



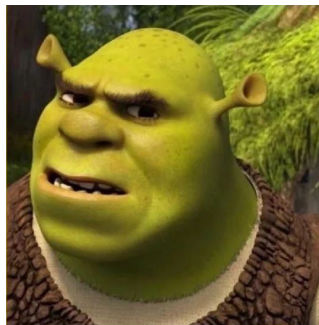
Computer  
Science



# Symbolic Reasoning About Large Language Models

Guy Van den Broeck

Simons Institute Workshop on Theoretical Aspects of Trustworthy AI - Apr 29 2025

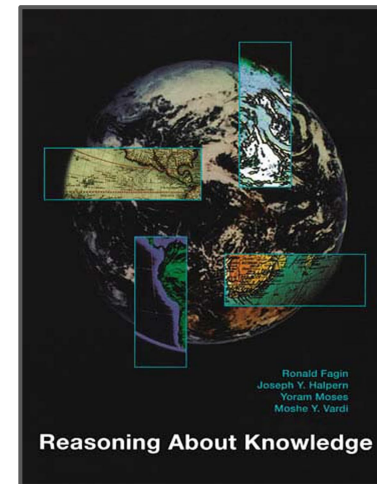


## Reasoning with Symbolic AI

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

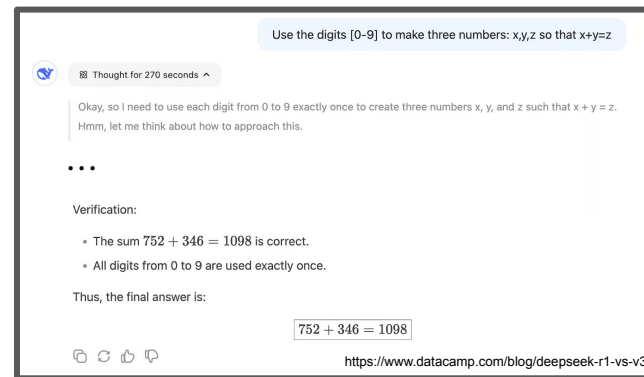


30 years



## Reasoning with Transformers

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



# Questions for this talk:

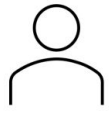


1. Do deductive reasoning algorithms still have a purpose in the age of transformers?
2. Where did reasoning algorithms go wrong?  
What should they look like today?
3. Can reasoning algorithms provide a path to language model alignment, safety?

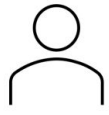
# Questions for this talk:



- 1. Do deductive reasoning algorithms still have a purpose in the age of transformers?**
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", following the given order.

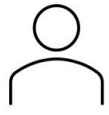


Generate a sentence using "frisbee", "caught" and "dog", following the given order.



After a perfect throw, the frisbee glided through the air, and the dog, with incredible agility, caught it mid-flight.

*ChatGPT*

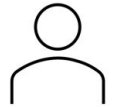


Generate a sentence using "frisbee", "caught" and "dog", following the given order.



After a perfect throw, the frisbee glided through the air, and the dog, with incredible agility, caught it mid-flight.

ChatGPT



That's not correct. Generate a sentence using "frisbee", "caught" and "dog". The keywords should appear in the order as specified.



Here's the correct sentence: The dog caught the frisbee in mid-air, showing off its amazing catching skills.

ChatGPT



A frisbee is caught by a dog.

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

Ctrl-G

# The Ctrl-G Architecture

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

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



# The Ctrl-G Architecture

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

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

Pre-trained  
Language Model



$x_{t+1}$	$\Pr_{LM}(x_{t+1}   x_{1:t})$
cold	0.05
warm	0.10

# The Ctrl-G Architecture

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

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

Pre-trained  
Language Model

$x_{t+1}$	$\Pr_{LM}(x_{t+1}   x_{1:t})$
cold	0.05
warm	0.10



*Using Bayes rule,*

$$p_{LM}(\text{next-token} | \alpha, \text{prefix})$$

$\propto$

$$p_{LM}(\text{next-token} | \text{prefix})$$

$$\cdot p_{LM}(\alpha | \text{next-token}, \text{prefix})$$

# The Ctrl-G Architecture

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

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

**✗ intractable**

Pre-trained  
Language Model

$x_{t+1}$	$\Pr_{LM}(x_{t+1}   x_{1:t})$
cold	0.05
warm	0.10



Using Bayes rule,

$$p_{LM}(\text{next-token} | \alpha, \text{prefix})$$

$\propto$

$$p_{LM}(\text{next-token} | \text{prefix})$$

$$\cdot p_{LM}(\alpha | \text{next-token}, \text{prefix})$$

**Intractable**



# The Ctrl-G Architecture

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

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

**✗ intractable**

Pre-trained  
Language Model

$x_{t+1}$	$\Pr_{LM}(x_{t+1}   x_{1:t})$
cold	0.05
warm	0.10

**✓ efficient**

Tractable  
Probabilistic Model

$x_{t+1}$	$\Pr_{TPM}(\alpha   x_{t+1}, x_{1:t})$
cold	0.50
warm	0.01



Using Bayes rule,

$$p_{LM}(\text{next-token} | \alpha, \text{prefix})$$

$\propto$

$$p_{LM}(\text{next-token} | \text{prefix})$$

$$\cdot p_{LM}(\alpha | \text{next-token}, \text{prefix})$$

**Intractable**



# The Ctrl-G Architecture

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

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

**✗ intractable**

Pre-trained  
Language Model

$x_{t+1}$	$\Pr_{LM}(x_{t+1}   x_{1:t})$
cold	0.05
warm	0.10

**✓ efficient**

Tractable  
Probabilistic Model

$x_{t+1}$	$\Pr_{TPM}(\alpha   x_{t+1}, x_{1:t})$
cold	0.50
warm	0.01

$x_{t+1}$	$p(x_{t+1}   \alpha, x_{1:t})$
cold	0.025
warm	0.001



*Abusing Bayes rule,*

$$p_{CTRL-G}(\text{next-token} | \alpha, \text{prefix})$$

$\propto$

$$p_{LM}(\text{next-token} | \text{prefix})$$

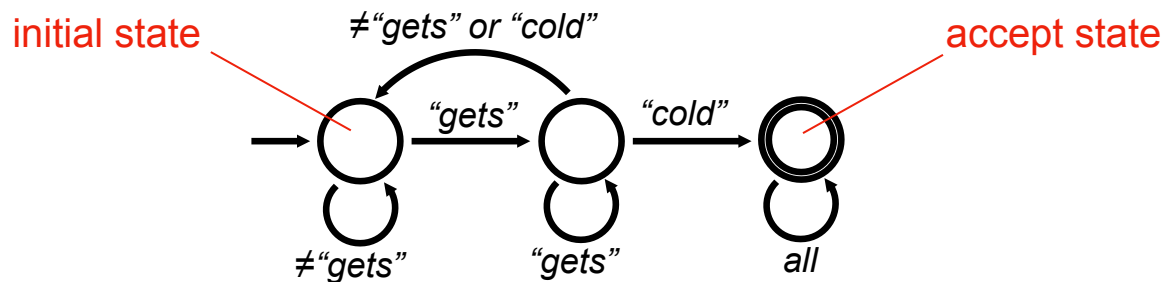
$$\cdot p_{TPM}(\alpha | \text{next-token}, \text{prefix})$$



# 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*

*Exactly  $k$  times.*

*Anything over fixed sequence lengths (BDD)*

*Must end a certain way*

*From a restricted vocabulary.*

*Any regex*

*...*

# 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."

# 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."



5 lines of code!

```
from CtrlG import *

prefix = "First they defeated a ..."
suffix = "are few humans left ..."

dfa_list = [
    DFA_all_of("alien mothership",
               "far from over"),
    DFA_word_count(25, 30),
]
dfa = DFA_logical_and(dfa_list)

lp = CtrlGLogitsProcessor(
    dfa, hmm, prefix, suffix)
llm.generate(logits_processor=lp)
```



# 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."

```
from CtrlG import *  
  
prefix = "First they defeated a ..."  
suffix = "are few humans left ..."  
  
dfa_list = [  
    DFA_all_of("alien mothership",  
              "far from over"),  
    DFA_word_count(25, 30),  
]  
dfa = DFA_logical_and(dfa_list)  
  
lp = CtrlGLogitsProcessor(  
    dfa, hmm, prefix, suffix)  
llm.generate(logits_processor=lp)
```

5 lines of code!

"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

CoAuthor



	<i>None</i>	<i>K</i>	<i>L</i>	<i>K&amp;L</i>
--	-------------	----------	----------	----------------

<i>Quality</i>				
----------------	--	--	--	--

TULU2	2.68	2.64	2.78	2.74
-------	------	------	------	------

GPT3.5	2.27	2.22	2.27	2.31
--------	------	------	------	------

GPT4	<b>3.79</b>	3.33	3.53	3.10
------	-------------	------	------	------

Ctrl-G	<b>3.77</b>	<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

CoAuthor



	<i>None</i>	<i>K</i>	<i>L</i>	<i>K&amp;L</i>
--	-------------	----------	----------	----------------

<i>Quality</i>				
----------------	--	--	--	--

TULU2	2.68	2.64	2.78	2.74
-------	------	------	------	------

GPT3.5	2.27	2.22	2.27	2.31
--------	------	------	------	------

GPT4	<b>3.79</b>	3.33	3.53	3.10
------	-------------	------	------	------

Ctrl-G	<b>3.77</b>	<b>3.56</b>	<b>3.73</b>	<b>3.59</b>
--------	-------------	-------------	-------------	-------------

→ *How many stars by humans?*

<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>
--------	---	-------------	-------------	-------------

→ *Follows instructions?*

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



	None	K	L	K&L
<i>Quality</i>				
TULU2	2.68	2.64	2.78	2.74
GPT3.5	2.27	2.22	2.27	2.31
GPT4	<b>3.79</b>	3.33	3.53	3.10
Ctrl-G	<b>3.77</b>	<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**

# 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:  
for any next-token  $x_{t+1}$  that would make  $\alpha$  unsatisfiable,  $p(x_{t+1} \mid x_{1:t}, \alpha) = 0$ .
2. Generalizes well to unseen reasoning tasks, because all tasks are unseen :-)  
(baselines train on a distribution over reasoning tasks – slow and brittle!)

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



# Questions for this talk:



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

# Probabilistic Reasoning Task

Marginal inference:

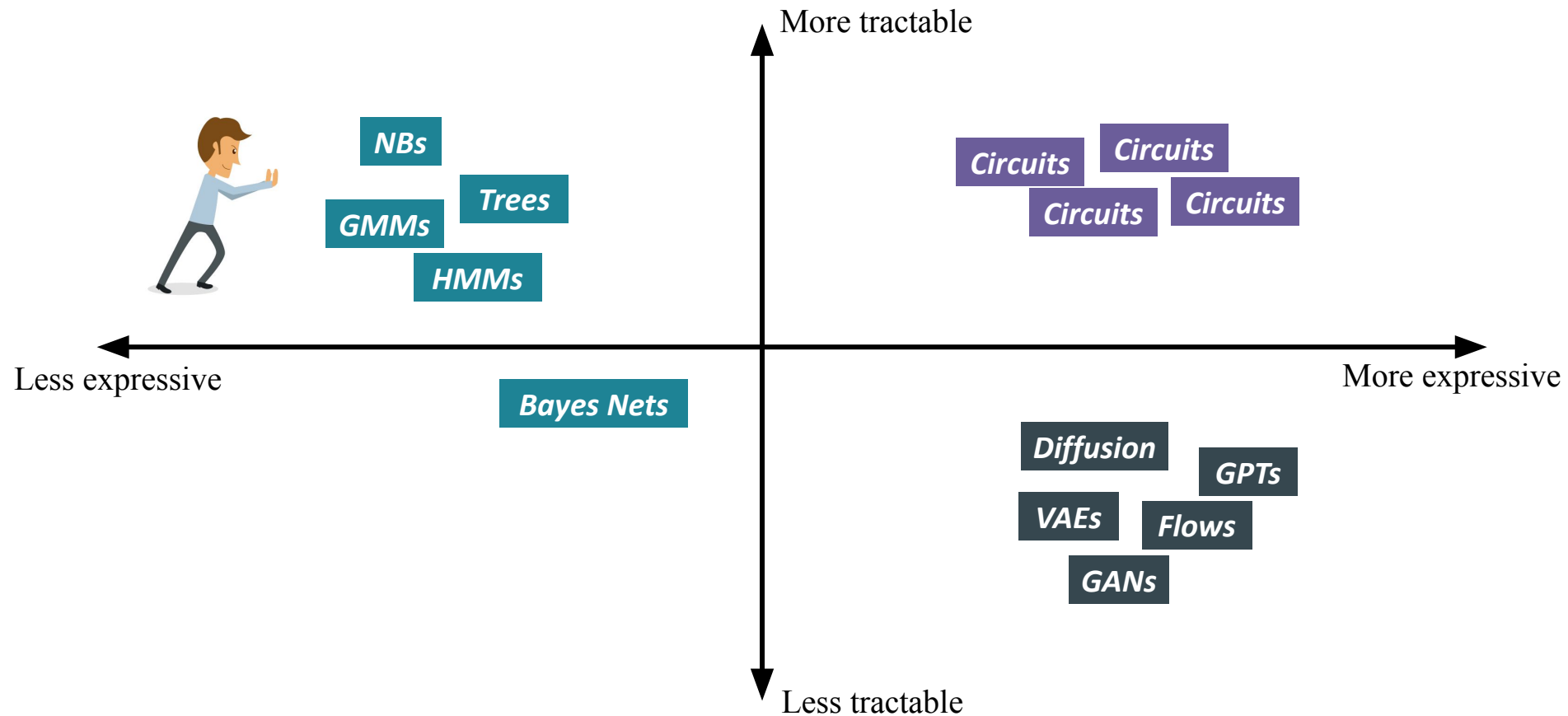
$X_1$	$X_2$	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

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

Application: Ctrl-G



$$\Pr(\text{next-token}|\text{prefix}, \alpha) \propto \sum_{\text{text}} \Pr(\text{next-token}, \text{text}, \text{prefix}, \alpha)$$



# Generative Models

polynomials model joint distributions

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

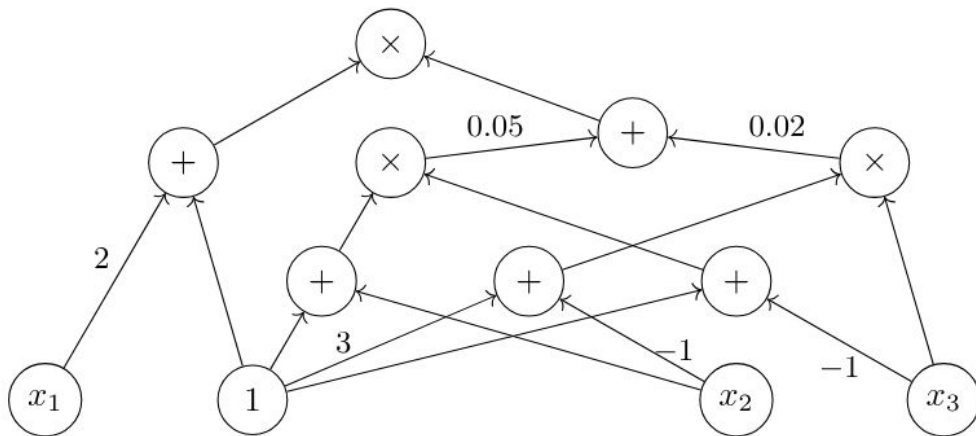
$X_1$	$X_2$	$X_3$	$p$
0	0	0	0.05
1	0	0	0.15
0	1	0	0.1
1	1	0	0.3
0	0	1	0.06
1	0	1	0.18
0	1	1	0.04
1	1	1	0.12

# Deep Generative Models

circuit polynomials model **joint distributions** compactly

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

$X_1$	$X_2$	$X_3$	$p$
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

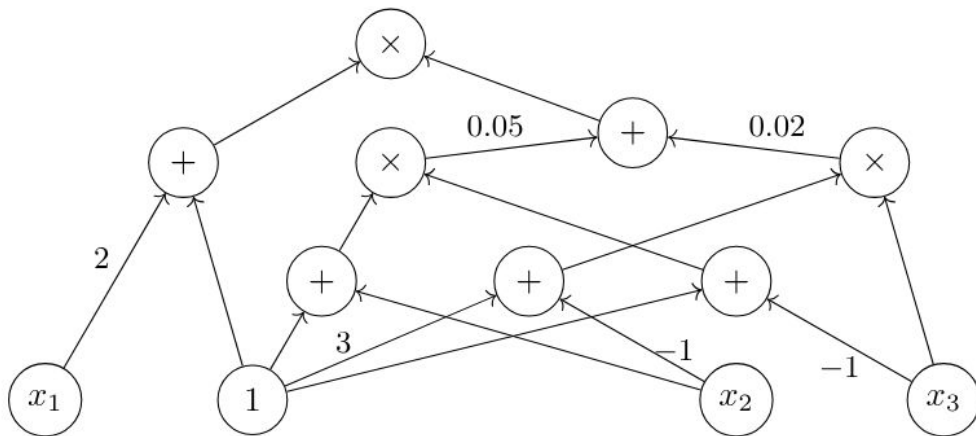


# Tractable Deep Generative Models

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

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

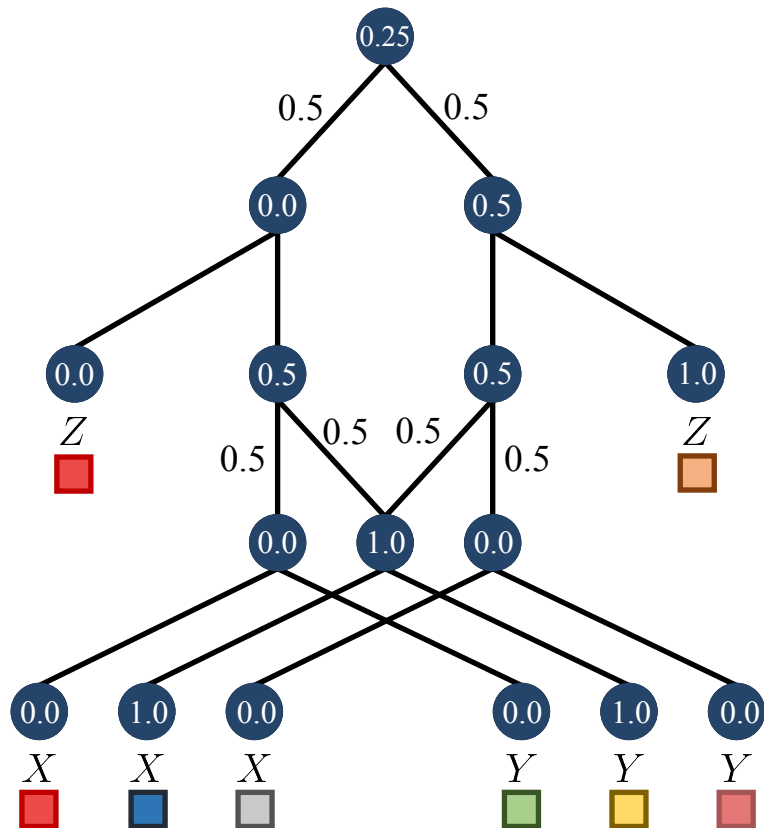
$X_1$	$X_2$	$X_3$	$p$
0	0	0	0.05
1	0	0	0.15
0	1	0	0.1
1	1	0	0.3
0	0	1	0.06
1	0	1	0.18
0	1	1	0.04
1	1	1	0.12



# Compute Likelihood

Compute  $p(x = \blacksquare, y = \blacksquare, z = \blacksquare) = 0.25$

- Readout likelihood from the **output node**.
- Compute the likelihood of every **sum/product node**.
- Compute the likelihood of every **input node**.



# Computing Marginals

Compute  $p(x = \blacksquare) = \iint p(x = \blacksquare, y, z) dydz$

- Sum node  $\oplus_a$**

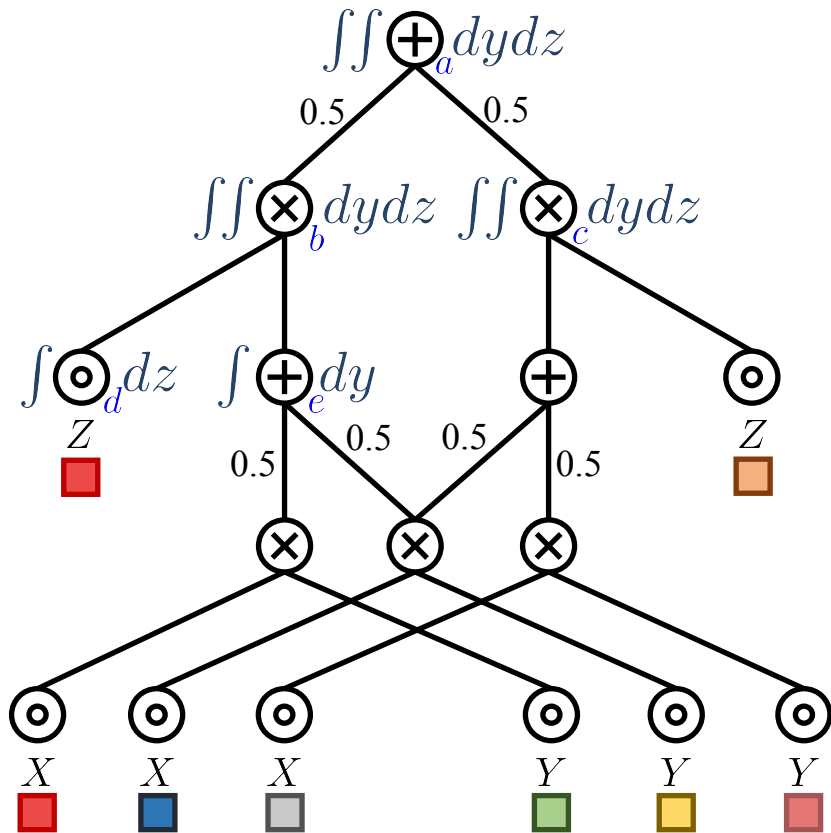
$$\begin{aligned} & \iint p_a(x = \blacksquare, y, z) dydz \\ &= \iint 0.5 \cdot p_b(x = \blacksquare, y, z) + 0.5 \cdot p_c(x = \blacksquare, y, z) dydz \\ &= 0.5 \underbrace{\iint p_b(x = \blacksquare, y, z) dydz}_{\iint \otimes_b dydz} + 0.5 \underbrace{\iint p_c(x = \blacksquare, y, z) dydz}_{\iint \otimes_c dydz} \end{aligned}$$

- Product node  $\otimes_b$**

$$\begin{aligned} & \iint p_b(x = \blacksquare, y, z) dydz \\ &= \iint p_d(z) \cdot p_e(x = \blacksquare, y) dydz \\ &= \underbrace{\int p_d(z) dz}_{\int \otimes_d dz} \cdot \underbrace{\int p_e(x = \blacksquare, y) dy}_{\int \otimes_e dy} \end{aligned}$$

- Input node  $\odot_d$**

$$\int p_d(z) = 1$$



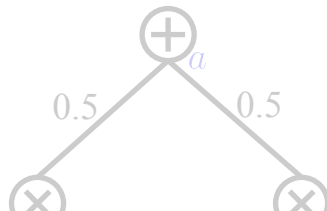


# Computing Marginals

Compute  $p(x = \square) = \iint p(x = \square, y, z) dy dz$

- Sum node  $\oplus_a$

$$\iint p_a(x = \square, y, z) dy dz$$



**Theorem.** Given

1. a DFA constraint  $\alpha$  with  $m$  edges and
  2. a PC  $p(\cdot)$  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.

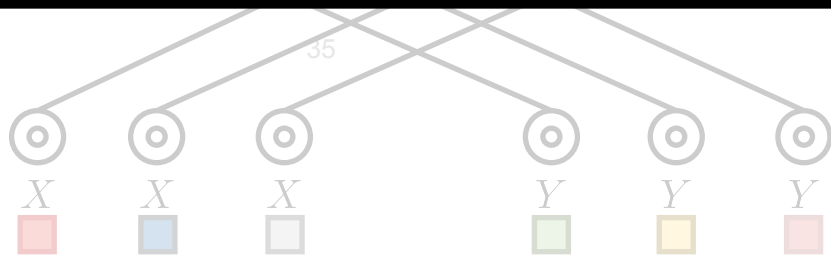


$$= \iint p_d(z) \cdot p_e(x = \square, y) dy dz$$

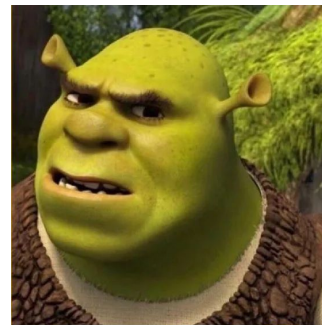
$$= \underbrace{\int p_d(z) dz}_{\int \otimes_d dz} \cdot \underbrace{\int p_e(x = \square, y) dy}_{\int \otimes_e dy}$$

- Input node  $\odot_d$

$$\int p_d(z) = 1$$



# 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) **marginals** for a family of distributions, then there exist **poly-size circuits** for their **multilinear** polynomials.*



# An Open-Source Package: PyJuice

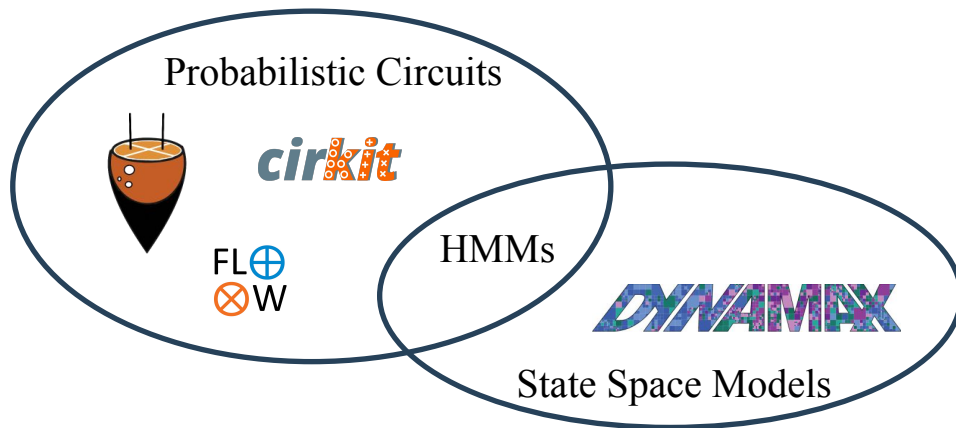


Runtime (in seconds) for training on **60K** samples

PD (Poon & Domingos, 2011)					
# nodes	172K	344K	688K	1.38M	2.06M
# edges	15.6M	56.3M	213M	829M	2.03B
SPFlow	>25000	>25000	>25000	>25000	>25000
EiNet	34.2±0.0	88.7±0.2	456.1±2.3	1534.7±0.5	OOM
Juice.jl	12.6±0.5	37.0±1.7	141.7±6.9	OOM	OOM
PyJuice	<b>2.0±0.0</b>	<b>5.3±0.0</b>	<b>15.4±0.0</b>	<b>57.1±0.2</b>	<b>203.7±0.1</b>
RAT-SPN (Peharz et al., 2020b)					
# nodes	58K	116K	232K	465K	930K
# edges	616K	2.2M	8.6M	33.4M	132M
SPFlow	6372.1±4.2	>25000	>25000	>25000	>25000
EiNets	38.5±0.0	83.5±0.0	193.5±0.1	500.6±0.2	2445.1±2.6
Juice.jl	6.0±0.3	9.4±0.3	25.5±2.4	84.0±4.0	375.1±3.4
PyJuice	<b>0.6±0.0</b>	<b>0.9±0.1</b>	<b>1.6±0.0</b>	<b>5.8±0.1</b>	<b>13.8±0.0</b>
HCLT (Liu & Van den Broeck, 2021)					
# nodes	89K	178K	355K	710K	1.42M
# edges	2.56M	10.1M	39.9M	159M	633M
SPFlow	22955.6±18.4	>25000	>25000	>25000	>25000
EiNet	52.5±0.3	77.4±0.4	233.5±2.8	1170.7±8.9	5654.3±17.4
Juice.jl	4.7±0.2	6.4±0.5	12.4±1.3	41.1±0.1	143.2±5.1
PyJuice	<b>0.8±0.0</b>	<b>1.3±0.0</b>	<b>2.6±0.0</b>	<b>8.8±0.0</b>	<b>24.9±0.1</b>
HMM (Rabiner & Juang, 1986)					
# nodes	33K	66K	130K	259K	388K
# edges	8.16M	32.6M	130M	520M	1.17B
Dynamax	111.3±0.4	441.2±3.9	934.7±6.3	2130.5±19.5	4039.8±38.3
Juice.jl	4.6±0.1	18.8±0.1	91.6±0.1	OOM	OOM
PyJuice	<b>0.6±0.0</b>	<b>1.0±0.0</b>	<b>2.9±0.1</b>	<b>10.1±0.2</b>	<b>39.9±0.1</b>

- Orders of magnitude **faster**!
- Extremely **scalable**!

Custom data structure +  
CUDA kernels



FL⊕W by Cambridge, TU Darmstadt, Max-Planck-Institute et al.

cirkkit by Edinburgh, EPFL et al.

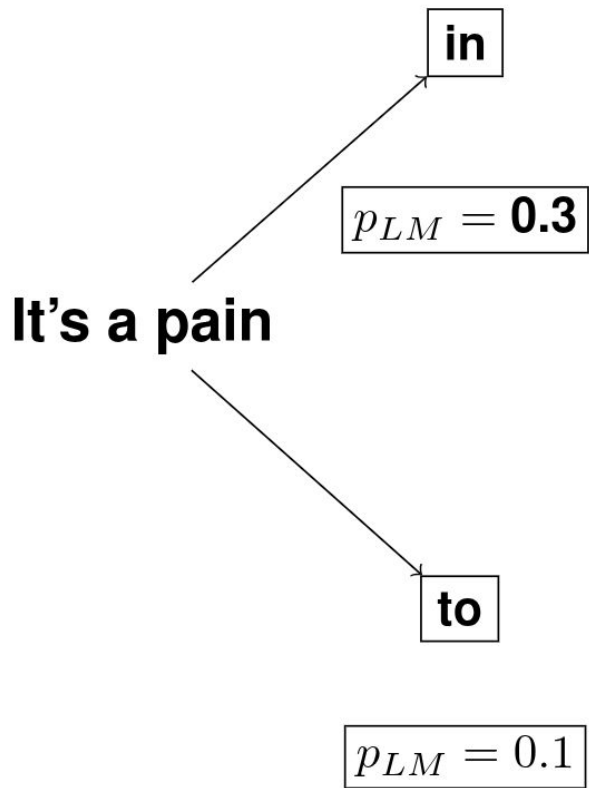
DYNAMAX by Google Deepmind et al.

<https://github.com/Tractables/pyjuice>

# Questions for this talk:



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



Attribute Probability



0 (toxic)

1 (nontoxic)

- No longer a logical constraint (no DFA)
- A “soft” **attribute** with some probability
- a.k.a. an exponentiated *reward function*

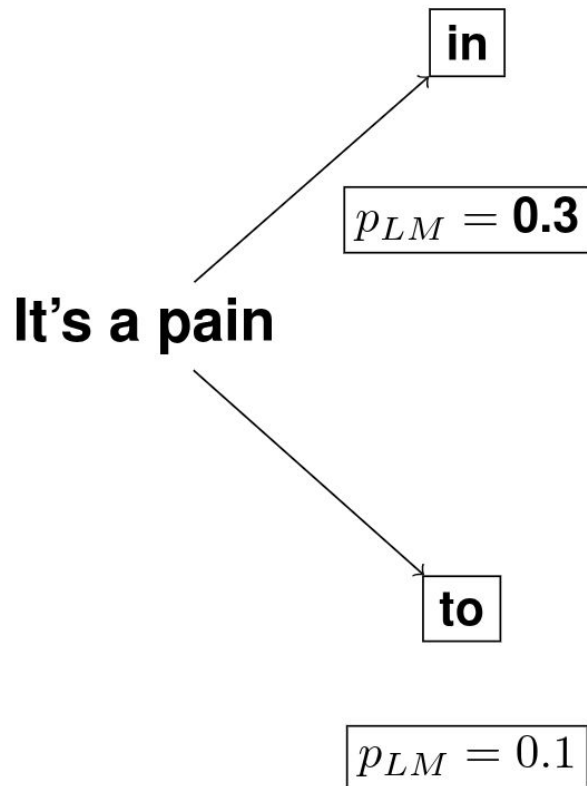


## Attribute Probability



0 (toxic)

1 (nontoxic)



future text	$p_{LM}(x_{>t} \mid x_{\leq t})$
the ass	0.3
the butt	0.15
the neck	0.05
...	...
...	...

Intractable to know future  
expected attribute probability (EAP)

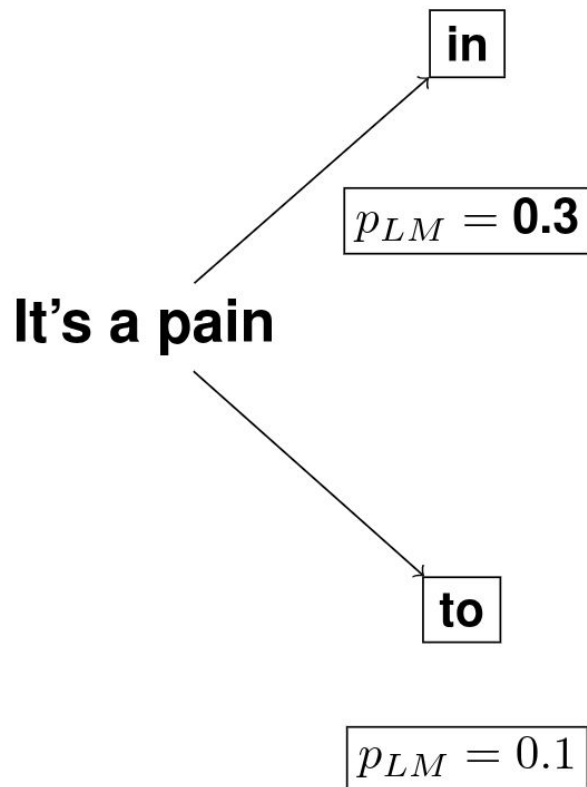


future text	$p_{LM}(x_{>t} \mid x_{\leq t})$
deal with	0.2
handle	0.1
...	...
...	...

## Attribute Probability



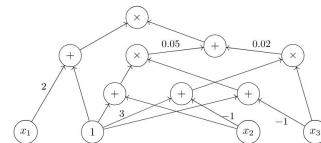
0 (toxic)      1 (nontoxic)



future text	$p_{TPM}(x_{>t} \mid x_{\leq t})$
the ass	0.3
the butt	0.15
the neck	0.05
...	...
...	...

future text	$p_{TPM}(x_{>t} \mid x_{\leq t})$
deal with	0.2
handle	0.1
...	...
...	...

Tractable  
Probabilistic Model



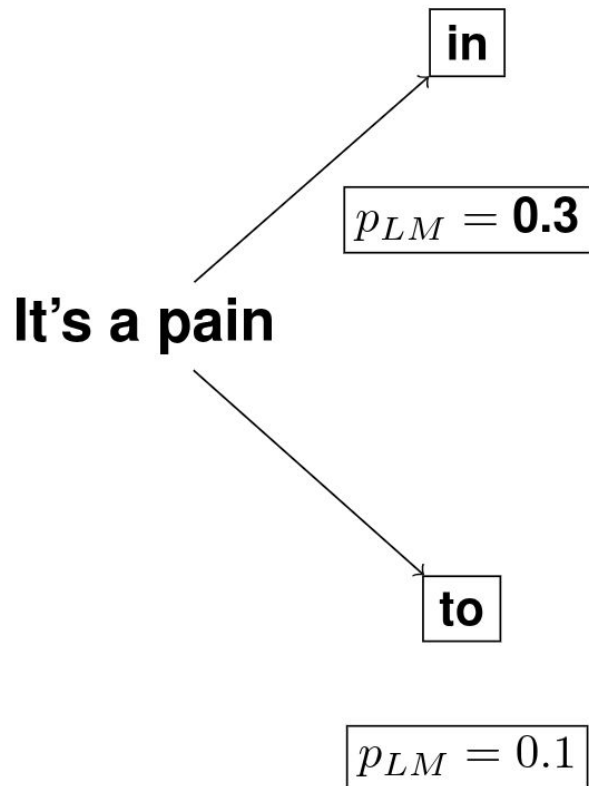
+ Log-Linear  
Attribute Classifier



# Attribute Probability



0 (toxic)      1 (nontoxic)



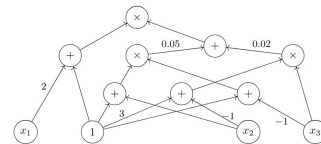
future text	$p_{TPM}(x_{>t} \mid x_{\leq t})$
the ass	0.3
the butt	0.15
the neck	0.05
...	...
...	...

$EAP = 0.1$

future text	$p_{TPM}(x_{>t} \mid x_{\leq t})$
deal with	0.2
handle	0.1
...	...
...	...

$EAP = 0.8$

Tractable  
Probabilistic Model



+ Log-Linear  
Attribute Classifier



=

Efficient Expected  
Attribute Probability!





# Attribute Probability



0 (toxic)

1 (nontoxic)

It's a pain

in

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

future text	$p_{TPM}(x_{>t} \mid x_{\leq t})$
the ass	0.3
the butt	0.15
the neck	0.05
...	...
...	...

$$EAP = 0.1$$

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

to

$$p_{LM} = 0.1 \times$$

future text	$p_{TPM}(x_{>t} \mid x_{\leq t})$
deal with	0.2
handle	0.1
...	...
...	...

$$EAP = 0.8$$

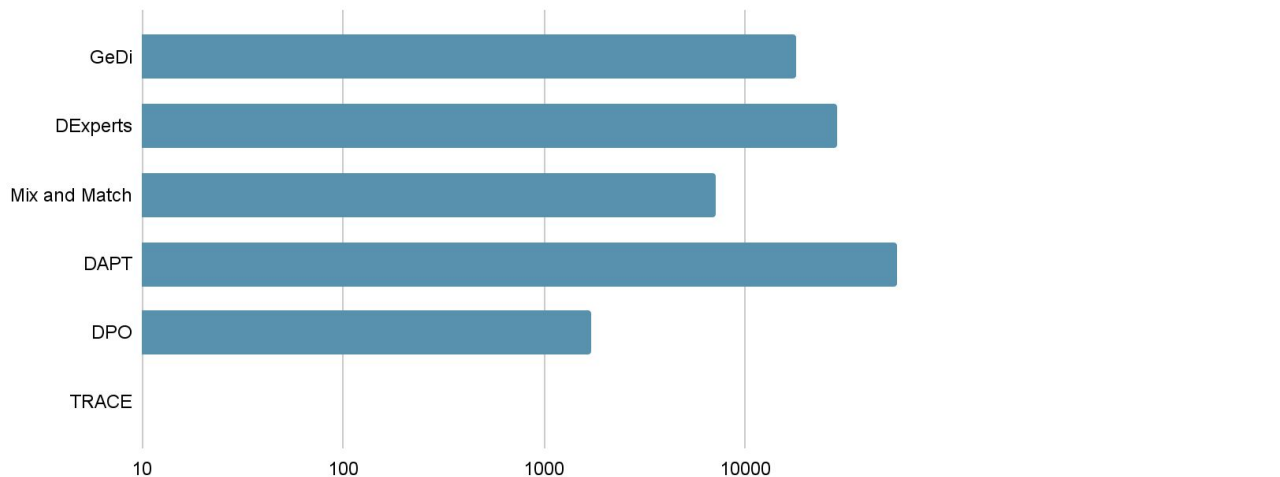
$$= p_{TRACE} \propto \mathbf{0.08}$$



# TRACE is Blazingly Fast

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

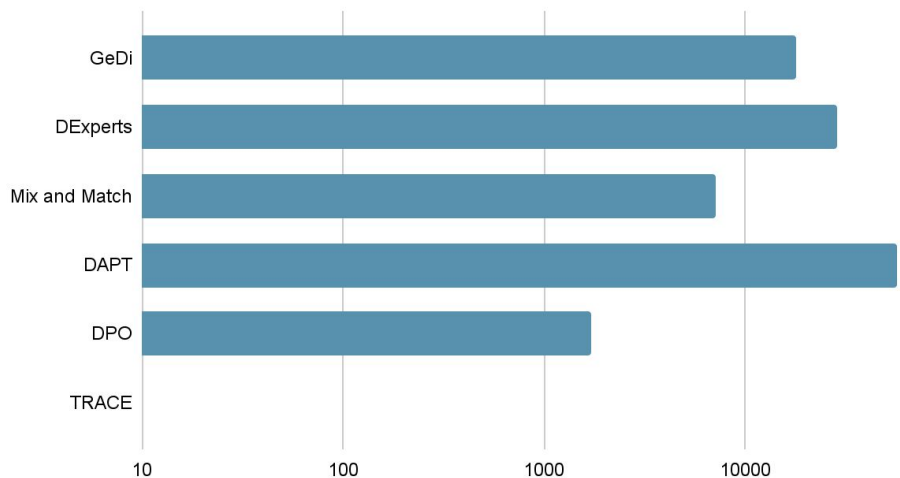
Training Time per Attribute (seconds)



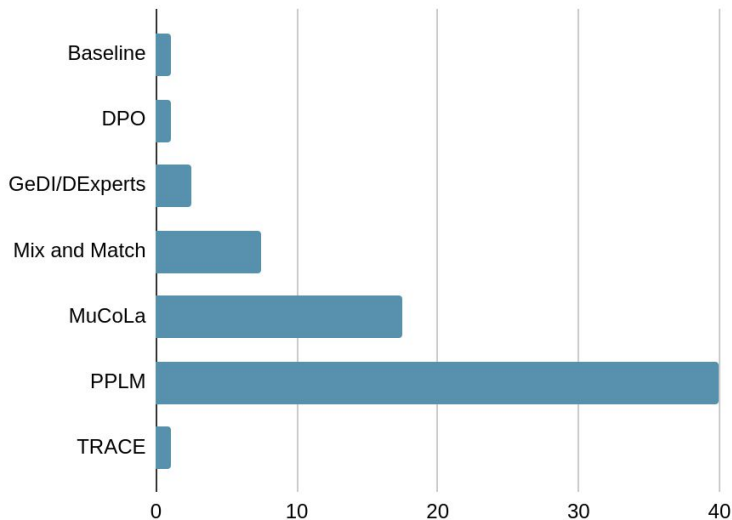
# TRACE is Blazingly Fast

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

Training Time per Attribute (seconds)



Inference Time



# 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)

# State-of-the-art LLM Detoxification

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

# 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
TRACE	<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
TRACE	<b>0.189</b>	<b>0.02</b>	<b>0.86</b>	<b>0.85</b>	17.68	Decoding (HMM Reasoning)

# 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?

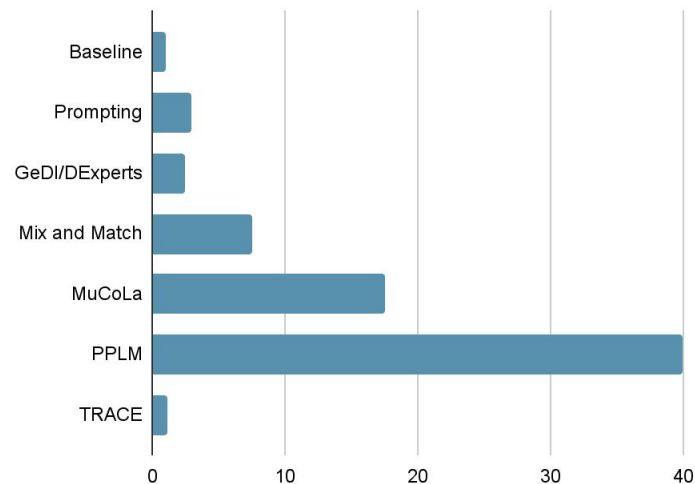
Generation

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

# 76 Personalized Language Models

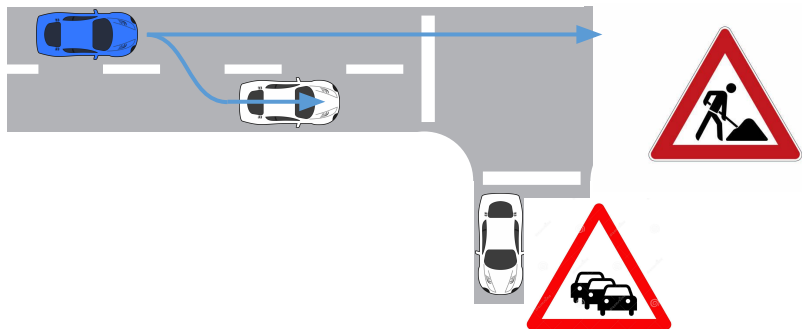


Inference Time





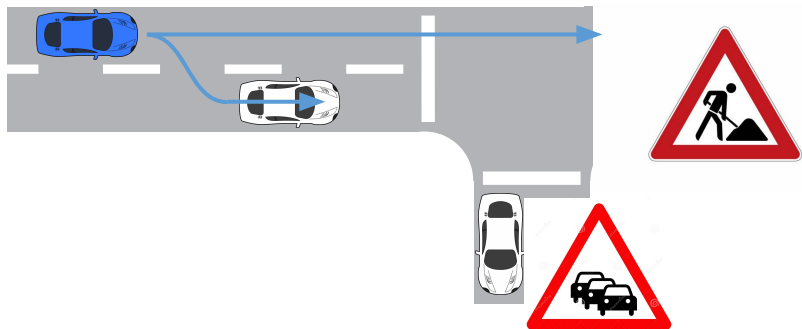
# Offline RL by Tractable Conditioning



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



# Offline RL by Tractable Conditioning

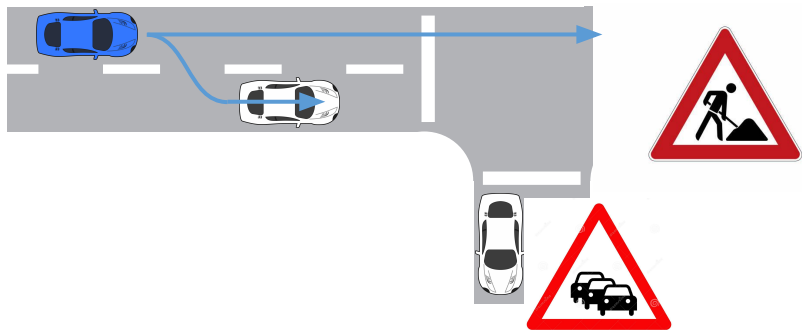


**Training:** model the joint distribution over **states**, **actions**, **rewards**, etc.

**Inference:** sample actions condition on past **states** and **actions**,



# Offline RL by Tractable Conditioning



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

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



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

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

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

# Offline RL by Tractable Conditioning



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**.

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

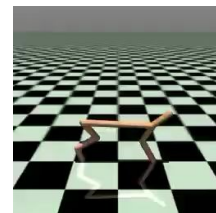
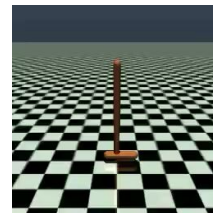
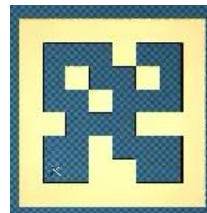
*Bayes' rule*



# 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

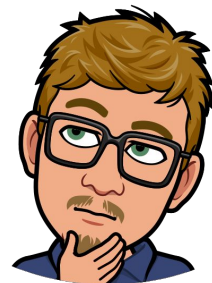
Dataset	Environment	Trifle	TT
Med-Expert	Halfcheetah	<b>81.9</b> $\pm$ 4.8	77.8 $\pm$ 5.4
Med-Expert	Hopper	<b>109.6</b> $\pm$ 2.4	100.0 $\pm$ 4.2
Med-Expert	Walker2d	<b>105.1</b> $\pm$ 2.3	103.6 $\pm$ 4.9

# Conclusions for this talk:

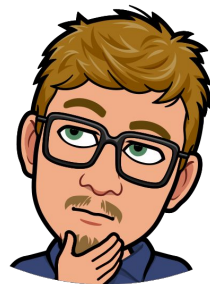
1. Do deductive reasoning algorithms still have a purpose in the age of transformers?

2. Where did reasoning algorithms go wrong?

What should they look like today?



# Conclusions for this talk:



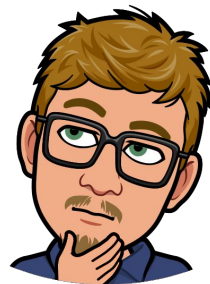
1. Do deductive reasoning algorithms still have a purpose in the age of transformers?

***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 transformers?

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

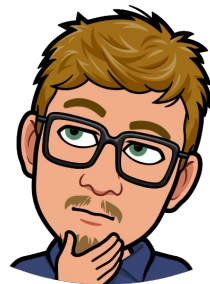
2. Where did reasoning algorithms go wrong?

***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 transformers?

***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>