	Decoding (HMM Reasoning)
Gemma-2B Results			
Gemma-2B	0.359	0.23	Baseline
DPO <sup>(9)</sup>	0.222	0.06	RL
TRACE	<b>0.189</b>	<b>0.02</b>	Decoding (HMM Reasoning)

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

# State-of-the-art LLM

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

Method	Entropy (↑)
GPT2-large	52.06
DPO	39.52
<b>TRACE</b>	<b>52.54</b>

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...*



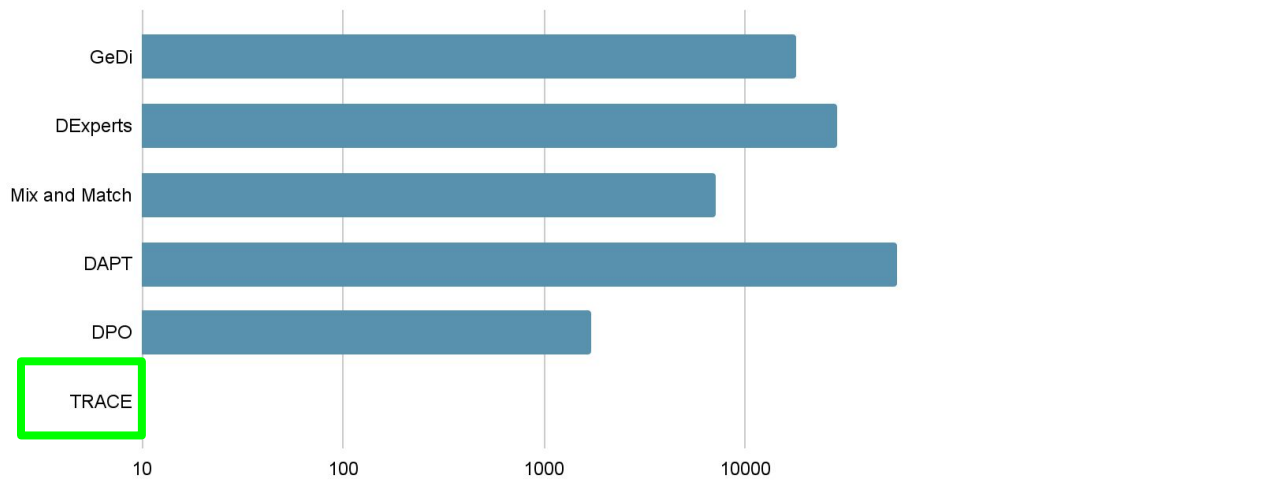
# State-of-the-art LLM Detoxification

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	<b>25.57</b>	Baseline
DAPT <sup>(1)</sup>	0.428	0.360	0.84	0.84	31.21	Finetuning
GeDi <sup>(2)</sup>	0.363	0.217	0.84	0.83	60.03	Decoding (Trained Guide)
FUDGE <sup>(3)</sup>	0.302	0.371	0.78	0.82	<del>12.97</del> *	Decoding (Trained Guide)
DExperts <sup>(4)</sup>	0.314	0.128	0.84	0.84	32.41	Decoding (Trained Guide)
PPLM <sup>(5)</sup>	0.520	0.518	0.86	0.86	32.58	Decoding (Logit Control)
MuCoLa <sup>(6)</sup>	0.308	0.088	0.82	0.83	29.92	Decoding (Sampling)
PPO <sup>(7)</sup>	0.218	0.044	0.80	0.84	<del>14.27</del> *	RL
Quark <sup>(8)</sup>	0.196	0.035	0.80	0.84	<del>12.47</del> *	RL
DPO <sup>(9)</sup>	0.180	0.026	0.76	0.78	<del>21.59</del> *	RL
<b>TRACE</b>	<b>0.163</b>	<b>0.016</b>	0.85	0.85	29.83	Decoding (HMM Reasoning)
Gemma-2B Results						
Gemma-2B	0.359	0.23	0.86	0.85	<b>15.75</b>	Baseline
DPO <sup>(9)</sup>	0.222	0.06	0.74	0.77	<del>14.39</del> *	RL
<b>TRACE</b>	<b>0.189</b>	<b>0.02</b>	<b>0.86</b>	<b>0.85</b>	17.68	Decoding (HMM Reasoning)

# TRACE is Blazingly Fast

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

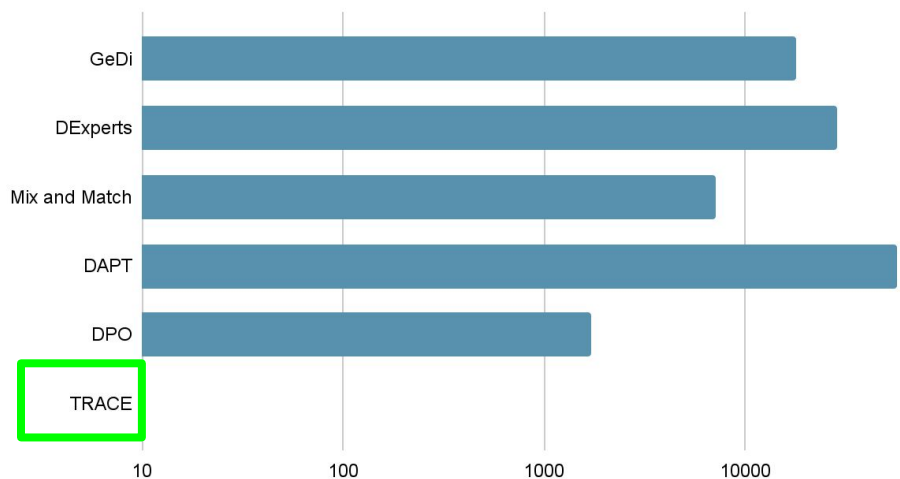
Training Time per Attribute (seconds)



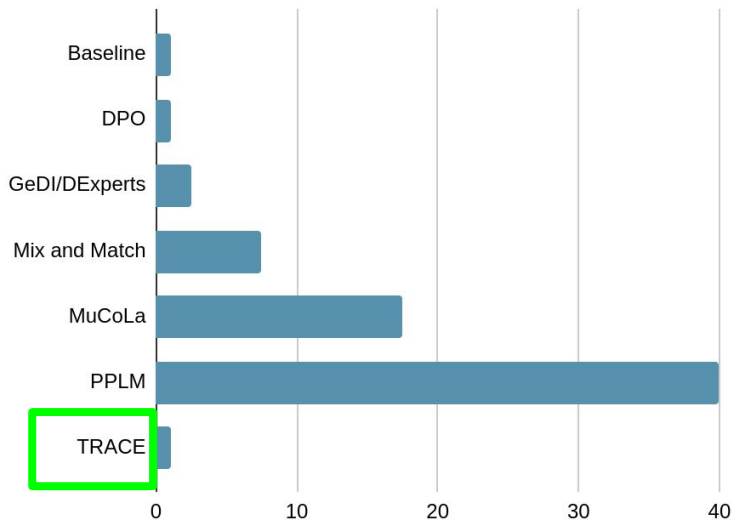
# TRACE is Blazingly Fast

Given a language model, and its tractable circuit twin,  
train log-linear attribute classifier,  
then use Bayesian logits at decoding time (**weighted model counting**)

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



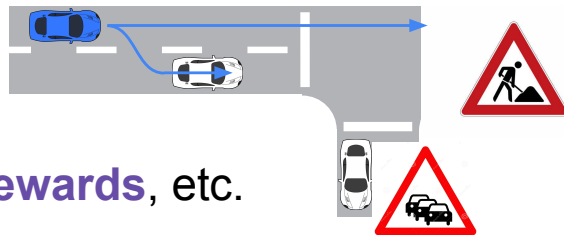
Prompt

How is the weather?

Generation

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

# 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**.



Reward:  $\sum_{t' \geq t} R_{t'} \geq \text{threshold}$

State:  $\text{state}_t \in \text{safe states}$

Action:  $\text{action}_t \in \text{safe actions}$

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

# Reasoning about all Future Tokens: *Offline RL*



Reward:  $\sum_{t' \geq t} \text{R}_{t'} \geq \text{threshold}$

State: state<sub>t</sub>  $\in$  safe states

Action: action<sub>t</sub>  $\in$  safe actions

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

$$\begin{aligned}
 & p(\text{action}_t \mid \text{state}_{\leq t}, \text{action}_{< t}, \text{Constraints}) \\
 \propto & \underbrace{p(\text{action}_t \mid \text{state}_{\leq t}, \text{action}_{< t})}_{\text{Autoregressive Transformers (GPTs)}} \cdot \underbrace{p(\text{Constraints} \mid \text{state}_{\leq t}, \text{action}_{< t})}_{\text{Probabilistic Circuits (PCs)}}
 \end{aligned}$$

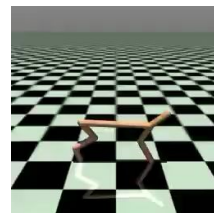
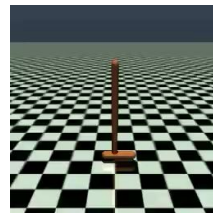
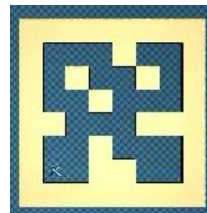
*Bayes' rule*



# 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	<b>95.1</b> $\pm$ 0.3	82.3 $\pm$ 6.1	<b>89.9</b> $\pm$ 4.6	86.8 $\pm$ 1.3	<b>91.9</b> $\pm$ 1.9	90.6	86.7	91.6	92.9	90.7
Med-Expert	Hopper	110.0 $\pm$ 2.7	<b>113.0</b> $\pm$ 0.4	74.7 $\pm$ 6.3	<b>78.5</b> $\pm$ 6.4	107.6 $\pm$ 1.8	/	111.8	91.5	105.4	110.9	98.0
Med-Expert	Walker2d	101.9 $\pm$ 6.8	<b>109.3</b> $\pm$ 0.1	109.3 $\pm$ 2.3	<b>109.6</b> $\pm$ 0.2	108.1 $\pm$ 0.2	<b>108.6</b> $\pm$ 0.3	108.8	<b>109.6</b>	108.8	109.0	110.1
Medium	HalfCheetah	46.9 $\pm$ 0.4	<b>49.5</b> $\pm$ 0.2	48.7 $\pm$ 0.3	<b>48.9</b> $\pm$ 0.3	42.6 $\pm$ 0.1	<b>44.2</b> $\pm$ 0.7	49.1	47.4	44.0	42.5	48.3
Medium	Hopper	61.1 $\pm$ 3.6	<b>67.1</b> $\pm$ 4.3	55.2 $\pm$ 3.8	<b>57.8</b> $\pm$ 1.9	67.6 $\pm$ 1.0	/	<b>79.3</b>	66.3	58.5	56.9	59.3
Medium	Walker2d	79.0 $\pm$ 2.8	<b>83.1</b> $\pm$ 0.8	82.2 $\pm$ 2.5	<b>84.7</b> $\pm$ 1.9	74 $\pm$ 1.4	<b>81.3</b> $\pm$ 2.3	<b>82.5</b>	78.3	72.5	75.0	83.7
Med-Replay	HalfCheetah	41.9 $\pm$ 2.5	<b>45.0</b> $\pm$ 0.3	48.2 $\pm$ 0.4	<b>48.9</b> $\pm$ 0.3	36.6 $\pm$ 0.8	<b>39.2</b> $\pm$ 0.4	39.3	44.2	45.5	40.6	44.6
Med-Replay	Hopper	91.5 $\pm$ 3.6	<b>97.8</b> $\pm$ 0.3	83.4 $\pm$ 5.6	<b>87.6</b> $\pm$ 6.1	82.7 $\pm$ 7.0	/	<b>100.0</b>	94.7	95.0	75.9	60.9
Med-Replay	Walker2d	82.6 $\pm$ 6.9	<b>88.3</b> $\pm$ 3.8	84.6 $\pm$ 4.5	<b>90.6</b> $\pm$ 4.2	66.6 $\pm$ 3.0	<b>73.5</b> $\pm$ 0.1	<b>75.0</b>	73.9	77.2	62.5	81.8
Average Score		78.9	<b>83.1</b>	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	<b>-57</b>	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	<b>81.9</b> $\pm$ 4.8	77.8 $\pm$ 5.4
Med-Expert	Hopper	<b>109.6</b> $\pm$ 2.4	100.0 $\pm$ 4.2
Med-Expert	Walker2d	<b>105.1</b> $\pm$ 2.3	103.6 $\pm$ 4.9

# Questions for this talk:



1. What is the purpose of weighted model counting in the age of LLMs?
2. Ensuing theory questions
  - a. When can we multiply circuits?
  - b. How to capture all tractable distributions?



# Thanks

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



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