UCLA Computer Science

STAR AI RESEARCH LAB UCLA

Agentic AI is Neurosymbolic AI

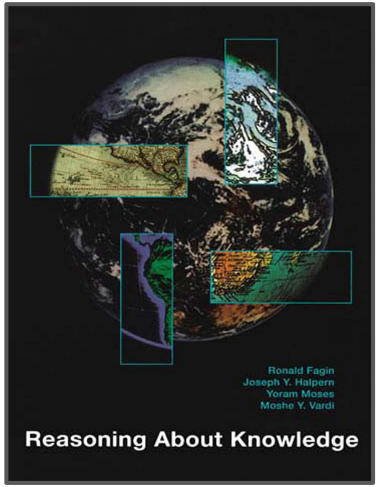
Guy Van den Broeck

10th Annual Center for Human-Compatible AI Workshop (CHAI) - Jun 5 2026

Reasoning with Symbolic AI

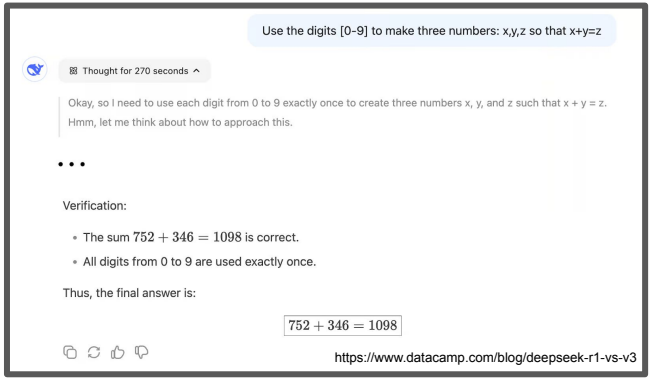
- Logic and probabilistic
- **Deductive** reasoning algorithms
- Correct on *all* problems
- Limited scope
- Intractable

30 years



Reasoning with Transformers

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



Questions for this talk:

1. Do deductive reasoning algorithms still have a purpose in the age of agentic AI?
2. Can we put “LLM probability” into the symbolic reasoning?
3. Can we put “symbolic reasoning” into the LLM?

What did I do this semester?

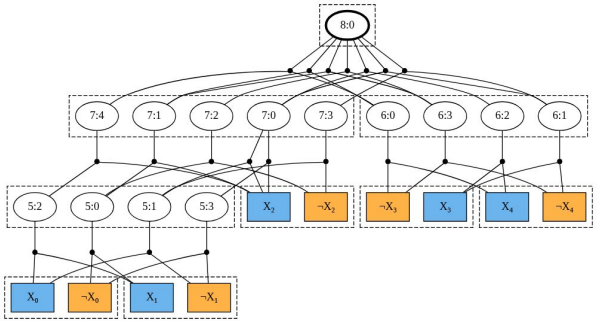
Tuesday 3/10: Submitted 30 pages of pure theory to SAT conference, defining a data structure for logical reasoning with nicer properties – a model counter

Wednesday 3/11: Gave the latex source to Claude Code, got a working implementation by EOD. Passes all tests, gives correct model counts,

Friday 3/13: Became faster than SoTA decision diagram compilers.

$$$$$$$$$$$

Today: Fastest model counter in the world



Florent Capelli, YooJung Choi, Stefan Mengel, Martín Muñoz and Guy Van den Broeck. A canonical generalization of OBDD, In SAT 2026.

Why did this happen?

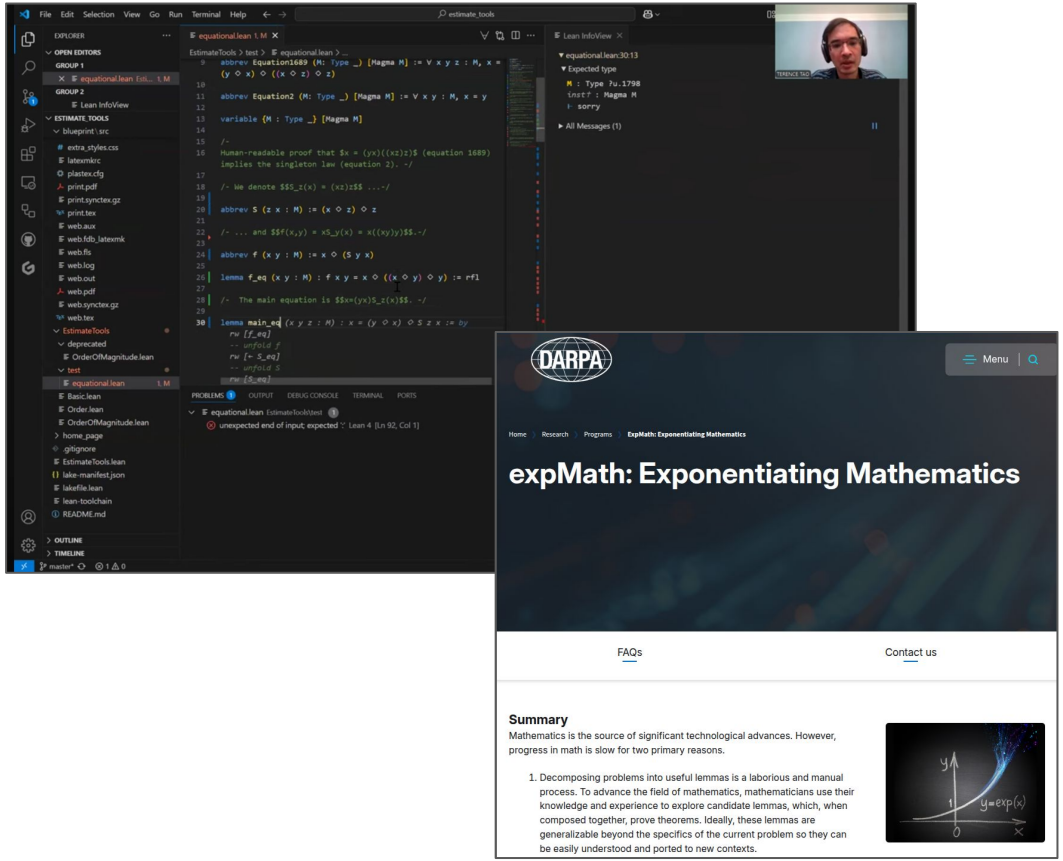
- Not because of **me**: *I don't know Rust and have written 0 LOC*
- Not because of **the LLM**: *it makes too many mistakes to write this code*
- Not because of **me and the LLM** working together:
  I don't have enough time to understand diffs for 458,609 LOC

Why?

- Called 100,000 ***symbolic*** tools (rust compiler, profiler, sat solvers, …)
- Strong ***symbolic*** guardrails: 1000 unit tests and ground-truth answers
- Strong ***symbolic*** objective: runtime on benchmarks

It happened because of neurosymbolic AI!

Math and CS Theory have become **Neurosymbolic**



How AI is changing the nature of mathematical research

What machine learning theorists learned using AI agents to generate proofs — and what comes next.

By Michael Kearns, Aaron Roth

March 9, 2026 | 10 min read

Share

Modern AI coding tools have revolutionized software engineering, with developers now using AI assistants to write a substantial fraction of their code across a range of applications. As scientists studying the theory of machine learning, we're already seeing a similar transformation in basic scientific methodology, especially for research of a mathematical nature.

More precisely, AI tools are now able to develop and write rigorous mathematical proofs only from prompts providing high-level proof sketches. These proofs are written in longstanding "languages" for detailing mathematical arguments, in the same way that code is written in formal programming languages like Python. AI seems to have become proficient in both kinds of languages and their underlying logics.

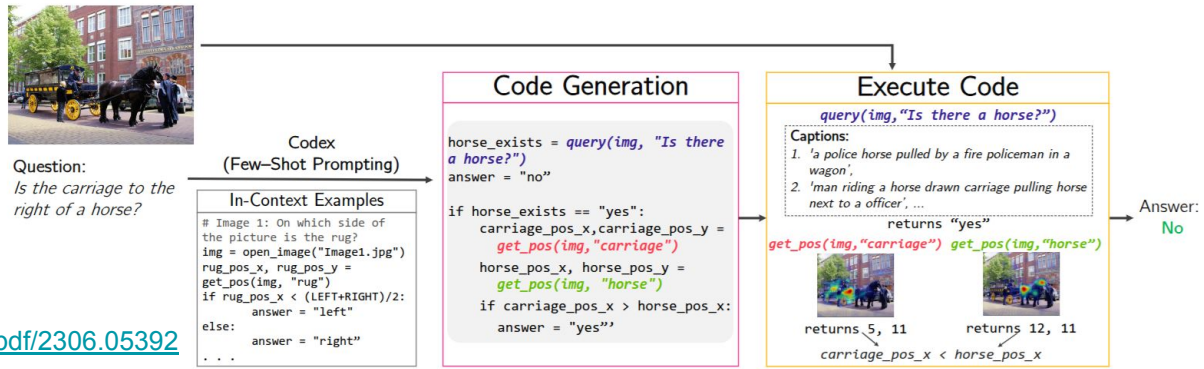
We came to this realization during a three-week period last summer, when we used agentic AI tools to write a mathematical paper that normally would have taken months. The 50-page paper describes and solves an optimization problem based on concepts from graph theory and machine learning. A typical prompt we would give the AI to set up the general framework for our paper looked like this: "Imagine a directed acyclic network of linear least-squares learning agents, each of which shares a common

"Working with proof-based AI tools is akin to collaborating with a smart, broadly educated but occasionally error-prone colleague."

The world has become Neurosymbolic

Victory!

- Software engineering has become Neurosymbolic
- Math and CS Theory have become Neurosymbolic
- Robotics has become Neurosymbolic
- Algorithms, Architecture has become Neurosymbolic (AlphaEvolve)
- AI4Science has become Neurosymbolic
- Computer Vision has become Neurosymbolic



https://arxiv.org/pdf/2306.05392

Why did neurosymbolic AI “win”? *No thanks to us.*

Success stories **despite** not understanding the symbolic side well.

- Ask a Computer Vision researcher about the VQA code they generate. Answer: I don’t know, it’s just Python.
- AlphaEvolve generated *hundreds of millions of programs*. No attempt to be smart about it.
- A $200-per-month Claude Code subscription uses $5,000 in compute, subsidized by Anthropic [forbes]

Scale and brute force compensate for this lack of understanding.

Why did neurosymbolic AI “win”?

The philosophy was always right.

Because of the overwhelming gravitational pull of

Compositionality

Abstraction

Soundness

Neurosymbolic data, training, and inference is vastly more powerful because of these properties!

What is the role of the neurosymbolic AI researcher?

Will my community win a Turing Award for neurosymbolic AI?

- Warning: ***offensive, unfair, facetious***

  The world thinks we mostly care about proving that doing the right thing is impossible, repeating Kahneman's System 1 and System 2, fuzzy logic, MNIST, and fighting culture wars from the Imagenet era of AI (†2022)

- We don't even know that we've "won", so how do we expect others to know?

- We can do better than prompting until it works (prompt and pray)
  - World needs tight integration of deductive reasoning and LLM
  - Neurosymbolic AI knows how to bridge using probabilistic logic!

Questions for this talk:

1. Do deductive reasoning algorithms still have a purpose in the age of agentic AI?
2. **Can we put “LLM probability” into the symbolic reasoning?**
3. Can we put “symbolic reasoning” into the LLM?

Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

🤔 **Let's ask an LLM.**

```
def compute_answer():
    total_students = 200
    smallest_group = (total_students - 10) // 3
    return smallest_group
```

Not exactly correct! Let's ask again.

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

🤔 **Let's ask an LLM a** second **time.**

```
def compute_answer():
    total_students = 200
    smallest_group = (total_students + 10) // 3
    return smallest_group
```

Not exactly correct! Let's ask again.

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

🤔 **Let's ask an LLM a** third **time.**

```
def compute_answer():
    total_students = 200
    smallest_group = (total_students + 10) // 3
    return smallest_group
```

Same program
Ask again.

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

🤔 **Let's ask an LLM a** fourth **time.**

```
def compute_answer():
    total_students = 200
    smallest_group = (total_students // 3) - 10
    return smallest_group
```

Not exactly correct! Let's ask again.

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

🤔 **Let's ask an LLM a** fifth **time.**

```
def compute_answer():
    total_students = 200
    difference = 10
    group1 = group2 = (total_students + difference) // 3
    group3 = group1 - difference
    return group3
```

Phew! finally

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

Why do we need so many samples from the language model?

Deterministic Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

Why do we need so many samples from the language model?

LLM is a distribution over token sequences.

We sample a deterministic program and throw away the uncertainty!!!

Probabilistic Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

🤔 **Let's ask an LLM but let's not throw away the uncertainty**

```
def compute_answer():
    total_students = 200
    smallest_group = (total_students + 10) // 3
    return smallest_group
```

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Probabilistic Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

🤔 **Let's ask an LLM but let's not throw away the uncertainty**

```
def compute_answer():
    total_students = X12 X13 X14
    smallest_group = (total_students + X25 X26) // X27
    return smallest_group
```

- Interpret digits as random variables.

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Probabilistic Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

🤔 **Let's ask an LLM but let's not throw away the uncertainty**

```
def compute_answer():
    total_students = X12 X13 X14
    smallest_group = (total_students Z24 X25 X26 ) Z27 X28
    return smallest_group
```

- Interpret digits as random variables.
- Interpret operators as random variables

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Probabilistic Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

```
def compute_answer():
    total_students = X12 X13 X14
    smallest_group = (total_students Z24 X25 X26 ) Z27 X28
    return smallest_group
```

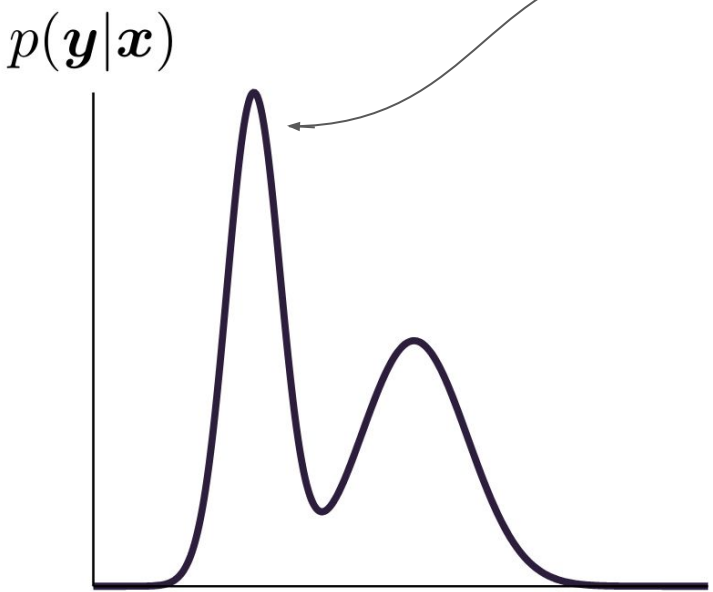
This is a probabilistic program!!

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

This probabilistic program approximates LLM distribution around a sample

```
def compute_answer():
    total_students = x12 x13 x14
    smallest_group = (total_students z24 x25 x26 ) z27 x28
    return smallest_group
```

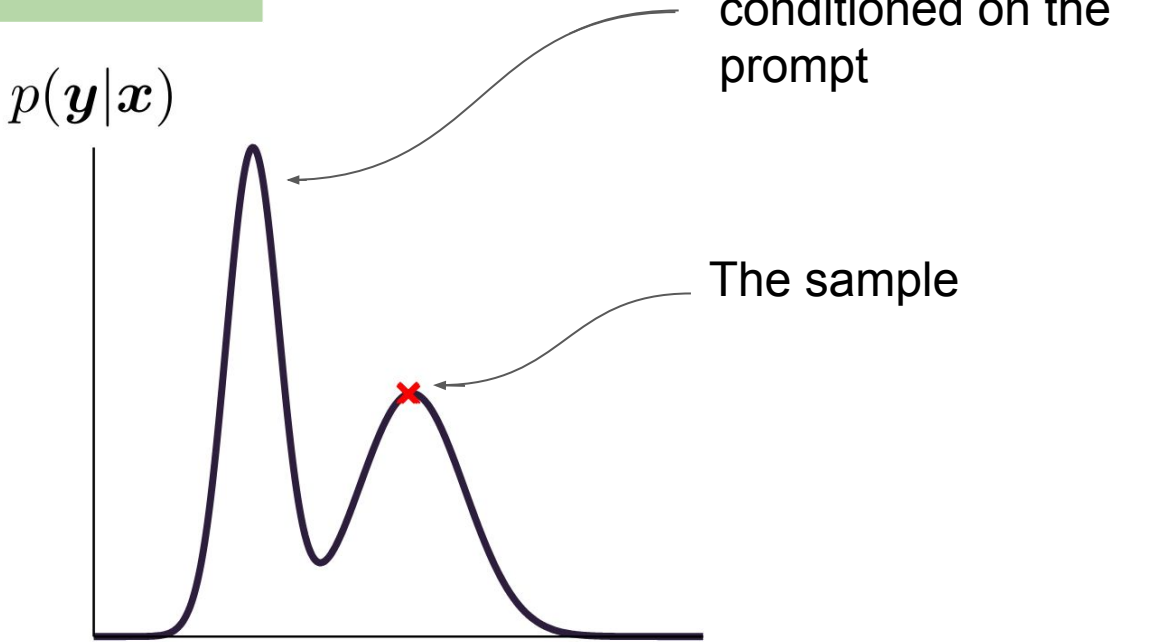
The LLM distribution conditioned on the prompt



Kareem Ahmed, Kai-Wei Chang, Guy Van den Broeck, A Pseudo-Semantic Loss for Autoregressive Models with Logical Constraints, 2023

This probabilistic program approximates LLM distribution around a sample

```
def compute_answer():
    total_students = X12 X13 X14
    smallest_group = (total_students Z24 X25 X26 ) Z27 X28
    return smallest_group
```

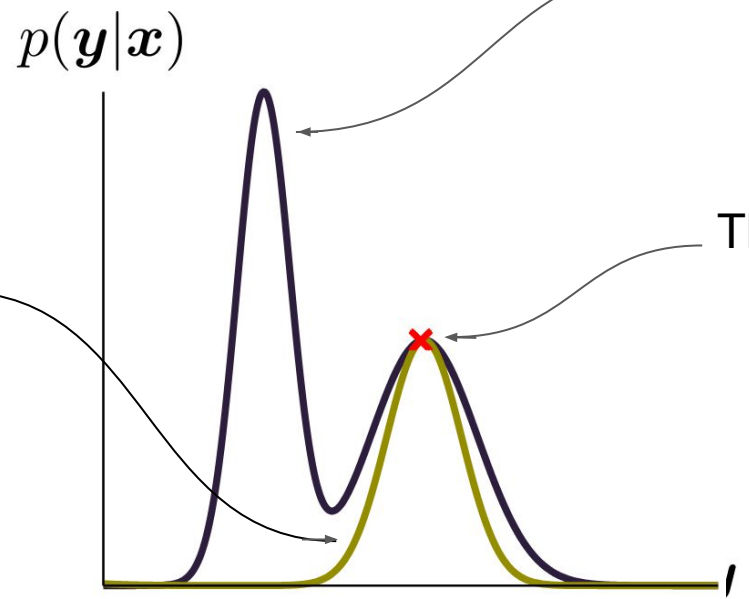


Kareem Ahmed, Kai-Wei Chang, Guy Van den Broeck, A Pseudo-Semantic Loss for Autoregressive Models with Logical Constraints, 2023

This probabilistic program approximates LLM distribution around a sample

```
def compute_answer():
    total_students = x12 x13 x14
    smallest_group = (total_students z24 x25 x26 ) z27 x28
    return smallest_group
```

The LLM distribution conditioned on the ompt



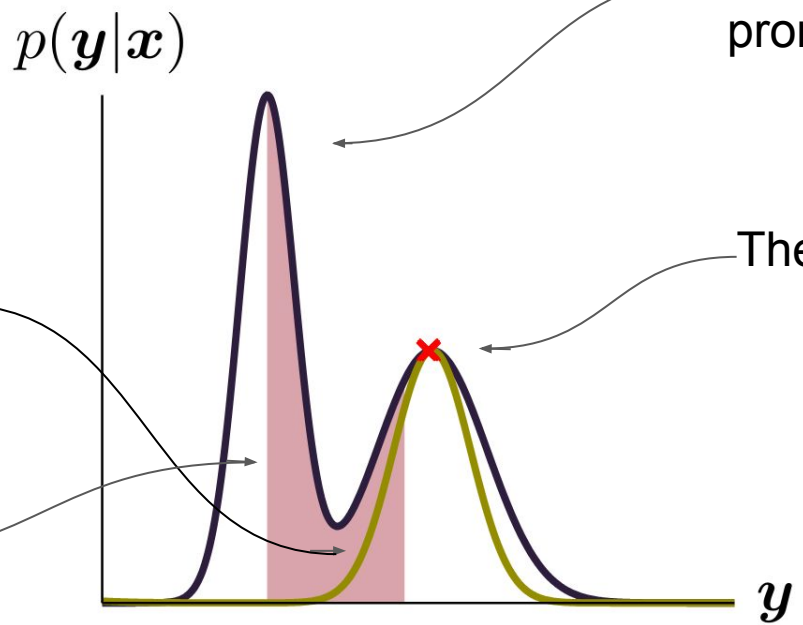
The sample

The distribution of the probabilistic program

Kareem Ahmed, Kai-Wei Chang, Guy Van den Broeck, A Pseudo-Semantic Loss for Autoregressive Models with Logical Constraints, 2023

This probabilistic program approximates LLM distribution around a sample

```
def compute_answer():
    total_students = x12 x13 x14
    smallest_group = (total_students z24 x25 x26 ) z27 x28
    return smallest_group
```



The LLM distribution conditioned on the prompt

The sample

The distribution of the probabilistic program

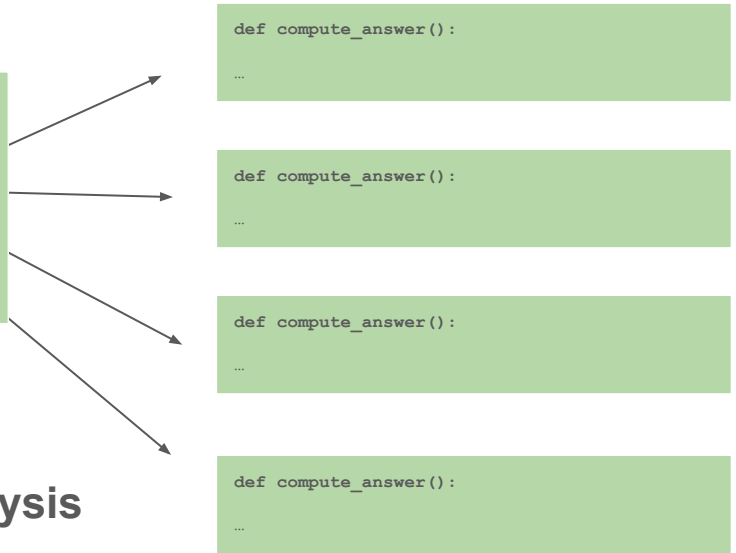
The space of the correct programs

Kareem Ahmed, Kai-Wei Chang, Guy Van den Broeck, A Pseudo-Semantic Loss for Autoregressive Models with Logical Constraints, 2023

Probabilistic Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

```
def compute_answer():
    total_students = x12 x13 x14
    smallest_group = (total_students z24 x25 x26 ) z27 x28
    return smallest_group
```



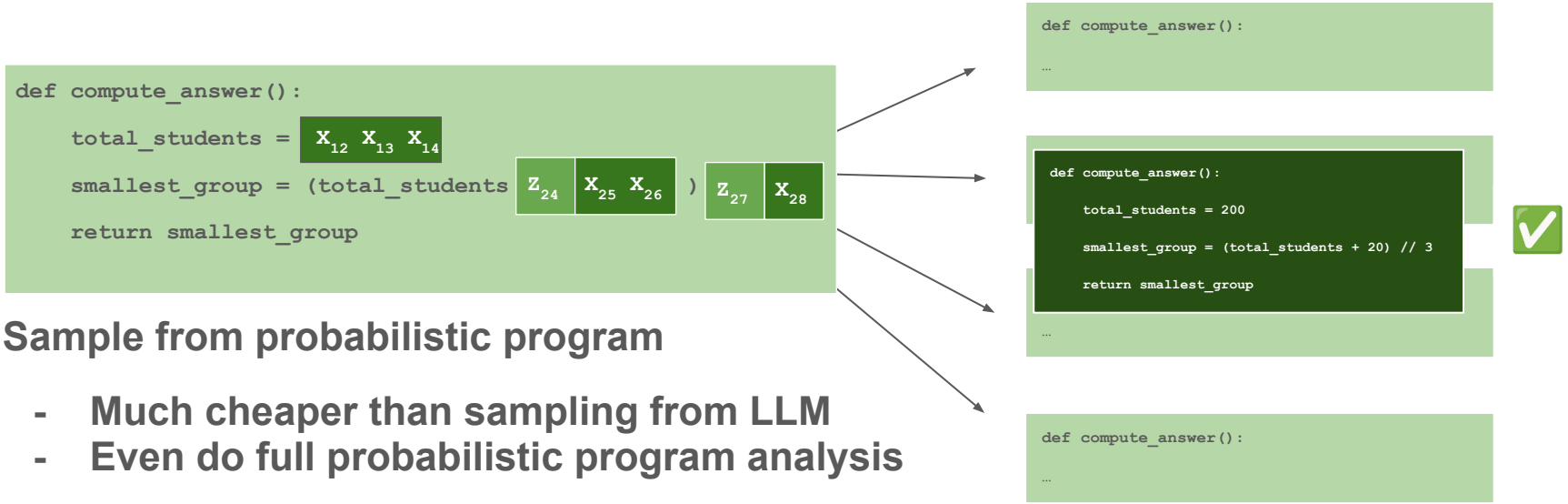
Sample from probabilistic program

- **Much cheaper than sampling from LLM**
- **Even do full probabilistic program analysis**

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Probabilistic Programs-of-thought for grade school math

Question: A class of 200 students is split into 3 groups such that 2 of them are equal in number and the last one (which is the smallest) is 10 less than each of the other groups. How many students are in this (smallest) group?

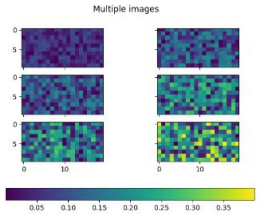


Sample from probabilistic program

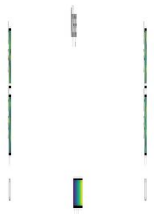
- **Much cheaper than sampling from LLM**
- **Even do full probabilistic program analysis**

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Plot2Code

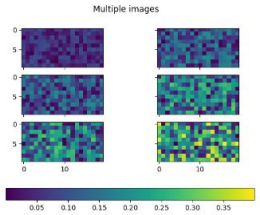


ground-truth

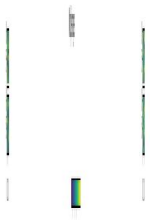


LLM sample

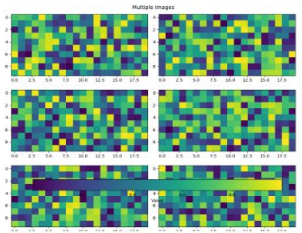
Plot2Code



ground-truth

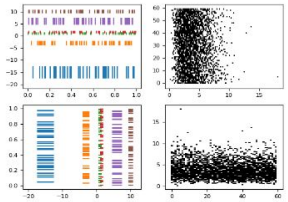


LLM sample



PPoT sample

Plot2Code



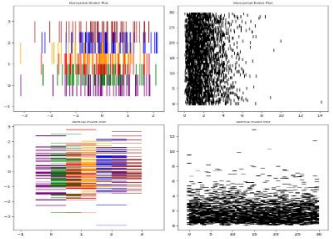
ground-truth

```
--- LLM sample
+++ PPoT sample
- data1 = np.random.randn(5, 50)
+ data1 = np.random.randn(6, 50)
- data3 = np.random.gamma(1, 1, (60, 50))
+ data3 = np.random.gamma(2, 1, (60, 50))
+ fig, axs = plt.subplots(2, 2,
      figsize=(80, 80))
- fig, axs = plt.subplots(2, 2,
      figsize=(10, 10))
```

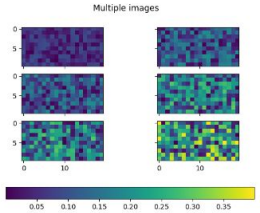
diff

ERROR

LLM sample



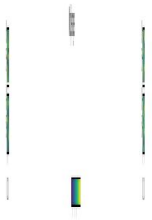
PPoT sample



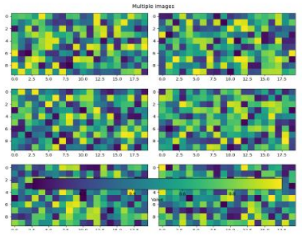
ground-truth

```
--- LLM sample
+++ PPoT sample
  # Create a figure with a grid of
      subplots
- fig, axs = plt.subplots(2, 2,
      figsize=(80, 1))
+ fig, axs = plt.subplots(3, 2,
      figsize=(10, 8))
  for ax, d in zip(axs.flat, data):
      im = ax.imshow(d,
           vmin=np.min(data),
           vmax=np.max(data))
```

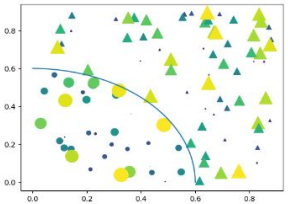
diff



LLM sample



PPoT sample



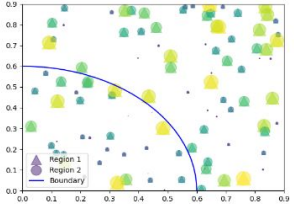
ground-truth

```
--- LLM sample
+++ PPoT sample
- x = np.random.rand(100) * 0.9
+ x = np.random.rand(200) * 0.9
- plt.scatter(x, y, s=marker_area,
      c=marker_color, marker='^',
      alpha=0.5, label='Region 1')
+ plt.scatter(x, y, s=marker_area,
      c=marker_color, marker='^',
      alpha=0.6, label='Region 1')
```

diff

ERROR

LLM sample



PPoT sample

Probabilistic Programs-of-thought

Best-of-n performance: 1.5% - 7% boost
for the same amount of GPU compute

math reasoning — program inversion — plot generation

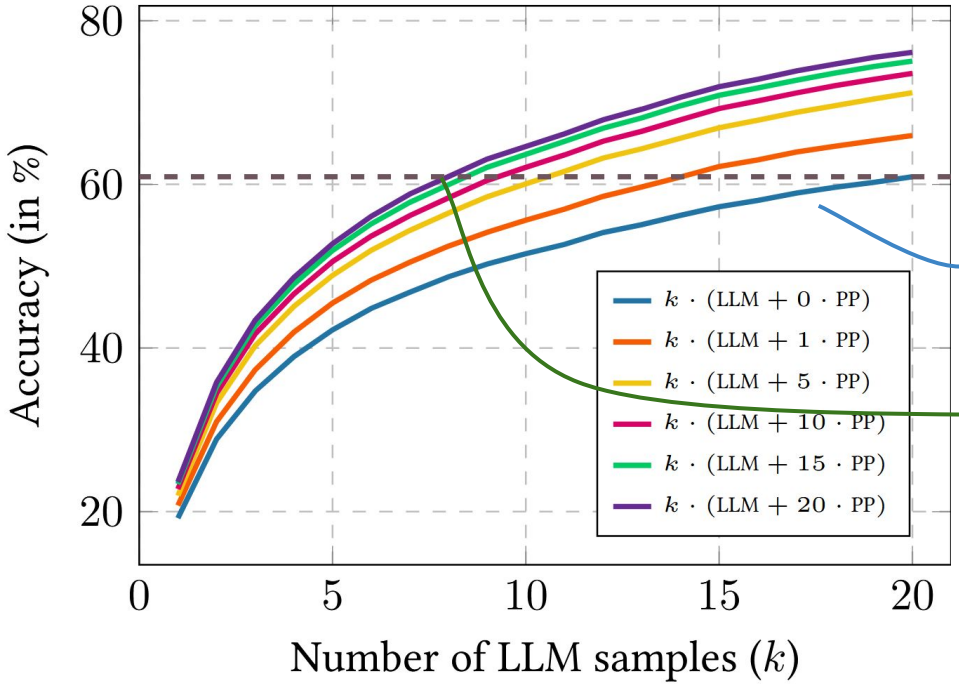
PPoT samples

LLM samples

$k \cdot (\text{LLM} + m \cdot \text{PP})$	GSM8k Qwen2.5 Coder 0.5B	GSM8k Qwen2.5 Coder 3B	GSM8k Qwen2.5 Coder 7B	CRUXEval Qwen2.5 Coder 0.5B	CRUXEval Qwen2.5 Coder 3B	CRUXEval Qwen2.5 Coder 7B	Plot2Code Qwen2.5 VL 3B	Plot2Code Qwen2.5 VL 7B
$1 \cdot (\text{LLM} + 0 \cdot \text{PP})$	24.94	72.63	82.71	30.87	40.5	57.12	39.44	36.70
$1 \cdot (\text{LLM} + 5 \cdot \text{PP})$	**28.51**	**76.19**	**84.76**	**37.37**	**45.87**	**60.62**	**41.89**	**42.48**
$5 \cdot (\text{LLM} + 0 \cdot \text{PP})$	42.53	86.20	91.20	46.13	59.13	73.37	63.86	69.92
$5 \cdot (\text{LLM} + 5 \cdot \text{PP})$	**49.13**	**89.08**	**93.02**	**53.13**	**63.87**	**76.87**	**64.84**	**71.95**
Metric	Accuracy						Text Match	

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Probabilistic Programs-of-thought



How much can we gain from PPoT?

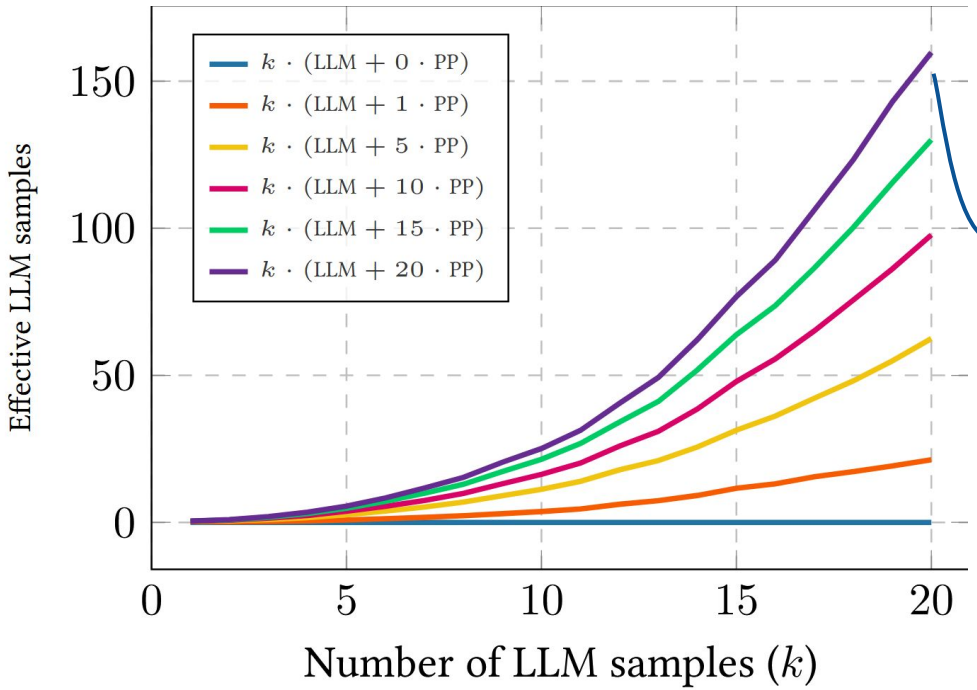
This curve is your usual LLM best-of-n setting

Same accuracy for less than half the number of LLM calls!

Qwen 2.5 Coder on GSM8k

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

Probabilistic Programs-of-thought: scaling laws



How much can we gain from PPoT?

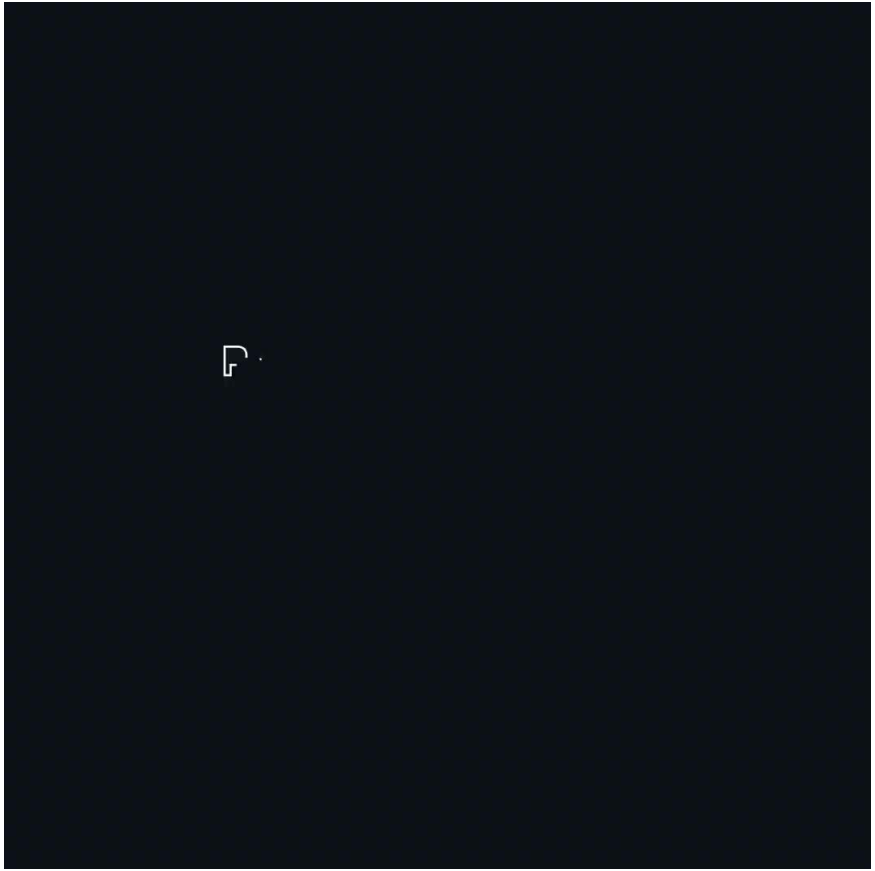
You need >150 LLM samples to achieve the same accuracy as PPoT!

Qwen 2.5 Coder on GSM8k

Poorva Garg, Renato Geh, Daniel Israel, Todd Millstein, Kyle Richardson, Guy Van den Broeck. Probabilistic Programs of Thought, 2026.

PPoT Conclusions

1. LLM comes with uncertainty
2. Pushing it forward
   through the symbolic program
   lets us reason about it
3. We can do better than
   prompting + LLM sampling
4. Programs of thought
   were always meant to be
   probabilistic programs of thought

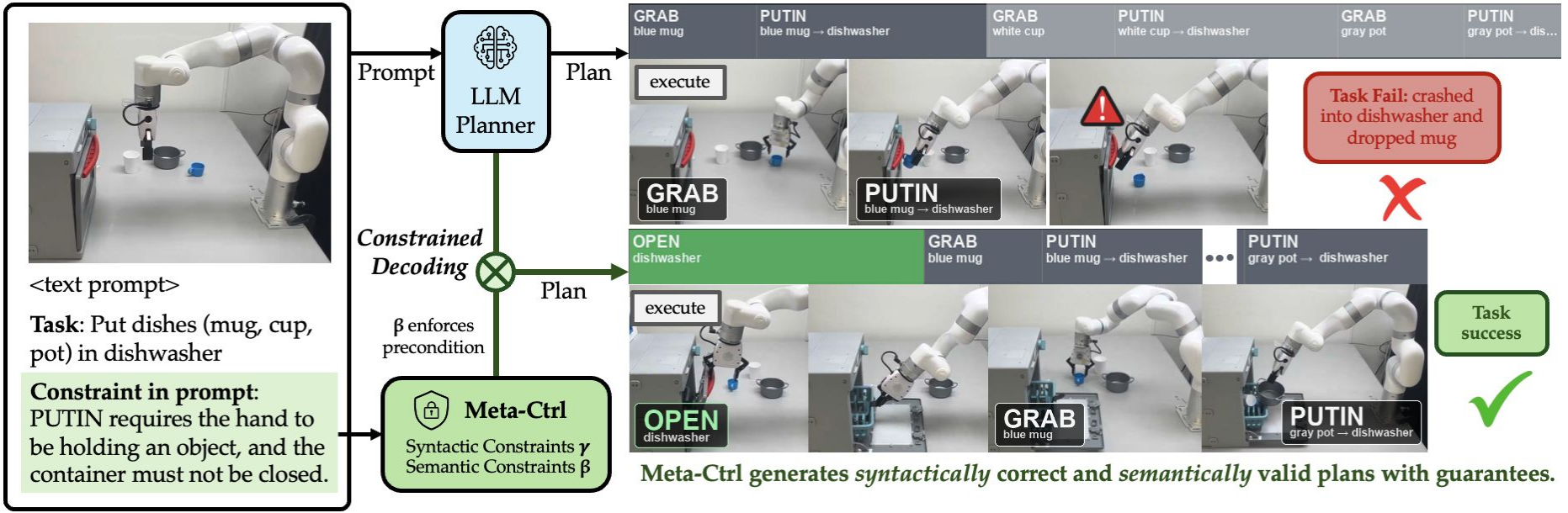


Questions for this talk:

1. Do deductive reasoning algorithms still have a purpose in the age of agentic AI?
2. Can we put “LLM probability” into the symbolic reasoning?
3. **Can we put “symbolic reasoning” into the LLM?**

Probabilistic constrained decoding for robot planning

Probabilistic constrained decoding for robot planning



Reasoning about all Future Tokens: *Constraints*

p(next-token | *α*, prefix)

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

Lexical Constraint α: sentence contains keyword “winter”

Reasoning about all Future Tokens: *Constraints*

p(next-token | α, prefix)

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

Lexical Constraint α: sentence contains keyword "winter"

$\propto$ p(next-token | prefix) $\cdot$ $p(\alpha$ | next-token, prefix)

Bayes' rule lets us reason backwards in time!

Reasoning about all Future Tokens: *Constraints*

p(next-token | α, prefix)

cold	0.025
warm	0.001

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

Lexical Constraint α: sentence contains keyword "winter"

$\propto$ p(next-token | prefix)

cold	0.05
warm	0.10

$\cdot$ $p(\alpha$ | next-token, prefix)

cold	0.50
warm	0.01

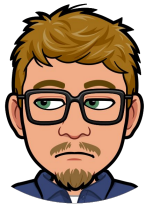
Reasoning about all Future Tokens

p_{lm}(next-token | α, prefix)

$\propto$ p_{lm}(next-token | prefix) · p_{lm}(α | next-token, prefix)

Using Bayes rule,

Intractable



Reasoning about all Future Tokens

p_{lm}(next-token | α, prefix)

Abusing Bayes rule,

$\propto$ p_{lm}(next-token | prefix) $\cdot$ $p_{circuit}$(α | next-token, prefix)

Use a tractable circuit model distilled from the transformer LLM…

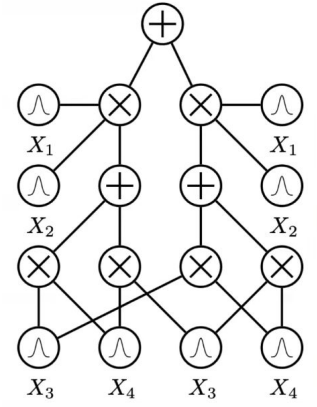
A `tractable digital twin'

Reasoning about all Future Tokens: Constraints

p_{lm}(next-token | α, prefix)

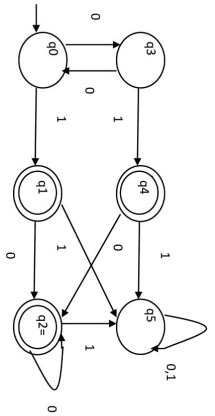
Abusing Bayes rule,

$\propto$ p_{lm}(next-token | prefix) · $p_{circuit}$(α | next-token, prefix)



weights/circuit p

X



models/dfa α

Honghua Zhang, Po-Nien Kung, Masahiro Yoshida, Guy Van den Broeck and Nanyun Peng. Adaptable Logical Control for Large Language Models, *In NeurIPS*, 2024.

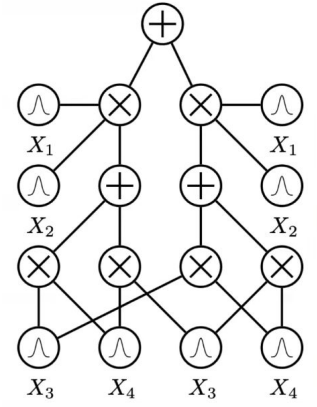
Reasoning about all Future Tokens: Constraints

p_{lm}(next-token | α, prefix)

Abusing Bayes rule,

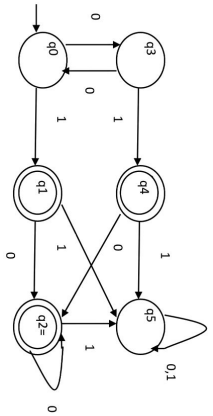
$\propto$ p_{lm}(next-token | prefix) · $p_{circuit}$(α | next-token, prefix)

$\sum_{\text{future}}$



weights/circuit p

x



models/dfa α

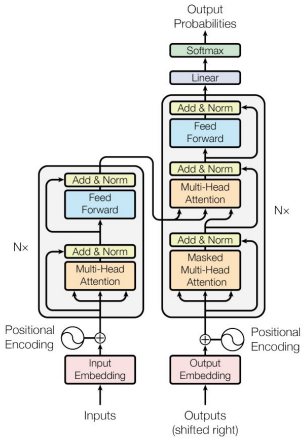
Honghua Zhang, Po-Nien Kung, Masahiro Yoshida, Guy Van den Broeck and Nanyun Peng. Adaptable Logical Control for Large Language Models, *In NeurIPS*, 2024.

Reasoning about all Future Tokens: Constraints

p_{lm}(next-token | α, prefix)

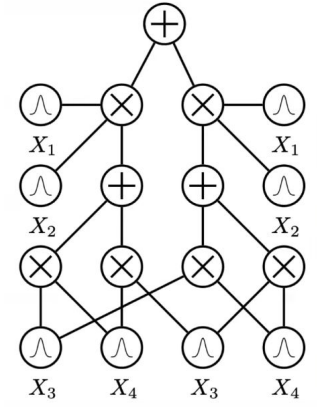
Abusing Bayes rule,

$\propto$ p_{lm}(next-token | prefix) · $p_{circuit}$(α | next-token, prefix)



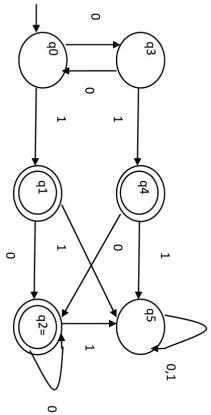
transformer p

· $\sum_{future}$



weights/circuit p

x



models/dfa α

Honghua Zhang, Po-Nien Kung, Masahiro Yoshida, Guy Van den Broeck and Nanyun Peng. Adaptable Logical Control for Large Language Models, *In NeurIPS*, 2024.

Condition LLM on constraints at two levels

① Syntax — token level

Base LM ✗

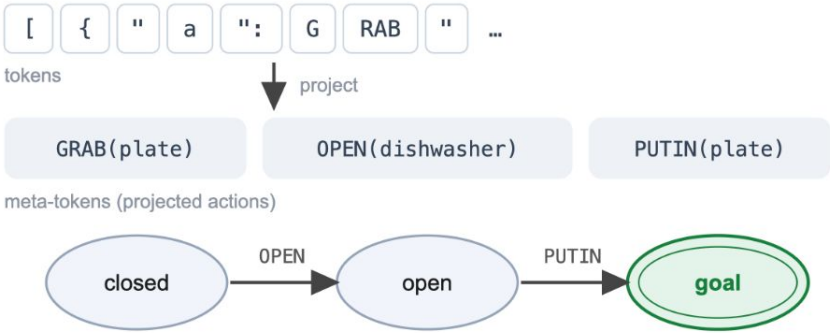
"To solve this problem… the action is TYPE…"

+ syntax (γ) ✓

```
{"WALK":["computer"], "GRAB":["mouse"]}
```

Well-formed — but still wrong at the action level.

② Semantics — action level



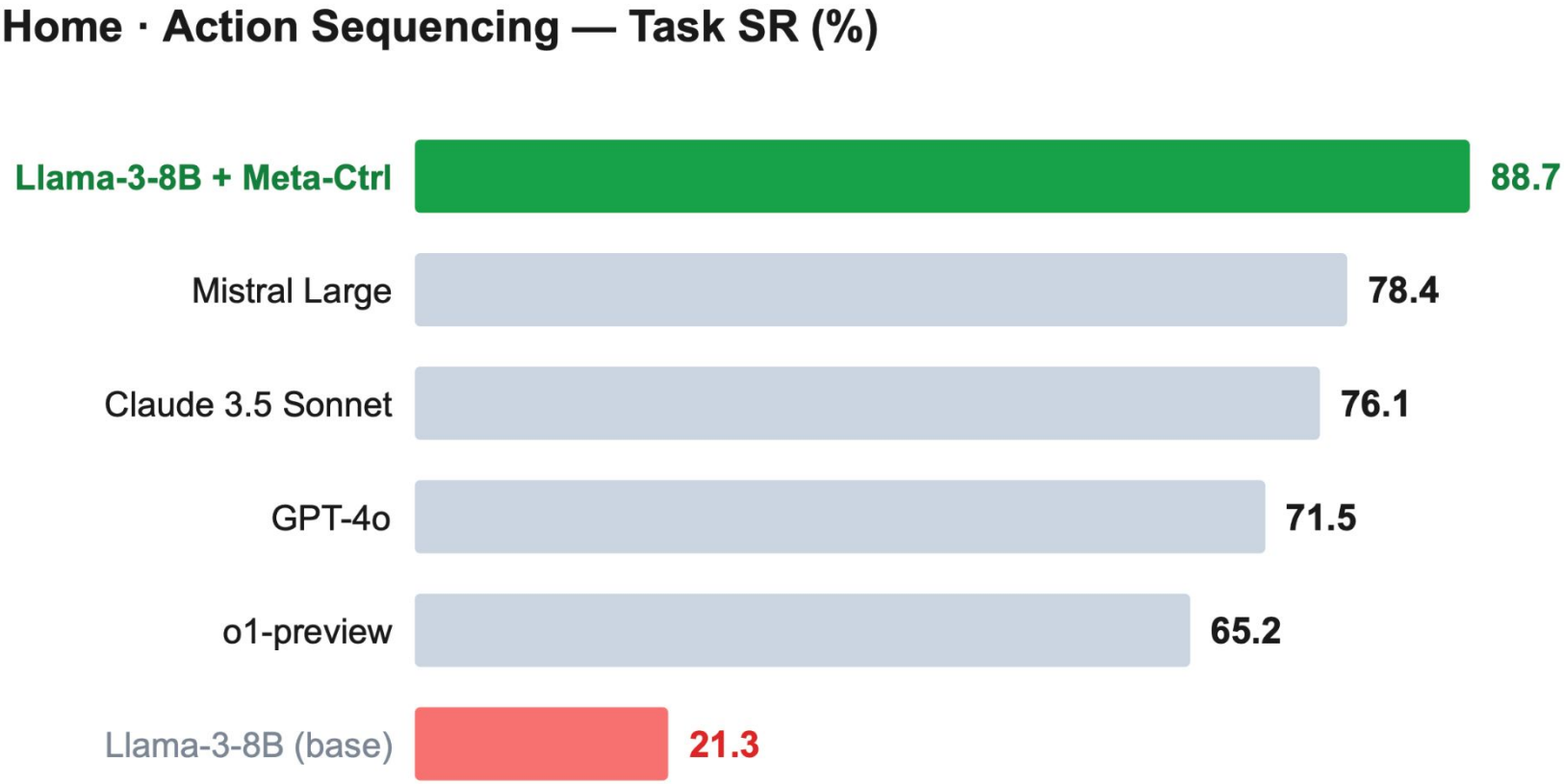
✗ Precondition violation

PUTIN into a closed dishwasher — must OPEN first.

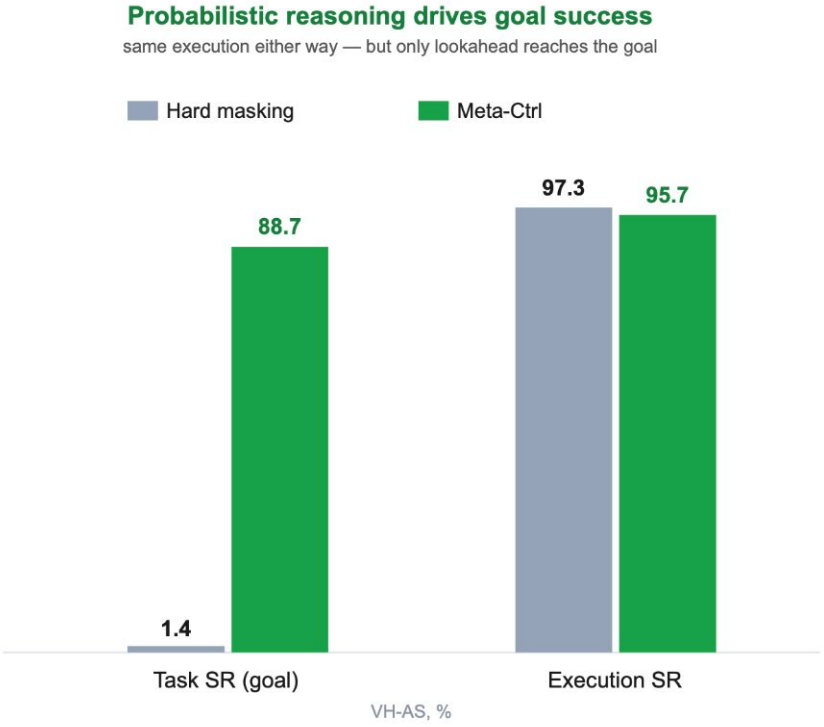
✗ Goal not met

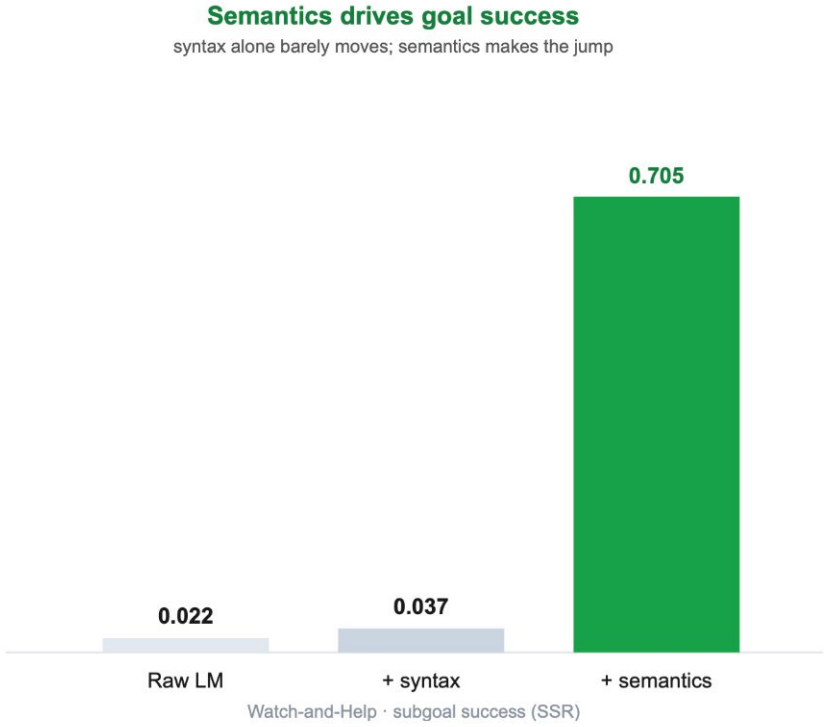
plan stops at OPEN — the goal needs the plate inside.

An 8B open model tops the leaderboard



What drives the gain





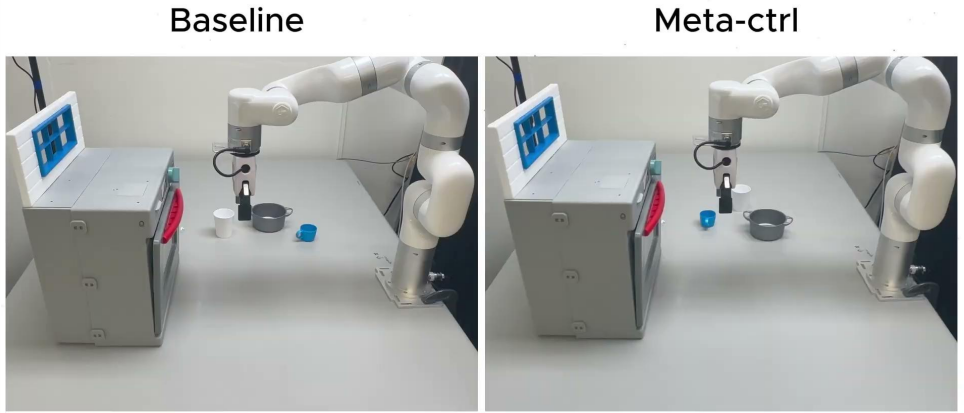
Meta-Ctrl Conclusions

1. Constraint α is guaranteed to be satisfied:
   if next-token makes α unsatisfiable, $p_{lm}(\text{next-token} \mid \alpha, \text{prefix}) = \mathbf{0}$.

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

2. Bayesian = goal-oriented
   (↔ masking)

3. Deployed on a real robot that now knows to *open the microwave before putting something in the microwave*



Why not use RL to solve all your problems?

Toxicity

Base LLM	0.385
PPO[7]	0.218
Quark[8]	0.196
DPO[9]	0.180

lower = less toxic

Perplexity

Base LLM	**25.57**
PPO[7]	14.27
Quark[8]	12.47
DPO[9]	21.59

lower = higher LLM probability

Base LLM	52.06
DPO	39.52

Lower **entropy** means lower diversity

DPO "thinks" that 99.9996% of all internet text is toxic.
Ad-Hoc Probabilistic Reasoning - broken

Questions for this talk:

1. Do deductive reasoning algorithms still have a purpose in the age of agentic AI?
2. Can we put “LLM probability” into the symbolic reasoning?
3. Can we put “symbolic reasoning” into the LLM?

Thanks

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

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