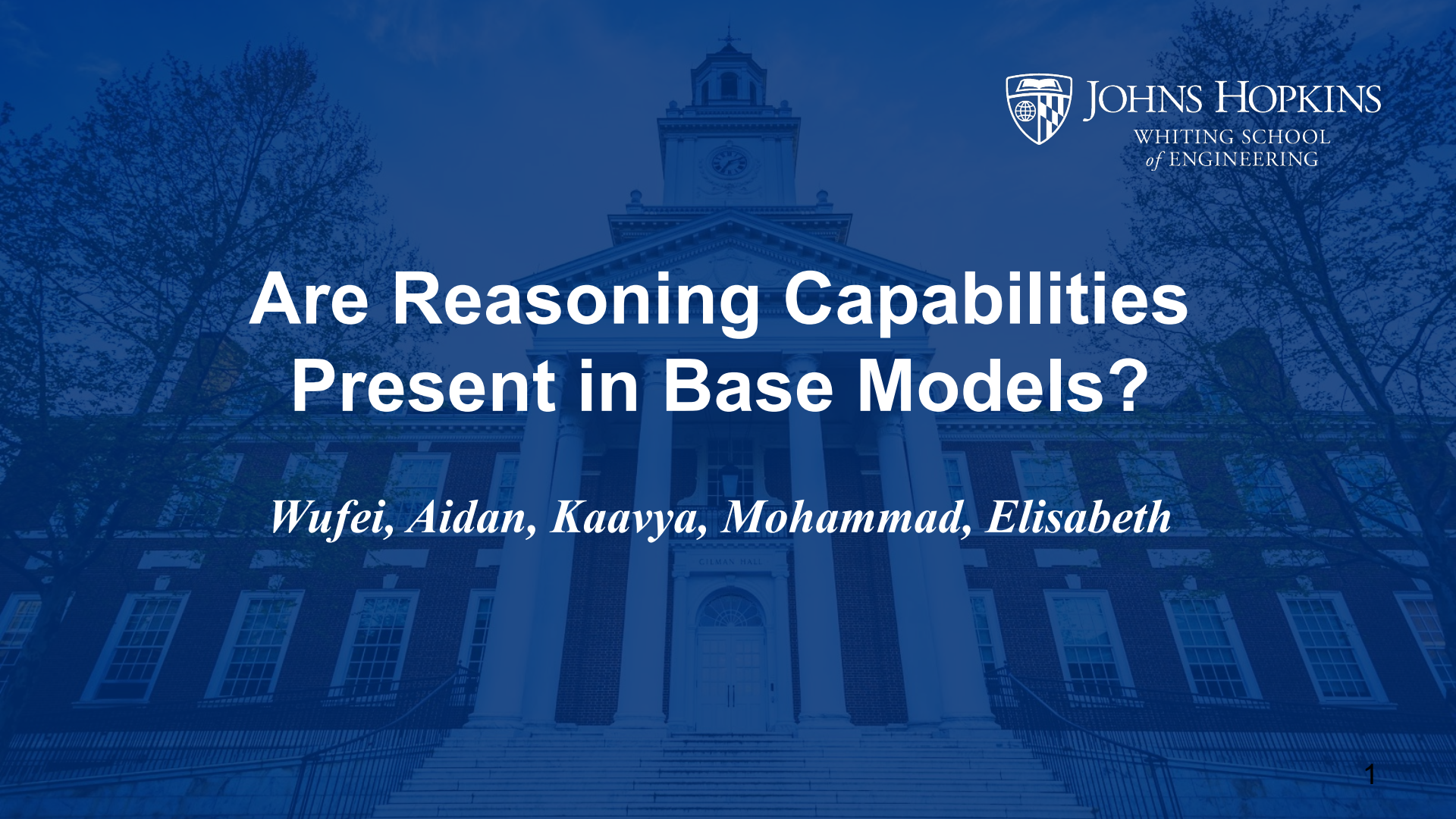
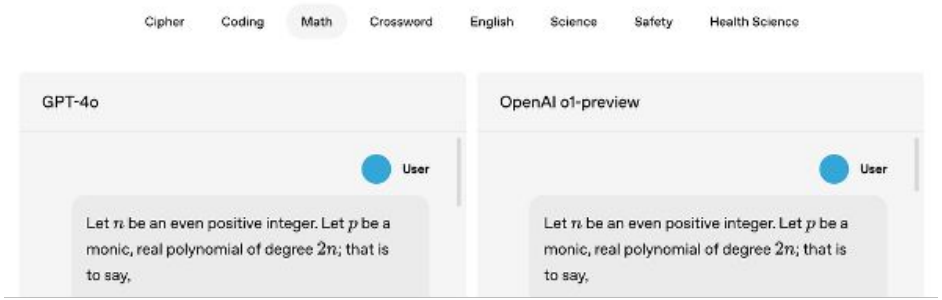
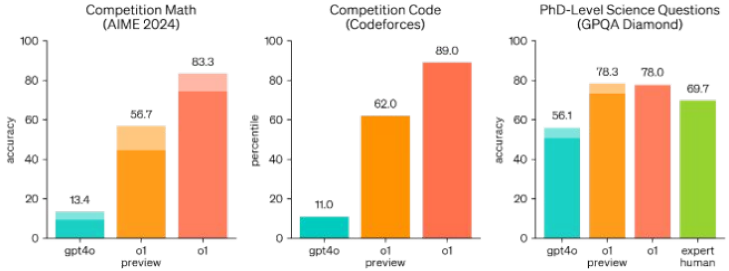

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING
Are Reasoning Capabilities Present in Base Models?
Wufei, Aidan, Kaavya, Mohammad, Elisabeth
GILMAN HALL

“Reasoning” Models

More complex reasoning problems

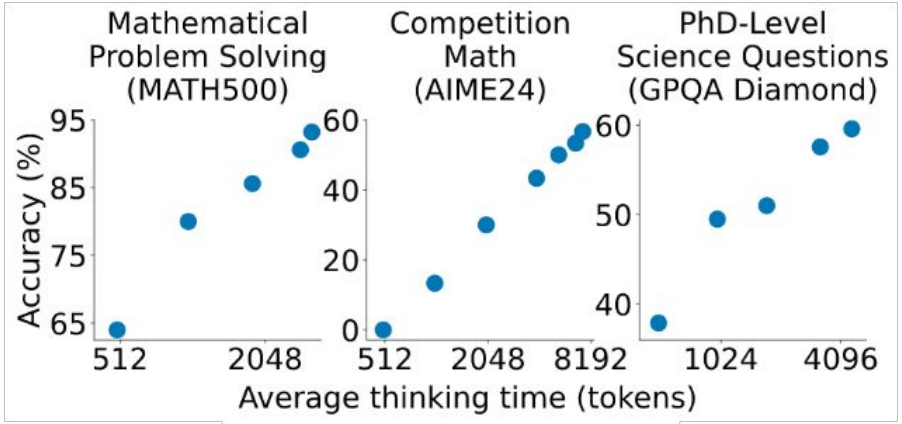


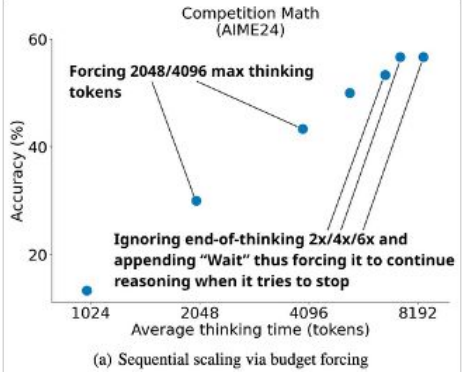
Power of reasoning models



Test-Time Scaling

- 1,000 reasoning samples for SFT training
- “Budget forcing” for test-time scaling.
  - “Wait,” and “Final answer:”

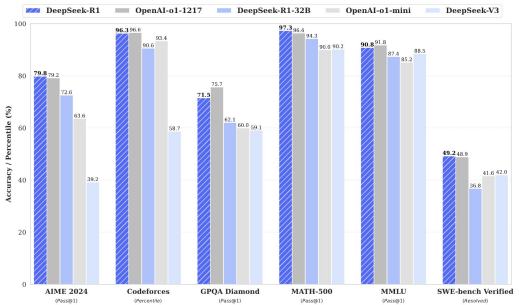


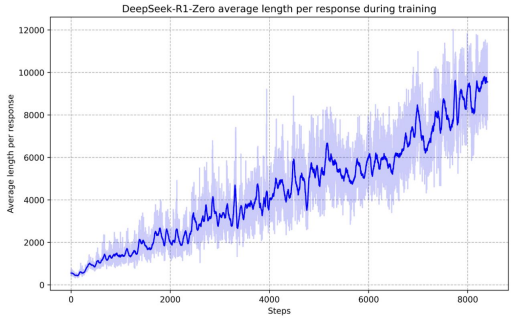


(a) Sequential scaling via budget forcing

GRPO

- An RL recipe that achieves o1-like performance
- GRPO-trained model more often exhibit:
  - Increasing response length
  - Self-reflection





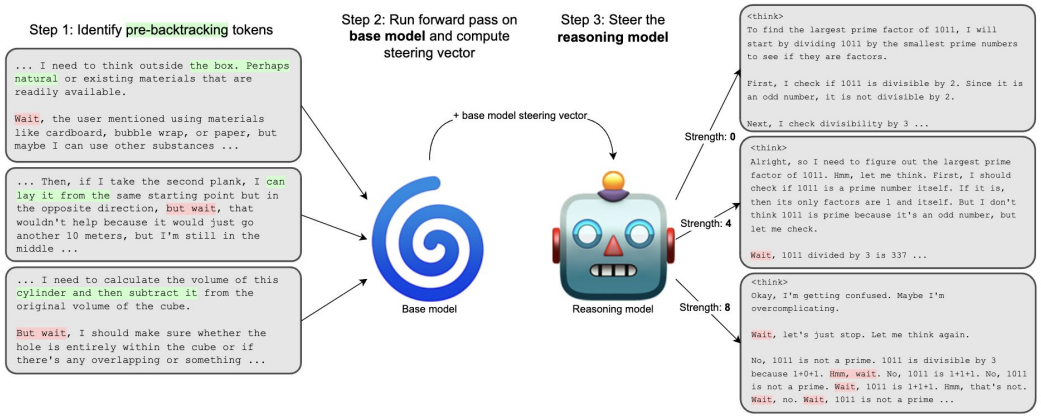
Rethinking RL

- What is the difference between base and thinking models that allows the latter to achieve superior performance?

- Are the capabilities emerged from RL fundamentally novel behaviors not in the base model?

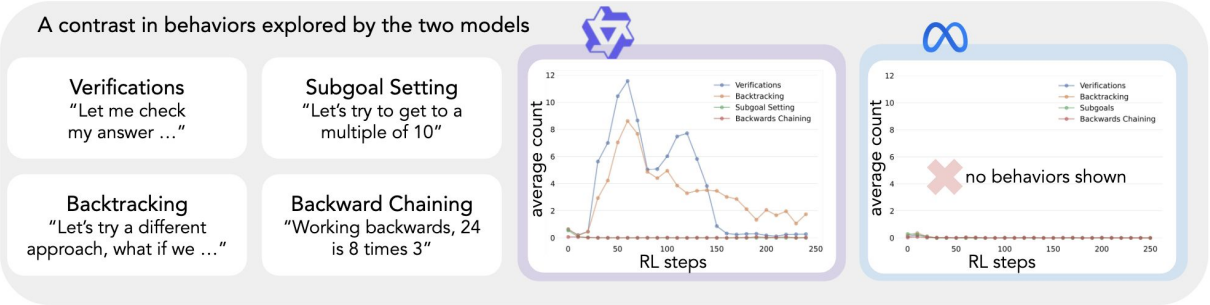
Hypotheses

1. Entirely new reasoning abilities acquired during RL.
2. Repurposing pre-existing representations for new mechanisms.
    a. A “steering vector” that unlocks backtracking.



Hypotheses

1. Entirely new reasoning abilities acquired during RL.
2. Repurposing pre-existing representations for new mechanisms.
3. Reasoning abilities in base model.
    a. Behaviors of verification, subgoal setting, backtracking, backward chaining…



Hypotheses

1. Entirely new reasoning abilities acquired during RL.
2. Repurposing pre-existing representations for new mechanisms.
3. Reasoning abilities in base model.
4. Amplifying effective reasoning patterns before RL.

Hypotheses

1. Entirely new reasoning abilities acquired during RL.
2. Repurposing pre-existing representations for new mechanisms.
3. Reasoning abilities in base model.
4. Amplifying effective reasoning patterns before RL.

If base models already have the knowledge necessary for reasoning, we can skip time-intensive RL/ SFT and use training-free methods to elicit desired behavior from base models.

Outline

Paper 1: Base Models Know How to Reason, Thinking Models Learn When

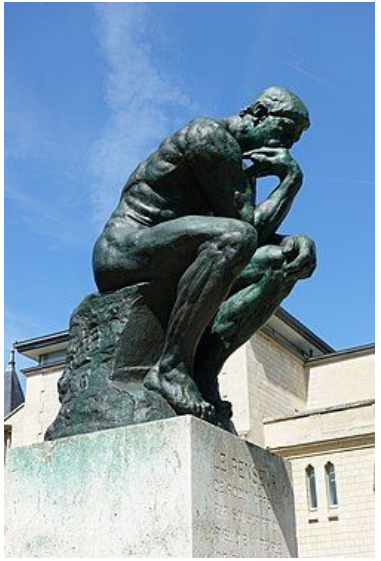
Paper 2: Reasoning with Sampling: Your Base Model is Smarter Than You Think

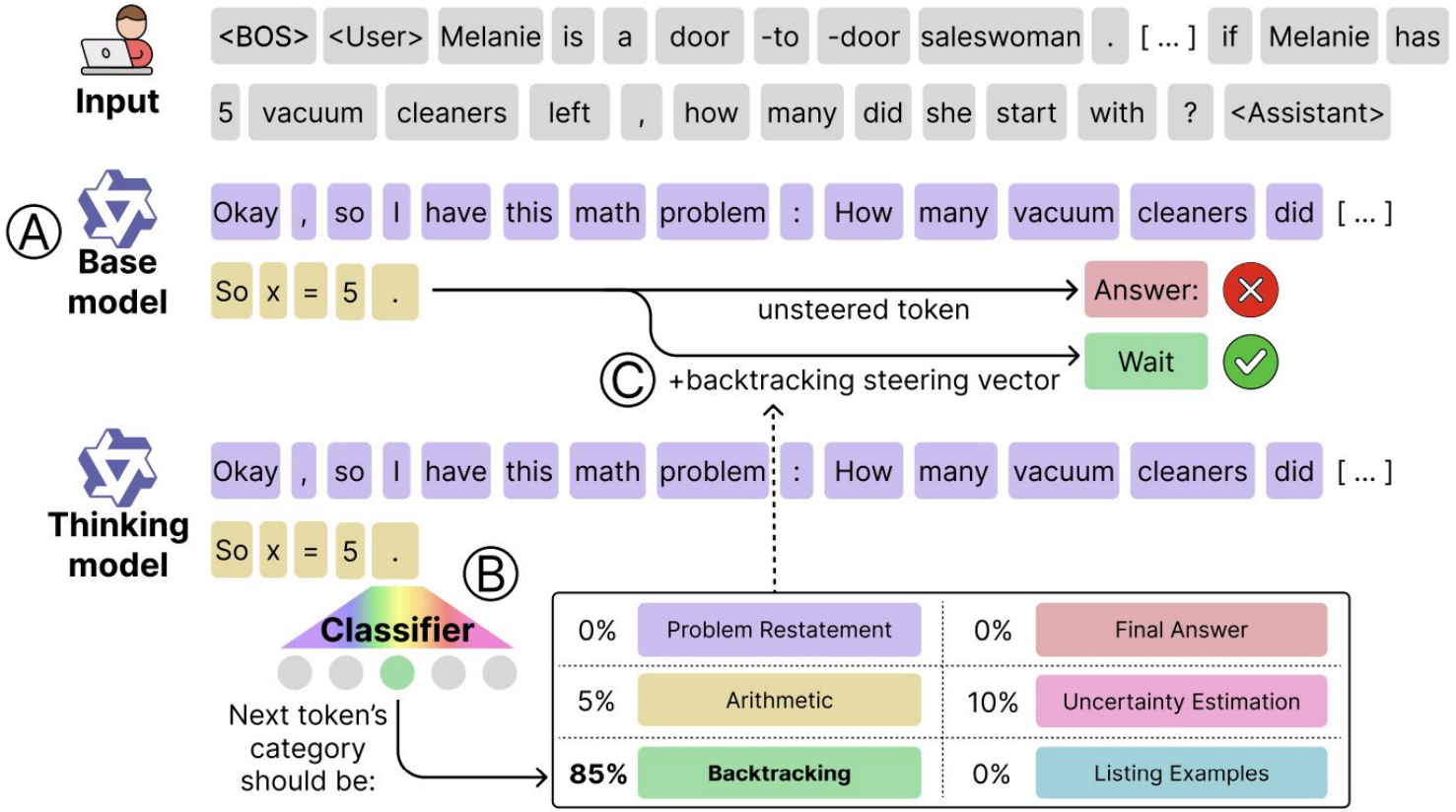
Base models already possess reasoning capabilities

Base Models Know How to Reason, Thinking Models Learn When

Premise / Summary

- What “types” of reasoning do (SFT) LLMs use?
  - Empirical view: 10-20 types (?)
- Are these types present in base models?
  - Knowing “when” to reason is in activations
- How do we prove the above?
  - Using SFT models to “steer” base models




Input
<BOS> <User> Melanie is a door -to -door saleswoman . [...] if Melanie has
5 vacuum cleaners left , how many did she start with ? <Assistant>
Ⓐ
Base model
Okay , so I have this math problem : How many vacuum cleaners did [...]
So x = 5 .
unsteered token
Answer:
Ⓒ
+backtracking steering vector
Wait
Thinking model
Okay , so I have this math problem : How many vacuum cleaners did [...]
So x = 5 .
Ⓑ
Classifier
Next token's category should be:
0% Problem Restatement
0% Final Answer
5% Arithmetic
10% Uncertainty Estimation
85% Backtracking
0% Listing Examples

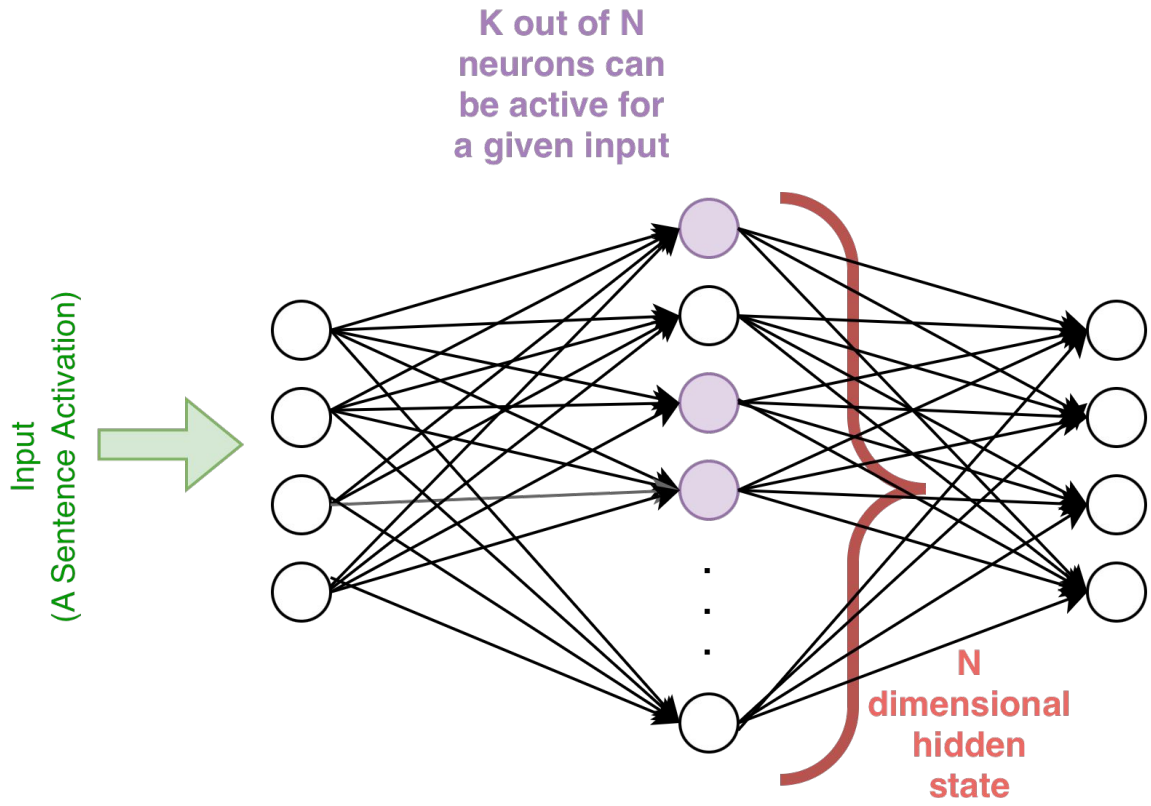
Taxonomy Curation

A taxonomy of reasoning methods must be:

- Interpretable
  - Each reasoning method should be understandable to humans
- Complete
  - Cover all of the possible reasoning methods
- Independent (aka disjoint)
  - No overlap between methods

Taxonomy Curation via Sparse Autoencoders

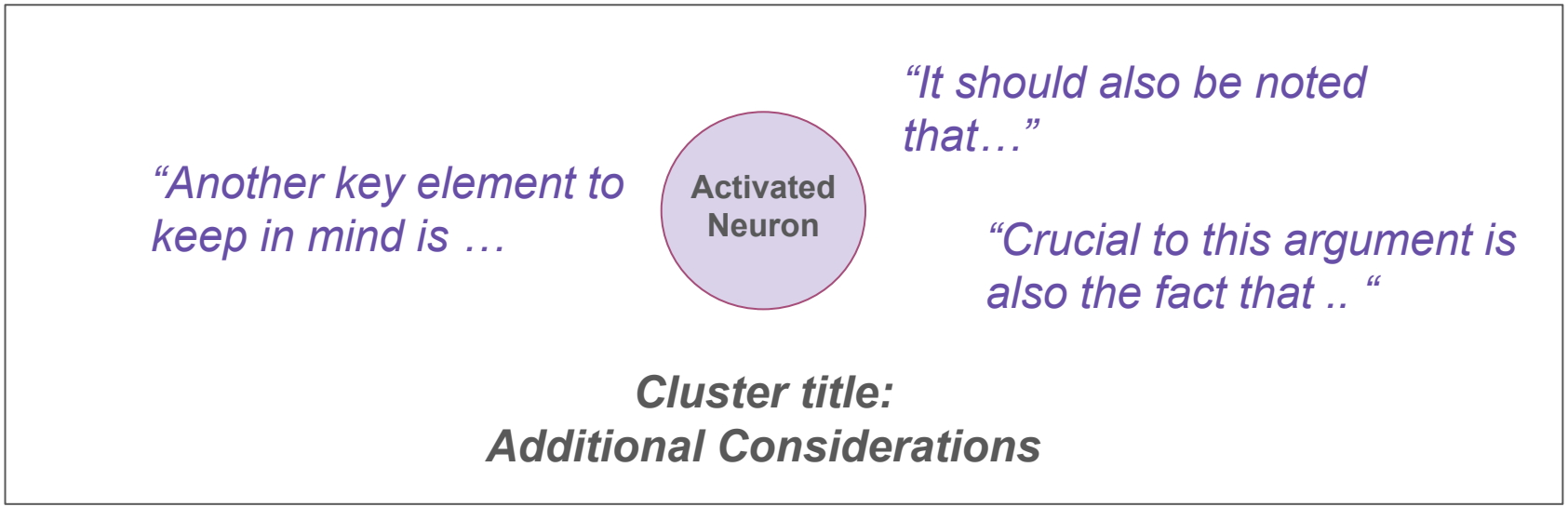
Autoencoders enable efficient latent representations



Taxonomy Curation via Sparse Autoencoders

Gpt o4-mini analyzes the sentences in each “cluster”

Extracts cluster title and description



Examples of Identified Reasoning Strategies

Category	Representative Example
Recalling Mathematical Formulas	"The formula for heat transfer through conduction in a cylindrical or spherical object is given by $Q = (k * A * \Delta T * t) / d$, where Q is the heat transferred, k is the thermal conductivity, A is the surface area, ΔT is the temperature difference, t is time, and d is the thickness."
Retrieving Factual Knowledge	"I think C3 refers to a type of photosynthesis where carbon fixation happens in the stroma of the chloroplast, and C4 is another type where it happens in a specialized cell structure called the bundle sheath."
Listing considerations	"I also think about the concept of market saturation."
Conditional Outcome Projection	"So, the son's argument would be that the covenant is enforceable against him because it's a real covenant that runs with the land, and the neighbor can enforce it. The son didn't have a valid defense because he didn't record and didn't know, but knowledge isn't necessary for real covenants."
Conditional Causal Reasoning	"I think the key point is that when both aggregate supply and aggregate demand increase, the price level might decrease because the demand is pulling it down, but the supply is also increasing, which might make the price level not decrease as much as it would if only demand was increasing."
Drawing Conclusions	"So, putting it all together, I think Aristotle's philosophy is the most consistent with the idea of three major life tasks because his teachings on virtue and the structure of a good life include these areas as important components."

Finding a Steering Vector

For each reasoning strategy in the taxonomy of reasoning strategies:

- Identify the sentences from the reasoning model which employ that reasoning strategy (reasoning sentence)

  “Therefore, the old policy must be immediately reinstated." (Strategy: Drawing Conclusions)

- Extract the tokens that lead up to each reasoning sentence in each respective reasoning trace (preceding tokens)

  Ex. "Crime rose 15% following the policy change."

- Optimize a steering vector in the base model to minimize cross entropy on the reasoning sentence given the preceding tokens

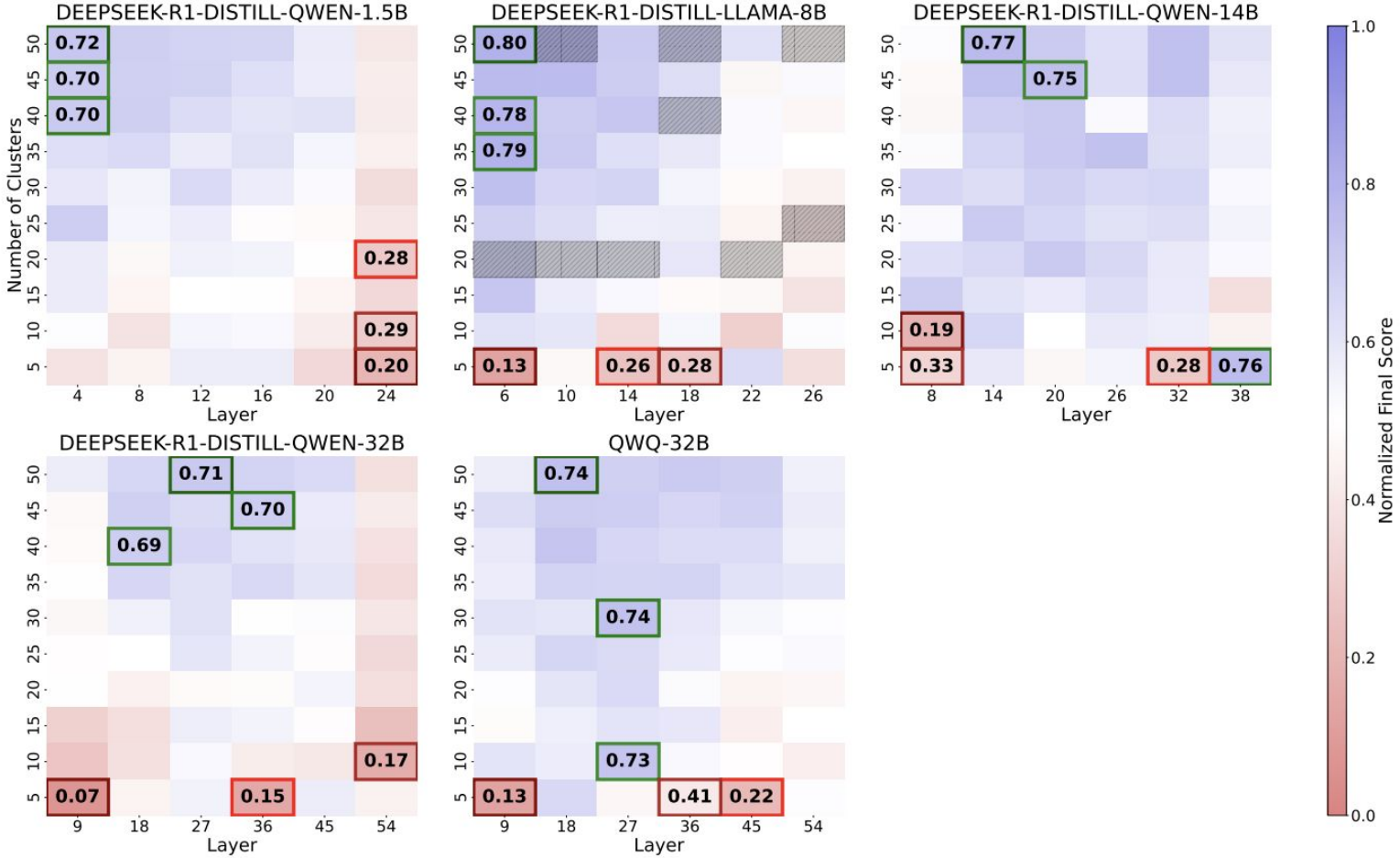
Generation with the hybrid model

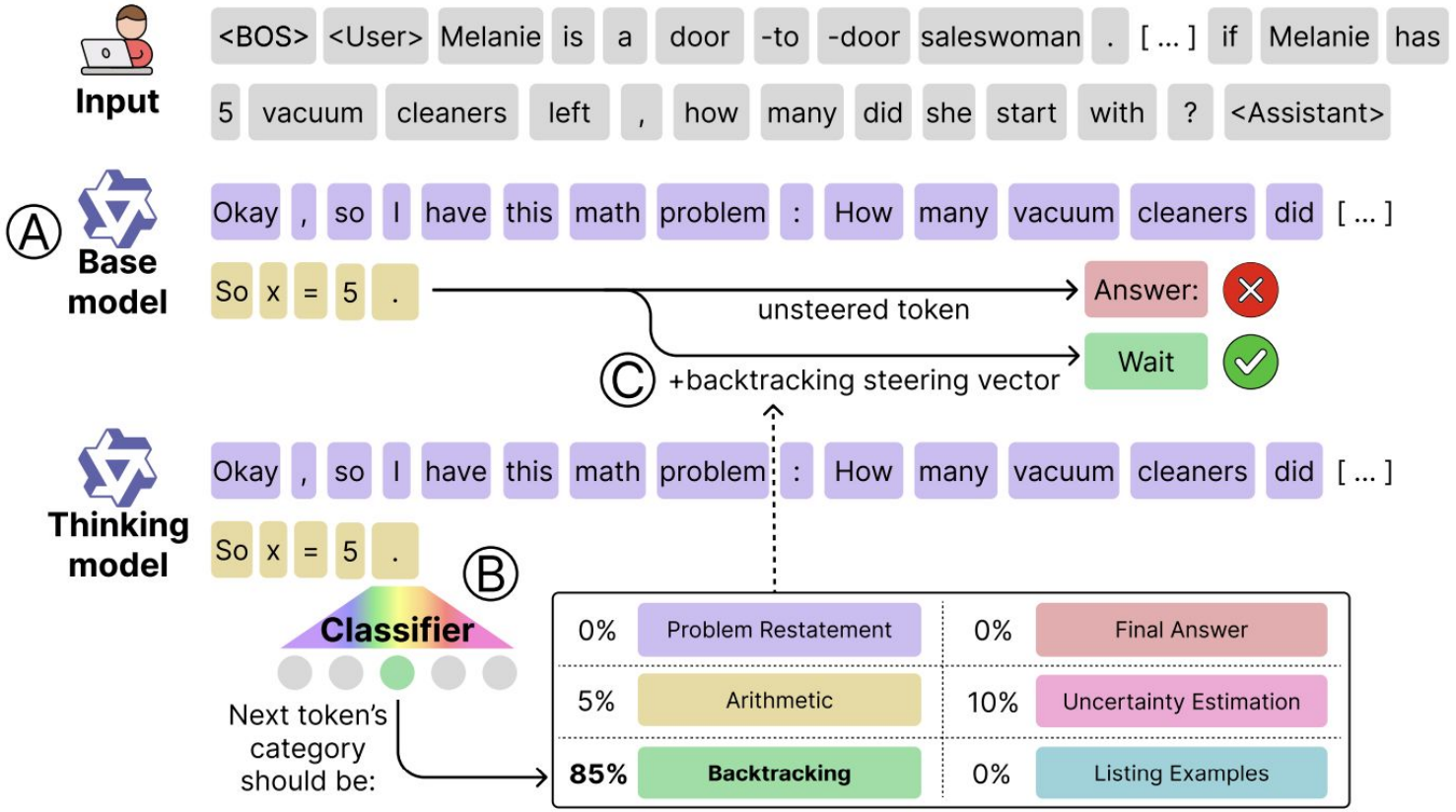
During generation:

- Use SAE to identify most active reasoning category at each token position in the base model's generated tokens

- Apply the steering vector for the strongest reasoning category to the base model activations

The hybrid model is the base model guided with steering vectors derived from the reasoning model


DEEPSEEK-R1-DISTILL-QWEN-1.5B
DEEPSEEK-R1-DISTILL-LLAMA-8B
DEEPSEEK-R1-DISTILL-QWEN-14B
DEEPSEEK-R1-DISTILL-QWEN-32B
QWQ-32B
Number of Clusters
Layer
Normalized Final Score
0.72
0.70
0.70
0.28
0.29
0.20
0.80
0.78
0.79
0.13
0.26
0.28
0.77
0.75
0.19
0.33
0.28
0.76
0.71
0.70
0.69
0.17
0.07
0.15
0.74
0.74
0.73
0.13
0.41
0.22


Input
<BOS> <User> Melanie is a door -to -door saleswoman . [...] if Melanie has
5 vacuum cleaners left , how many did she start with ? <Assistant>
Ⓐ
Base model
Okay , so I have this math problem : How many vacuum cleaners did [...]
So x = 5 .
unsteered token
Answer:
Ⓒ
+backtracking steering vector
Wait
Thinking model
Okay , so I have this math problem : How many vacuum cleaners did [...]
So x = 5 .
Ⓑ
Classifier
Next token's category should be:
0% Problem Restatement
0% Final Answer
5% Arithmetic
10% Uncertainty Estimation
85% Backtracking
0% Listing Examples

Articulating the Next Subgoal | Stating Known Equations | Planning Next Steps | Numeric computations | Presenting Conclusions | Adjusting problem framing | Proposing Causal Hypotheses

Base Response (Unsteered)

To find the distance between two points in a coordinate plane, we can use the distance formula, which is derived from the Pythagorean theorem. The distance formula is:

d = √[(x2 − x1)² + (y2 − y1)²]

where (x1, y1) and (x2, y2) are the coordinates of the two points.

In this case, the coordinates of the two points are (2, −6) and (−4, 3). Let's plug these values into the distance formula:

d = √[(−4 − 2)² + (3 − (−6))²]

Simplifying the expression inside the square root:

d = √[(−6)² + (9)²]

d = √[36 + 81]

d = √117

Since 117 is not a perfect square, we cannot simplify the square root any further. Therefore, the distance between the two points is √117 units.

Hybrid Response (Steered)

To find the distance between two points in a coordinate plane, we can use the distance formula. The distance formula is derived from the Pythagorean theorem and is given by:

\[d = \sqrt{(x_2 − x_1)^2 + (y_2 − y_1)^2} \]

Here, the coordinates of the first point are \((x_1, y_1) = (2, −6)\) and the coordinates of the second point are \((x_2, y_2) = (−4, 3)\).

First, we need to find the difference in the x-coordinates and the y-coordinates:

\[x_2 − x_1 = −4 − 2 = −6 \]
\[y_2 − y_1 = 3 − (−6) = 3 + 6 = 9 \]

Next, we square these differences:

\[(−6)^2 = 36 \]
\[9^2 = 81 \]

Now, we add these squared differences together:

\[36 + 81 = 117 \]

Finally, we take the square root of the sum to find the distance:

\[d = \sqrt{117} \]

Table 2: **Hybrid model performance on MATH500.** Results show accuracy percentages for base models, hybrid models (base + steering vectors), and thinking models. Performance improvements over base model are shown in parentheses next to hybrid and thinking model results.

Base Model	Thinking Model	Base	Hybrid	Thinking	Gap Recovery
Qwen2.5-Math-1.5B	DeepSeek-R1-Distill-Qwen-1.5B	66.2%	68.4% (+2.2%)	78.6% (+12.4%)	**17.7%**
Llama-3.1-8B	DeepSeek-R1-Distill-Llama-8B	27.8%	29.6% (+1.8%)	79.8% (+52.0%)	**3.5%**
Qwen2.5-14B	DeepSeek-R1-Distill-Qwen-14B	58.6%	75.4% (+16.8%)	86.4% (+27.8%)	**60.4%**
Qwen2.5-32B	DeepSeek-R1-Distill-Qwen-32B	59.4%	74.6% (+15.2%)	86.0% (+26.6%)	**57.1%**
Qwen2.5-32B	QwQ-32B	63.4%	84.4% (+21.0%)	86.4% (+23.0%)	**91%**

Table 3: **Average steered fraction of tokens per problem.** Average fraction of tokens receiving steering per problem on GSM8K and MATH500 for each base/thinking pair.

Base Model	Thinking Model	GSM8K	MATH500
Qwen2.5-Math-1.5B	DeepSeek-R1-Distill-Qwen-1.5B	12.8%	10.7%
Llama-3.1-8B	DeepSeek-R1-Distill-Llama-8B	21.5%	15.8%
Qwen2.5-14B	DeepSeek-R1-Distill-Qwen-14B	6.5%	7.8%
Qwen2.5-32B	DeepSeek-R1-Distill-Qwen-32B	11.2%	9.8%
Qwen2.5-32B	QwQ-32B	12.7%	12.0%

Paper Summary

1. Presents a taxonomy of reasoning behaviors that thinking (also known as reasoning) models exhibit when doing chain of thought

2. Using the taxonomy, derives ‘steering vectors’ for each reasoning method

3. **Demonstrates that a hybrid approach of base model + steering vectors allows the base model perform every reasoning method that a reasoning model employs**

Reasoning with Sampling: Your Base Model is Smarter Than You Think

Hypotheses

1. Entirely new reasoning abilities are acquired after RL.
2. Repurposing pre-existing representations for new mechanisms.
3. **“Amplifying” behaviors learned in pretraining.**

If base models already have the knowledge necessary for reasoning, we can skip time-intensive RL/ SFT and use training-free methods to elicit desired behavior from base models.

Intuition

- RL'd models are much better at reasoning tasks than base models. But *why?*
- Prior work, empirical observation→ $p_{RL}(x)$ looks like base p(x) but "sharpened"
- If so, is there any other way to "*sharpen*" and "*sample*" from the distribution?
- Authors present that sampling directly from the *sharpened* base model can achieve single-shot reasoning capabilities on par with those from RL
- They propose sampling algorithm is *training-free, dataset-free, and verifier-free*

Sharpening with Power Distributions

- Simply means re-weighting the distribution
- So that high likelihood sequences are further upweighted
- While low likelihood ones are downweighted

→ Bias the samples ***heavily*** towards higher likelihoods

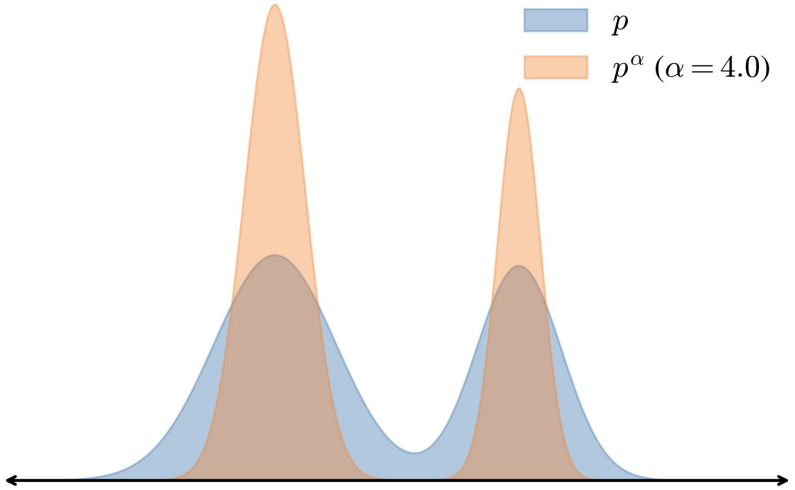


Figure 2: **A toy example of distribution sharpening.** Here p is a mixture of Gaussians, which we plot against p^{α} $(\alpha = 4.0)$.

Sharpening with Power Distributions

Example: For a given prompt, let base probabilities over possible sequences be:

$p(x^{(1)})$ = **0.50** → For the 1st possible sequence
$p(x^{(2)})$ = **0.30** → 2nd
$p(x^{(3)})$ = **0.20** → 3rd

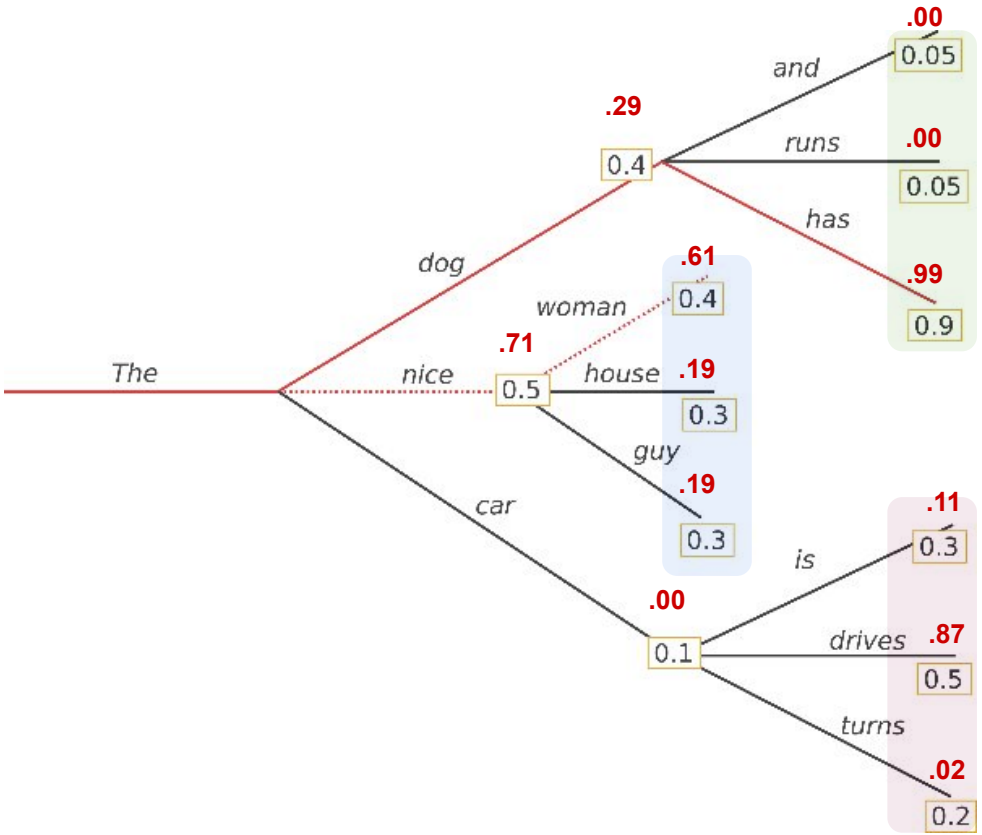
Let's pick α = 2:

$p_\alpha(x^{(1)}) \propto 0.50^2 = 0.25$, $p_\alpha(x^{(2)}) \propto 0.30^2 = 0.09$, $p_\alpha(x^{(3)}) \propto 0.20^2 = 0.04$

Sum = 0.38 → normalizing:

$p_\alpha(x^{(1)})$ = **0.66**, $p_\alpha(x^{(2)})$ = **0.24**, $p_\alpha(x^{(3)})$ = **0.10**

Temp. Sampling vs Power Distribution



	Base	Temp	Pow.
and	.02	.00	.00
runs	.02	.00	.00
has	.36	.29	.87
woman	.20	.43	.08
house	.15	.14	.02
guy	.15	.14	.02
is	.03	.00	.00
drives	.05	.00	.00
turns	.02	.00	.00

Why is this non-trivial?

- There are exponentially many possible sequences x
- We cannot enumerate them all or normalize $p(x)^{\alpha}$
- However, we **can** compute $p(x)^{\alpha}$ for any given completed answer x
- Still, we don't know the normalization constant or the whole space
- Enter Metropolis-Hastings (MH) - a Markov Chain Monte Carlo (MCMC) algorithm

The Metropolis-Hastings (MH) Algorithm

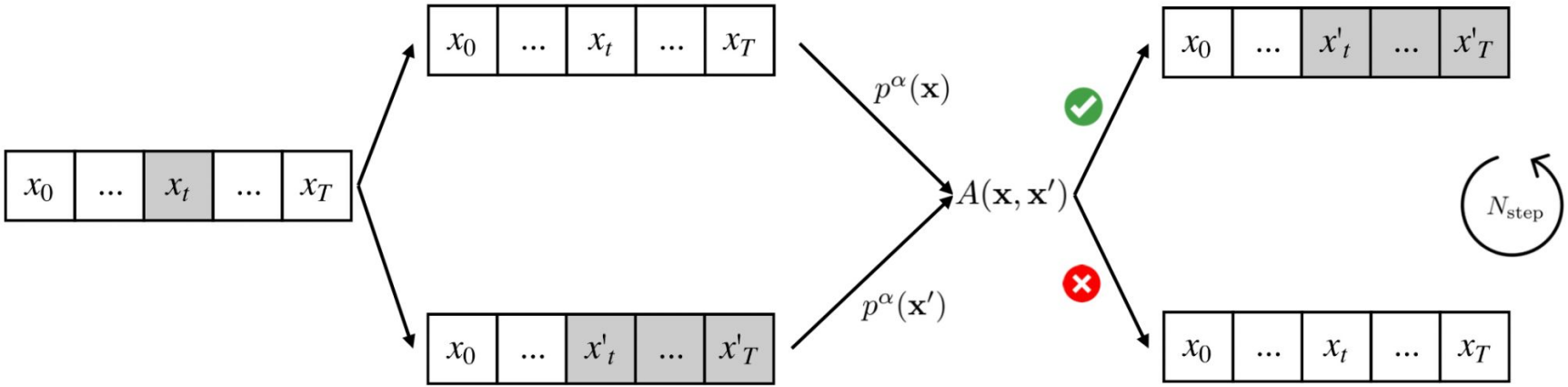


Figure 3: **Illustrating Metropolis-Hastings with random resampling.** A random index t is selected and a new candidate is generated by resampling. Based on the relative likelihoods, the candidate is accepted or rejected, and the process repeats.

$$A(\mathbf{x}, \mathbf{x}^i) = \min\left\{1, \frac{p^\alpha(\mathbf{x}) \cdot q(\mathbf{x}^i|\mathbf{x})}{p^\alpha(\mathbf{x}^i) \cdot q(\mathbf{x}|\mathbf{x}^i)}\right\}$$

The Metropolis-Hastings (MH) Algorithm

- Sample from a distribution proportional to $p(x)^{\alpha}$ instead of exactly $p(x)^{\alpha}$ using only **ratios** of the probabilities of candidate and current sequences, so the normalization constant cancels out

1. Start at some sequence $x^{(0)}$
2. Select a random index *t* in $x^{(0)}$, propose a new candidate sequence x' produced by generating rest of $x^{(0)}$ using some proposal distribution $q(x' \mid x^{(i)})$
3. Compute the acceptance probability
4. Draw u ~ Uniform(0, 1)
   a. If u <= A(x, $x^{(i)}$), accept and set $x^{(i+1)} = x'$
   b. Otherwise reject and set $x^{(i+1)} = x^{(i)}$t
5. If repeated many times, the state of the chain (current full answer), will in the long run be distributed according to our target distribution proportional to $p(x)^{\alpha}$

Autoregressive MCMC

Algorithm 1: Power Sampling for Autoregressive Models

Input : base p; proposal p_{prop}; power α; length T

Hyperparams: block size B; MCMC steps N_{MCMC}

Output : $(x_0, \ldots, x_T) \sim p^{\alpha}$

Notation: Define the unnormalized intermediate target

$$\pi_k(x_{0:kB}) \;\propto\; p(x_{0:kB})^{\alpha}.$$

for $k \leftarrow 0$ **to** $\lceil \frac{T}{B} \rceil - 1$ **do**

  Given prefix $x_{0:kB}$, we wish to sample from π_{k+1}. Construct initialization $\mathbf{x}^0$ by extending autoregressively with p_{prop}:

$$x_t^{(0)} \sim p_{\text{prop}}\big(x_t \mid x_{<t}\big), \qquad \text{for } kB+1 \le t \le (k+1)B.$$

  Set the current state $\mathbf{x} \leftarrow \mathbf{x}^0$.

  for $n \leftarrow 1$ **to** N_{MCMC} **do**

    Sample an index $m \in \{1, \ldots, (k+1)B\}$ uniformly.

    Construct proposal sequence $\mathbf{x}'$ with prefix $x_{0:m-1}$ and resampled completion:

$$x'_t \sim p_{\text{prop}}\big(x_t \mid x_{<t}\big), \qquad \text{for } m \le t \le (k+1)B.$$

    Compute acceptance ratio (9)

$$A(\mathbf{x}', \mathbf{x}) \;\leftarrow\; \min\left\{1,\; \frac{\pi_k(\mathbf{x}')}{\pi_k(\mathbf{x})} \cdot \frac{p_{\text{prop}}(\mathbf{x} \mid \mathbf{x}')}{p_{\text{prop}}(\mathbf{x}' \mid \mathbf{x})}\right\}.$$

    Draw $u \sim \text{Uniform}(0, 1)$;

    if $u \le A(\mathbf{x}', \mathbf{x})$ **then accept** and set $\mathbf{x} \leftarrow \mathbf{x}'$

  end

  Set $x_{0:(k+1)B} \leftarrow \mathbf{x}$ to fix the new prefix sequence for the next stage.

end

return $x_{0:T}$

```
Original (x):
The quick brown fox jumps over the lazy
```

Illustration of Metropolis-Hastings updates.

Question: If a square has side length four, what is its area?
Answer: The area is found

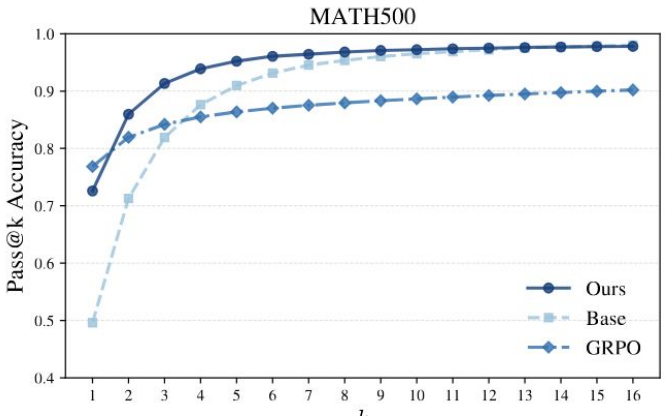
Illustration of block-wise autoregressive MCMC sampling.

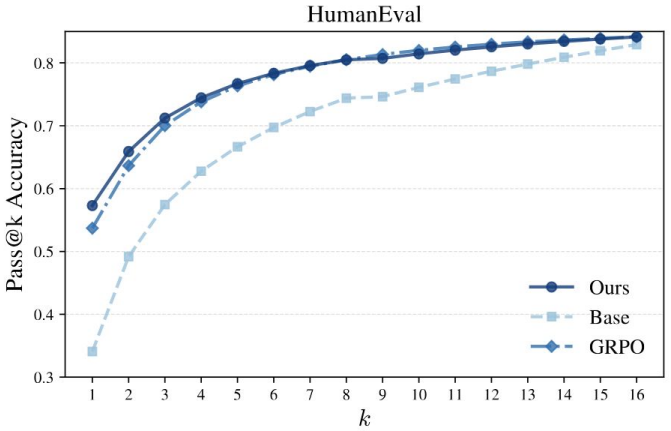
Results

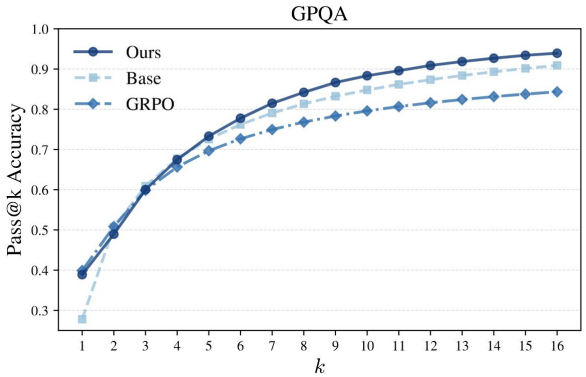
- Outperforms GRPO on the Phi model
- Better results on out of domain tasks
- Except GPQA

	MATH500	HumanEval	GPQA	AlpacaEval2.0
Qwen2.5-Math-7B				
Base	0.496	0.329	0.278	1.61
Low-temperature	0.690	0.512	0.353	2.09
Power Sampling (ours)	**0.748**	**0.573**	**0.389**	**2.88**
GRPO (MATH)	**0.785**	**0.537**	**0.399**	**2.38**
Qwen2.5-7B				
Base	0.498	0.329	0.278	7.05
Low-temperature	0.628	0.524	0.303	5.29
Power Sampling (ours)	**0.706**	**0.622**	**0.318**	**8.59**
GRPO (MATH)	**0.740**	**0.561**	**0.354**	**7.62**
Phi-3.5-mini-instruct				
Base	0.400	0.213	0.273	14.82
Low-temperature	0.478	0.585	0.293	**18.15**
Power Sampling (ours)	**0.508**	**0.732**	**0.364**	**17.65**
GRPO (MATH)	**0.406**	**0.134**	**0.359**	**16.74**

Pass @ k

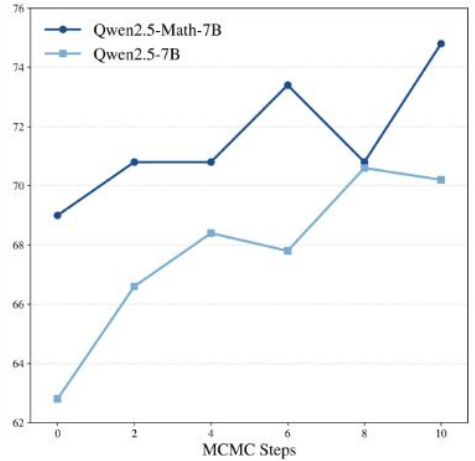

MATH500
Pass@k Accuracy
Ours
Base
GRPO

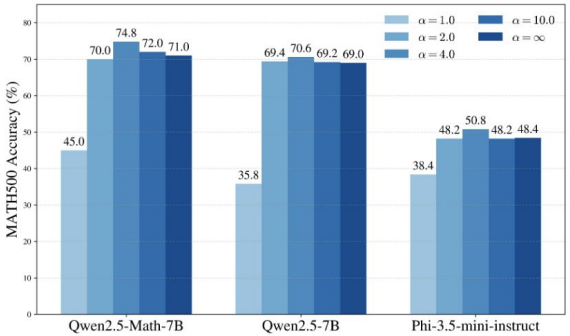

HumanEval
Pass@k Accuracy
k
Ours
Base
GRPO


GPQA
Pass@k Accuracy
k
Ours
Base
GRPO

Computational cost

- Average output length:
  - Base: 600
  - GRPO: 671
  - Sampling: 679
- Sampling generates 8.84x #tokens compared to the base model.
  - correction : 11.34x
- Leading to comparable inference costs as one epoch of GRPO (8-16 rollouts)





Discussion Questions

- Should we still care about RL and SFT?
  - Is there any reason to believe that reasoning models do things that base models will never be able to?

- What counts as “learning to reason”?
  - If RL simply ‘drives’ or ‘reweights’ existing behaviors, is that qualitatively different from how humans learn to reason?


JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING
Thank you!