Mathematics of Foundation Models

Lessons from Structured Problems

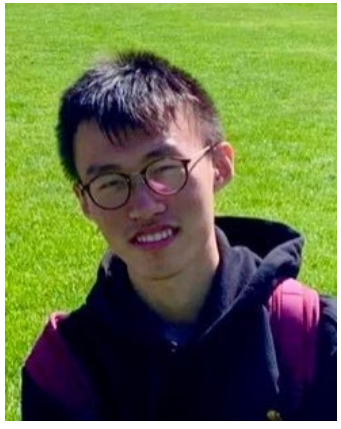
Yuejie Chi

Yale Statistics & Data Science



SIAM Conference on Mathematics of Data Science
Salt Lake City, November 2026

My brilliant student collaborators



Zixin Wen

CMU



Xingyu Xu

CMU



Ivan Lee

Yale

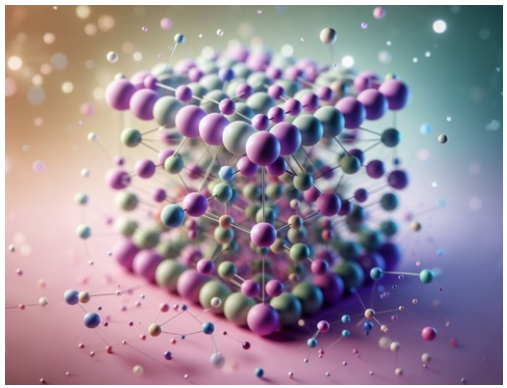


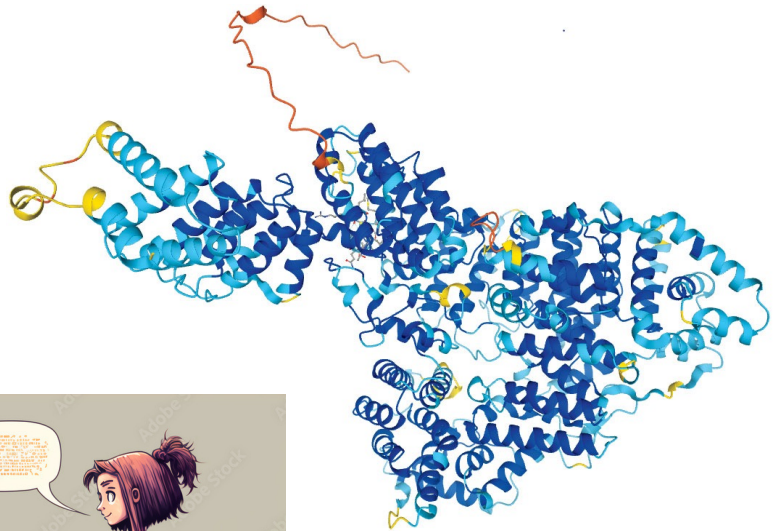
Yu Huang

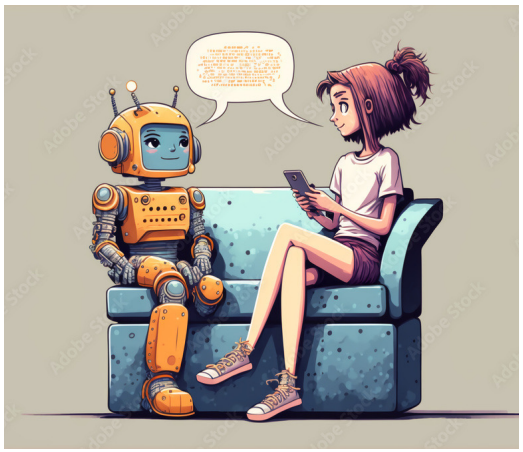
Penn

Along with faculty collaborators: Yuxin Chen (Penn), Aarti Singh (CMU), Yingbin Liang (OSU), Yuting Wei (Penn).

Generative AI











Generative AI is changing how we create, communicate, and do science.

Foundation models

The universe is a grand book written in the language of...

Autoregressive models

The universe is a grand book written in the language of mathematics.

Diffusion models



Strong reasoning capabilities

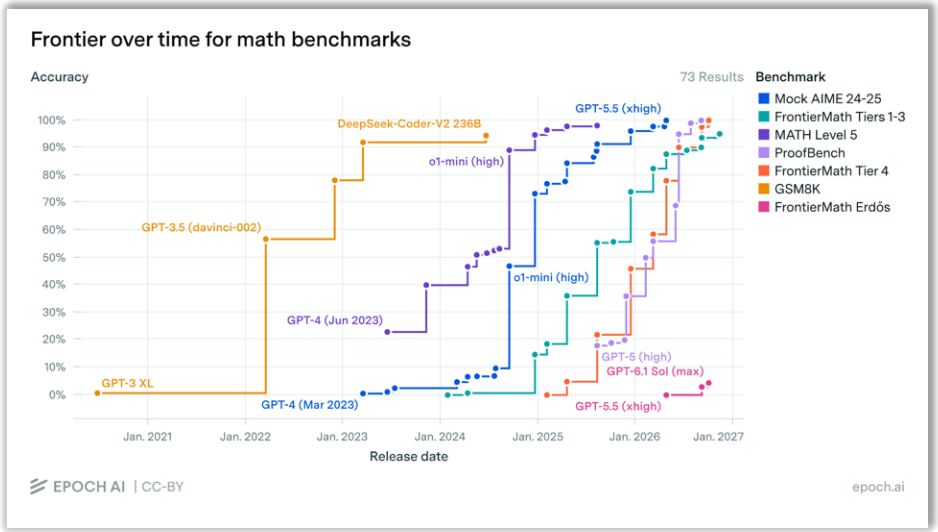


SCIENCEINSIDER | TECHNOLOGY

AI cracks a 27-year-old pillar of computer science, OpenAI and Anthropic claim

The result, one of three major claims from OpenAI, raises questions about sharing of AI-generated work

28 OCT 2026 · 5:25 PM ET · BY TOM HOWARTH



The models now can solve frontier math problems.

Amazing expressive power



The models now can generate highly complex and realistic scenes.

Towards building a mathematical understanding

The dynamics of information flow:
How is information encoded and processed by foundation models?

Towards building a mathematical understanding

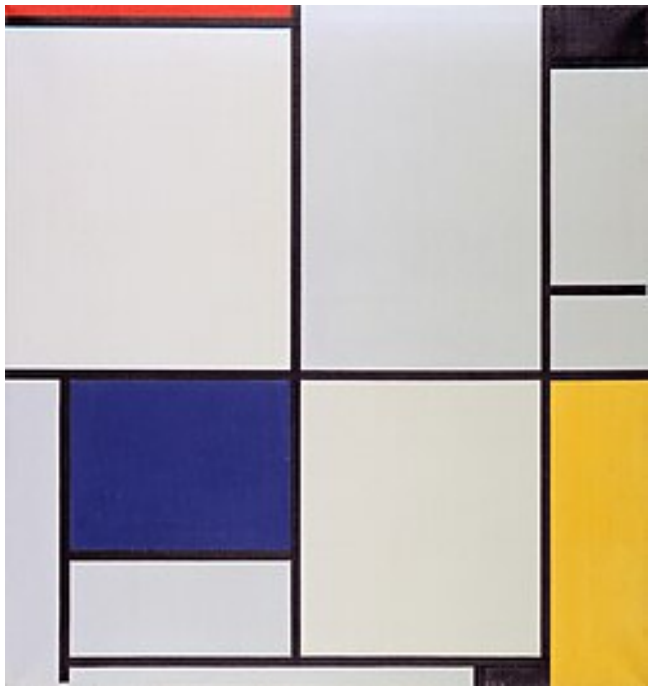
The dynamics of information flow:
How is information encoded and processed by foundation models?



Complex AI systems with many interacting components

Towards building a mathematical understanding

The dynamics of information flow:
How is information encoded and processed by foundation models?



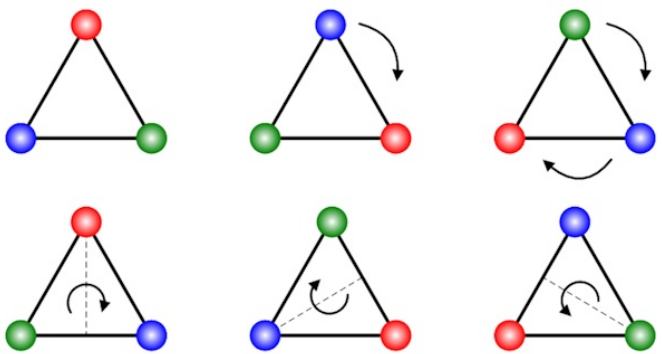
Stylized models to isolate mechanisms and gain deeper understanding

This talk: two vignettes

The dynamics of information flow:
How is information encoded and processed by foundation models?

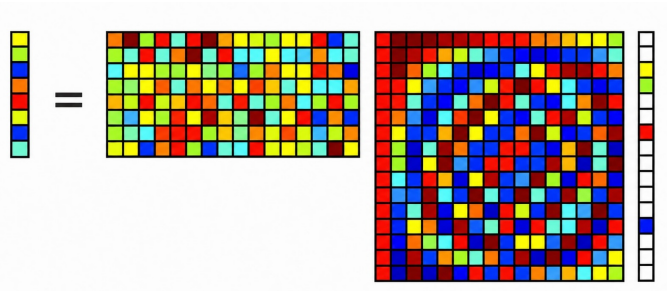
Reasoning models

How do transformers learn to reason step-by-step?



Diffusion models

Are diffusion models efficient priors for inverse problems?

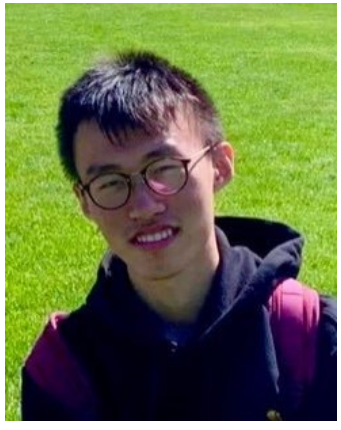


Mathematics of reasoning models: How do transformers learn to reason step-by-step?



Yu Huang

Penn



Zixin Wen

CMU

Two training paradigms for reasoning models

What is 17×24?

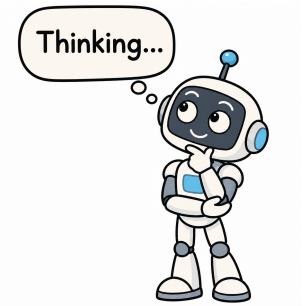
Thinking...

First, $17 \times 20 = 340$.

Next, $17 \times 4 = 68$.

Adding the two gives $340 + 68 = 408$.

The answer is $\mathbf{408}$.



Two training paradigms for reasoning models

What is 17×24?

reasoning trace

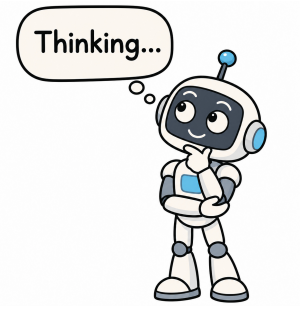
Thinking...

First, $17 \times 20 = 340$.

Next, $17 \times 4 = 68$.

Adding the two gives $340 + 68 = 408$.

The answer is $\mathbf{408}$.



Supervised finetuning (SFT)

Next-token prediction over the reasoning trace

Two training paradigms for reasoning models

What is 17×24?

reasoning trace

Thinking...

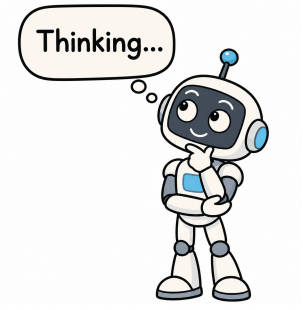
First, $17 \times 20 = 340$.

Next, $17 \times 4 = 68$.

Adding the two gives $340 + 68 = 408$.

final answer

The answer is $\mathbf{408}$.



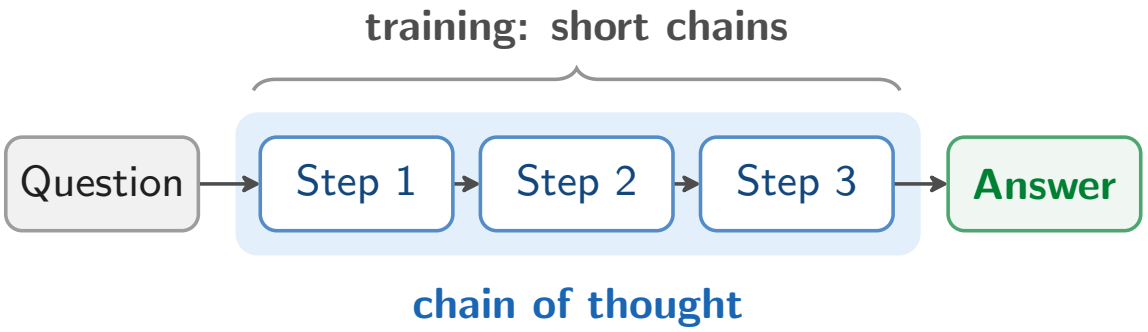
Supervised finetuning (SFT)

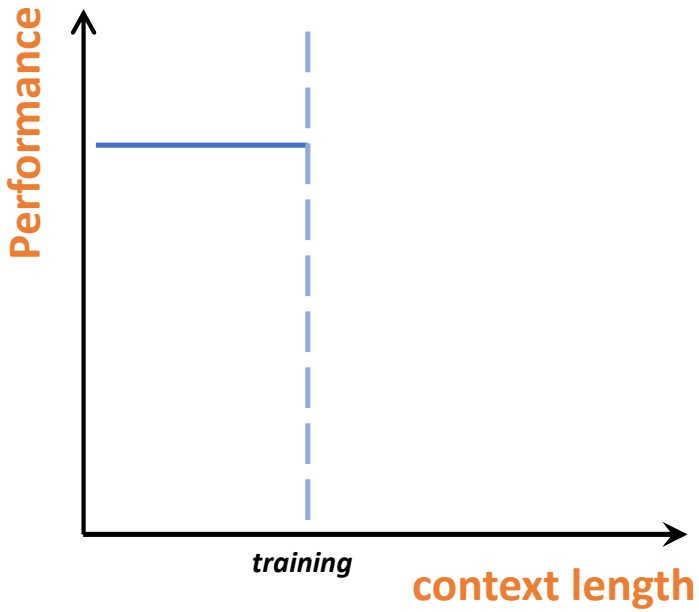
Next-token prediction over the reasoning trace

RL from verifiable reward (RLVR)

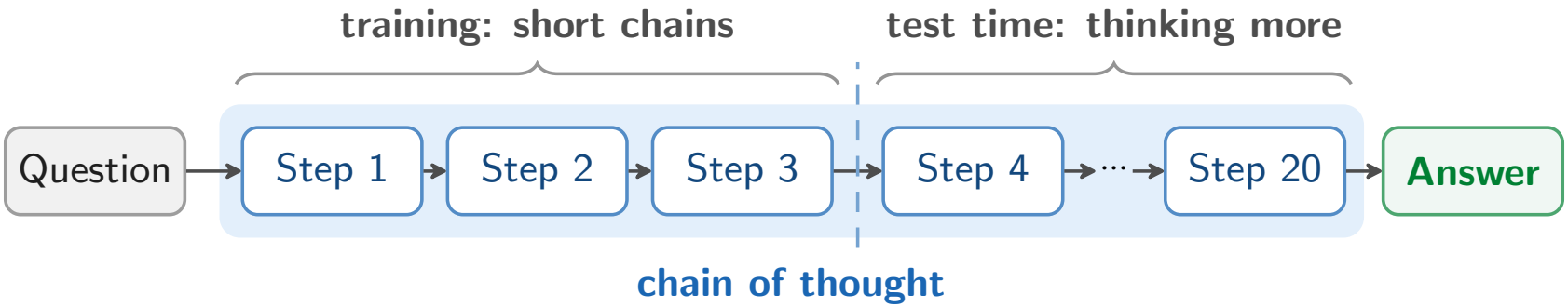
Reward only when the final answer is correct

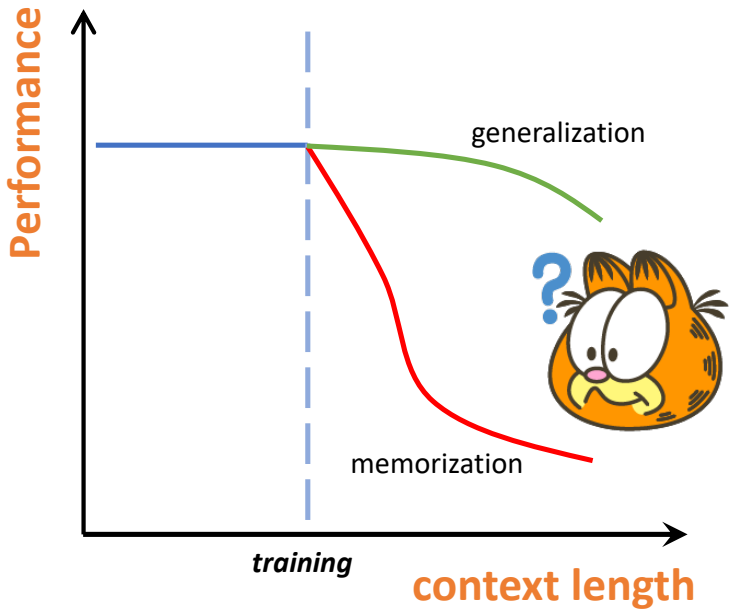
Train shorter, think longer?





Train shorter, think longer?



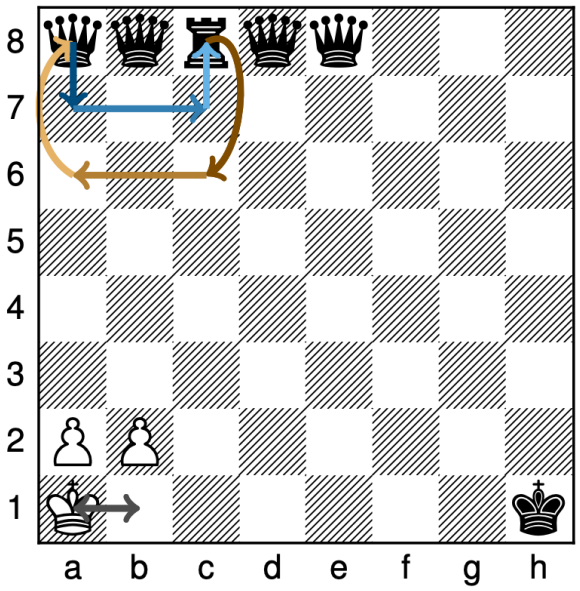


Can models trained on shorter tasks be generalized to longer ones?

Sandbox: state tracking

Given an initial *state* and a sequence of *actions*, compute the final *state*.

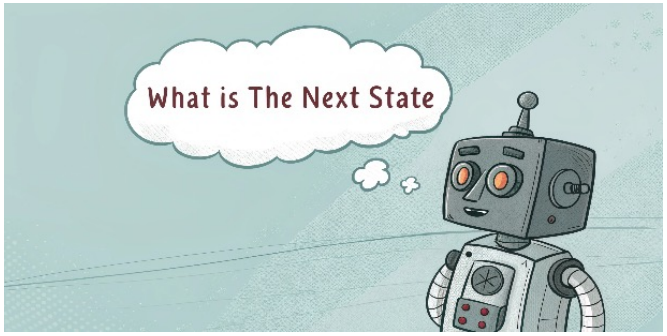
Playing Chess



Images modified from Merrill et al., 2024

Code evaluation

```
x = [0, 0, 1, 0, 0]
x[1], x[3] = x[3], x[1] # Swap 1, 3
```



Entities tracking

Alice, Bob, Carl, Dan, and Emma each have a coin. All are dimes except Carl's. Alice and Carl trade coins.

State tracking via LEGO

LEGO (Learning Equality and Group Operations) (Zhang et al., 2022):

$$\underbrace{\boxed{x_1 = g_1(x_0)}, \ldots, \boxed{x_L = g_L(x_{L-1})}, \boxed{x_0 = y_0}}_{\text{prompt}}$$

with action $g_i \in \mathcal{G}$, value $y_i \in \mathcal{Y}$, variable $x_i \in \mathcal{X}$.

State tracking via LEGO

LEGO (Learning Equality and Group Operations) (Zhang et al., 2022):

$$\underbrace{x_1 = g_1(x_0), \ldots, x_L = g_L(x_{L-1}), x_0 = y_0}_{\text{prompt}} \underbrace{x_1 = y_1, x_2 = y_2, \ldots}_{\text{CoT}}$$

with action $g_i \in \mathcal{G}$, value $y_i \in \mathcal{Y}$, variable $x_i \in \mathcal{X}$.

State tracking via LEGO

LEGO (Learning Equality and Group Operations) (Zhang et al., 2022):

$$\underbrace{x_1 = g_1(x_0), \ldots, x_L = g_L(x_{L-1}), x_0 = y_0}_{\text{prompt}} \quad \underbrace{x_1 = y_1, x_2 = y_2, \ldots}_{\text{CoT}} \quad \underbrace{x_L = y_L}_{\text{final answer}}$$

with action $g_i \in \mathcal{G}$, value $y_i \in \mathcal{Y}$, variable $x_i \in \mathcal{X}$.

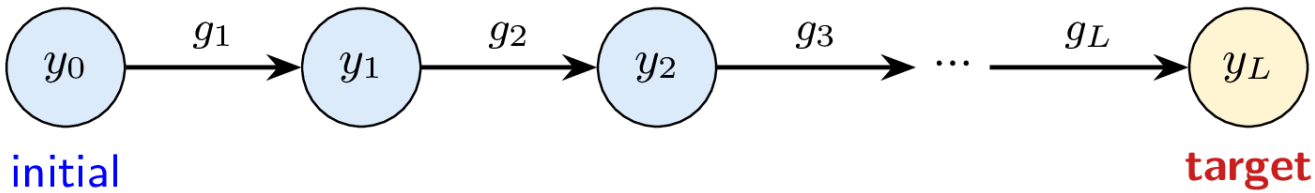
State tracking via LEGO

LEGO (Learning Equality and Group Operations) (Zhang et al., 2022):

$$\underbrace{x_1 = g_1(x_0), \ldots, x_L = g_L(x_{L-1}), x_0 = y_0}_{\text{prompt}} \underbrace{x_1 = y_1, x_2 = y_2, \ldots}_{\text{CoT}} \underbrace{x_L = y_L}_{\text{final answer}}$$

with action $g_i \in \mathcal{G}$, value $y_i \in \mathcal{Y}$, variable $x_i \in \mathcal{X}$.

Goal: correctly calculate the last value $y_L = g_L(g_{L-1}(\cdots g_1(y_0)))$.

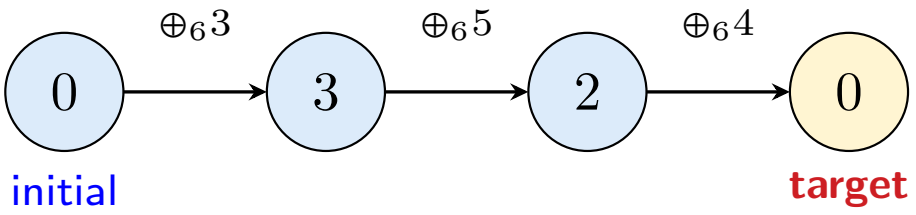


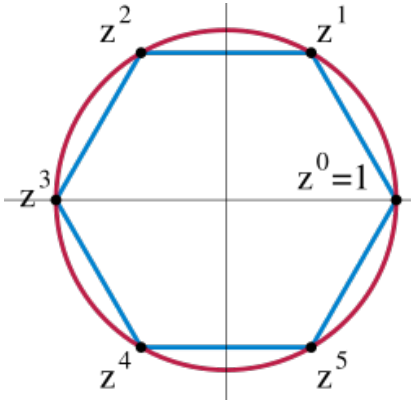
The **embedding dimension** d is *roughly* the limit of the chain length.

Two types of group actions

Simply transitive action:

- $\forall y, y'$, there is a **unique** $g \in \mathcal{G}$ s.t., $g(y) = y'$
- E.g., cyclic group acts on $[n]$.

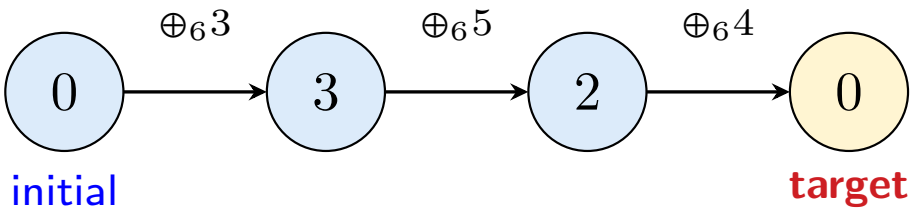


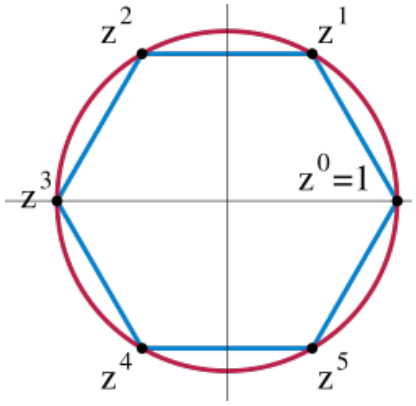


Two types of group actions

Simply transitive action:

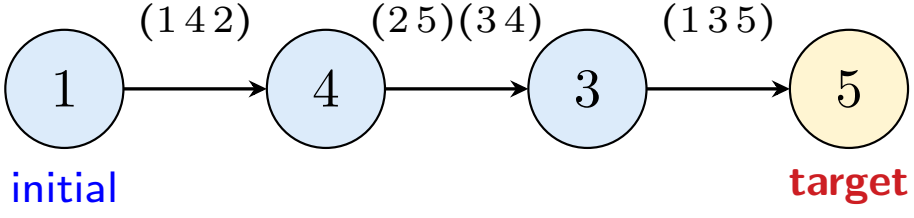
- $\forall y, y'$, there is a **unique** $g \in \mathcal{G}$ s.t., $g(y) = y'$
- E.g., cyclic group acts on $[n]$.

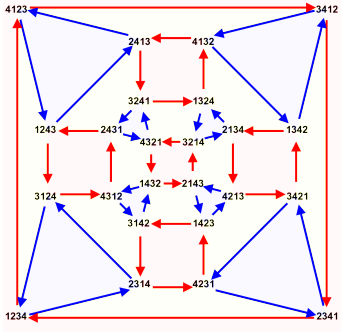




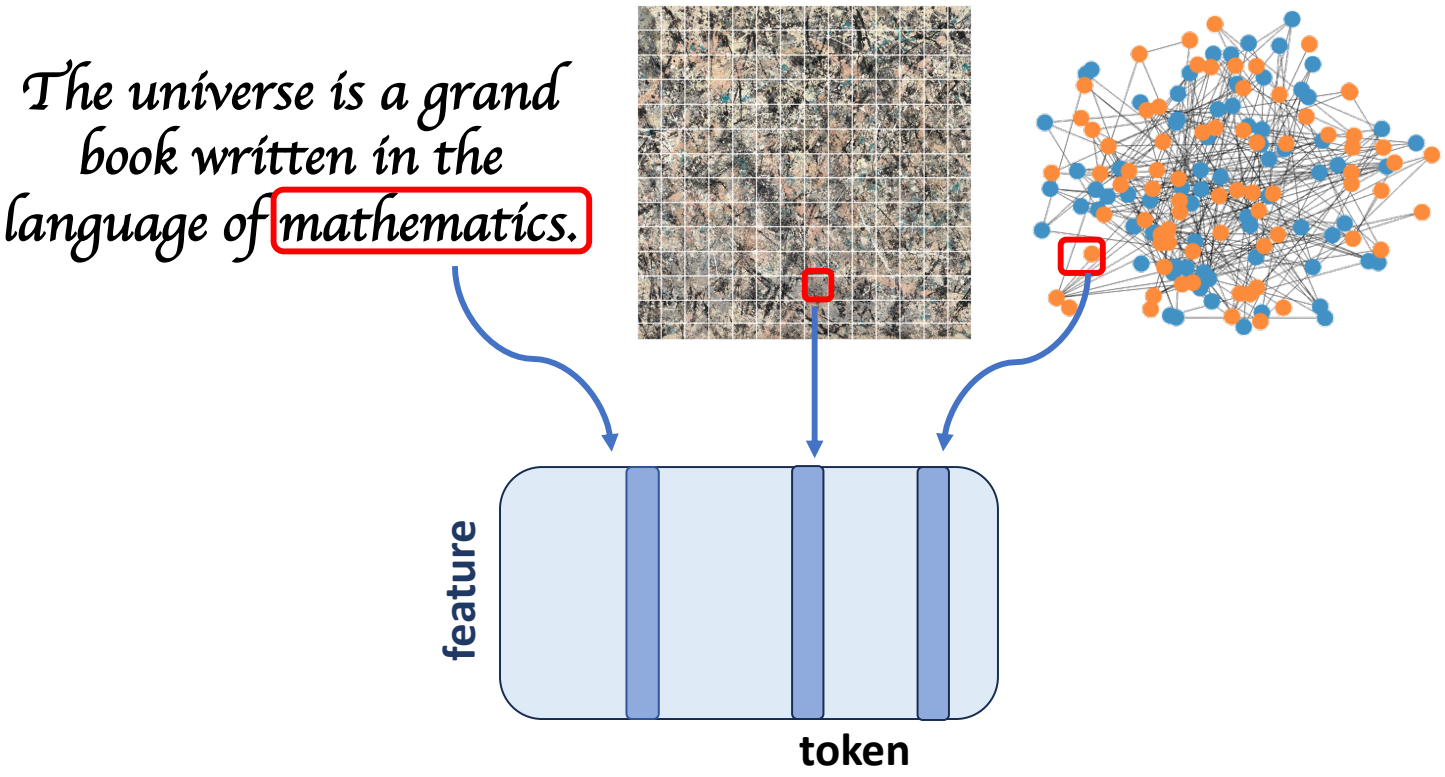
Symmetry action:

- $\forall y, y'$, there are **many** $g \in \mathcal{G}$ s.t., $g(y) = y'$
- E.g., S_n acts on $[n]$.

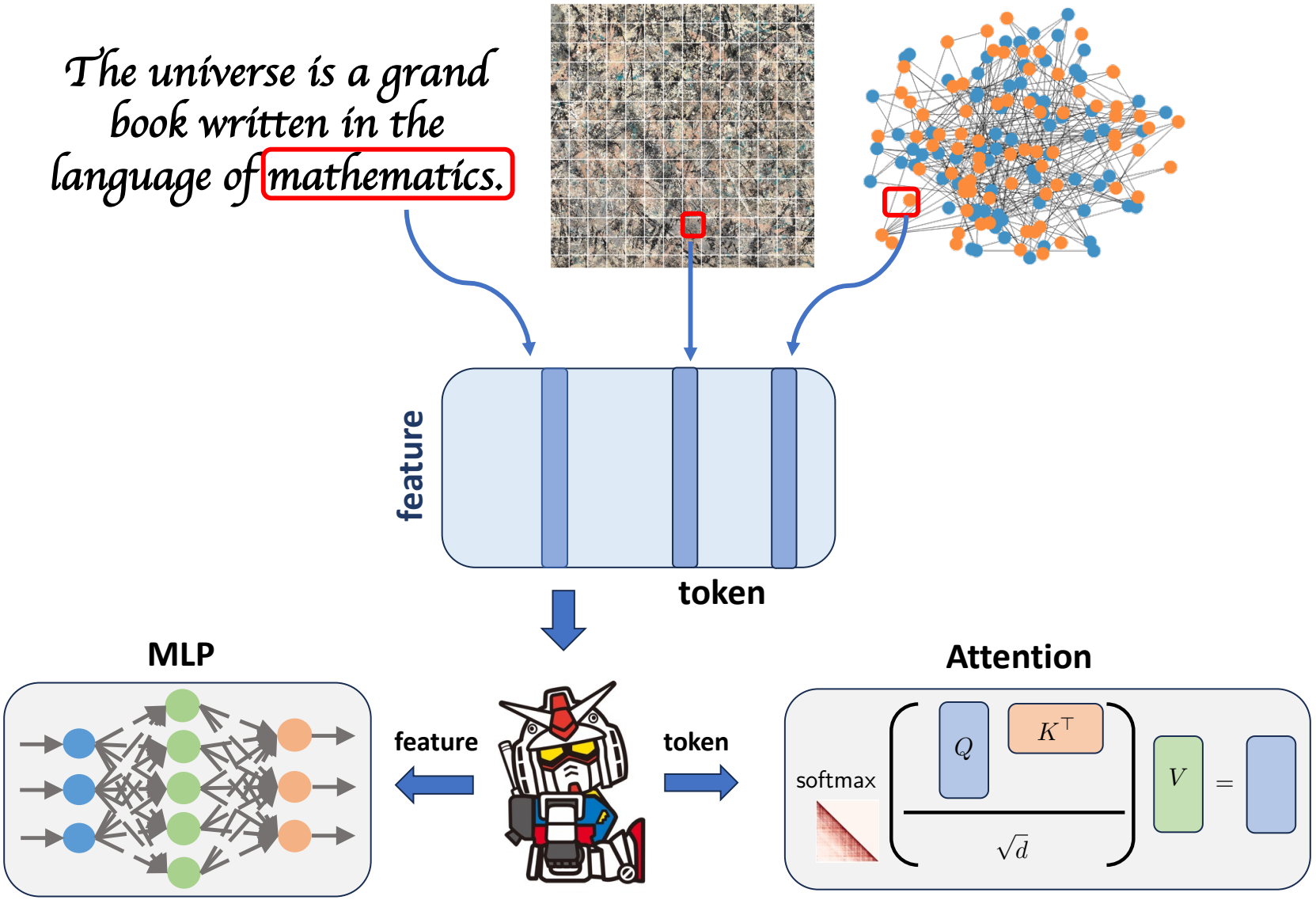




Transformers as the atom of foundation models

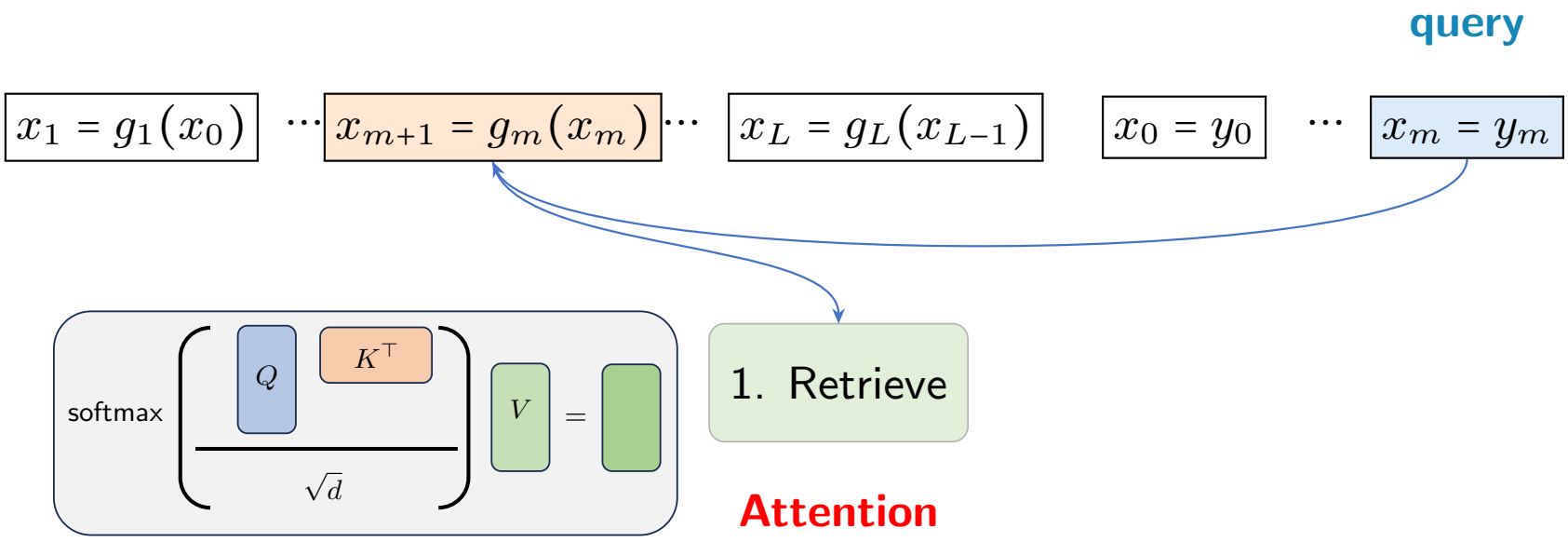


Transformers as the atom of foundation models



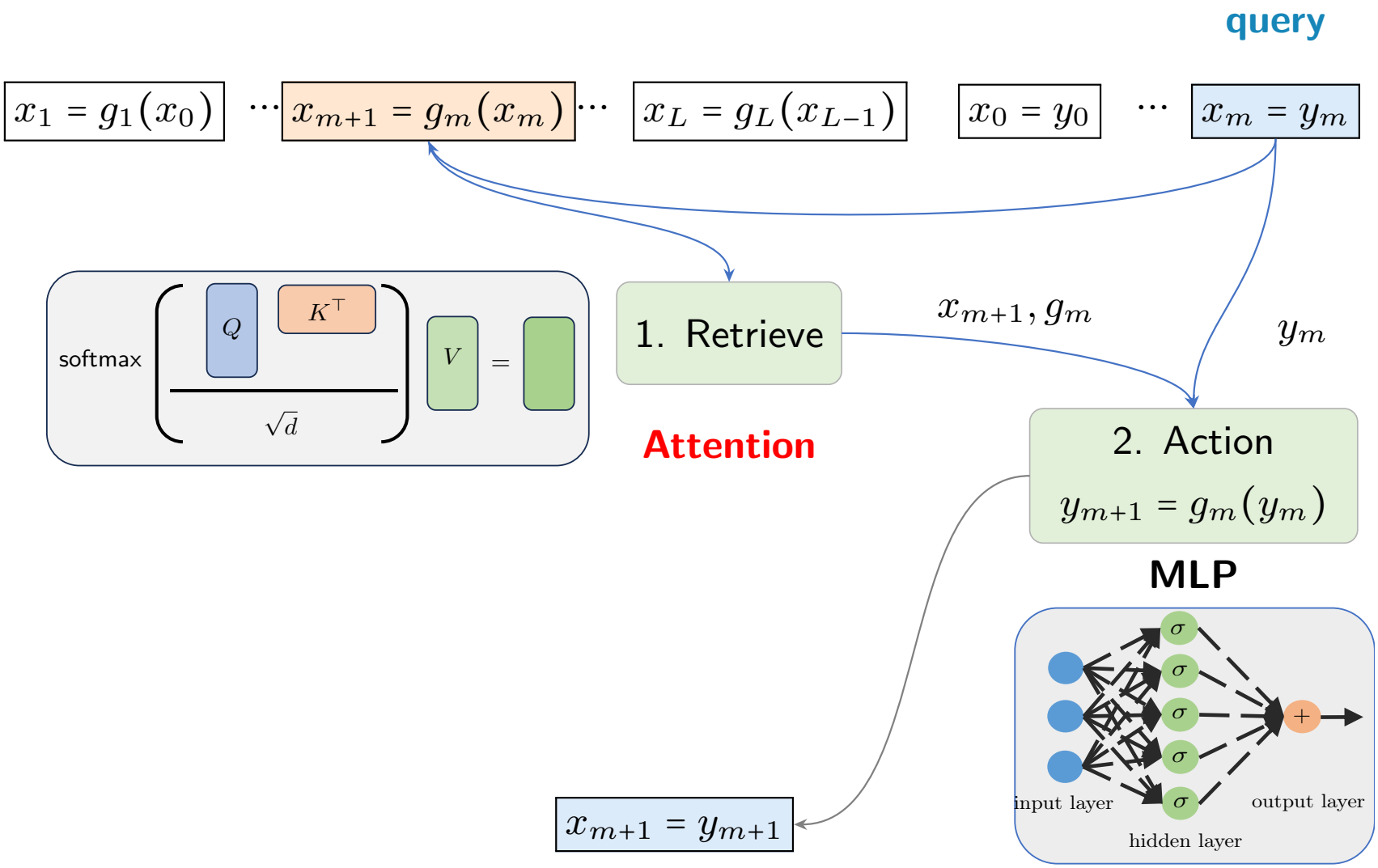
One transformer block does the magic

Mechanism: MLP perfectly captures action; while attention learns to concentrate on the **relevant** contexts.

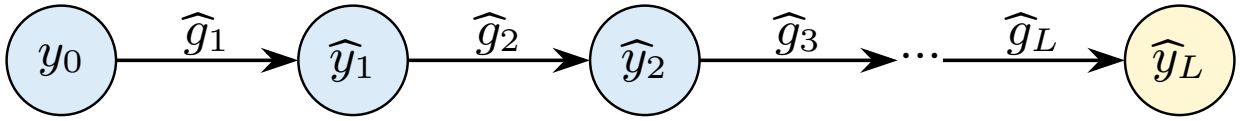


One transformer block does the magic

Mechanism: MLP perfectly captures action; while attention learns to concentrate on the **relevant** contexts.



Training objectives



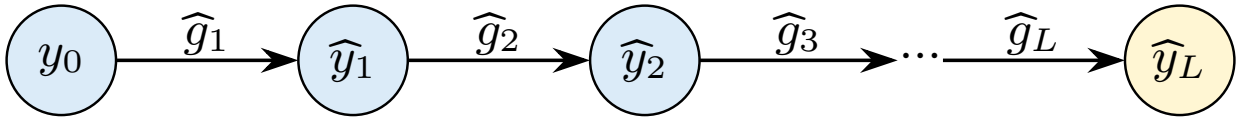
SFT objective with stepwise supervision:

$$\min_{\theta} \mathcal{L}_L(\theta) = \mathbb{E}_{\text{prompt}}\left[\, \mathbb{E}_{\widehat{y}^{1:L}\sim\pi_\theta(\cdot|\text{prompt})}\left[\, \underbrace{\sum_{m=1}^{L} -\log \mathbf{1}\{\widehat{y}_{m+1} = y_{m+1}\}}_{\text{stepwise supervision}} \,\right]\right].$$

RLVR objective with outcome reward:

$$\max_{\theta} \mathcal{J}_L(\theta) = \mathbb{E}_{\text{prompt}}\left[\, \mathbb{E}_{\widehat{y}^{1:L}\sim\pi_\theta(\cdot|\text{prompt})}\left[\, \underbrace{\mathbf{1}\{\widehat{y}_L = y_L\}}_{\text{terminal reward}} \,\right]\right]$$

Training objectives



SFT objective with stepwise supervision:

$$\min_{\theta} \mathcal{L}_L(\theta) = \mathbb{E}_{\text{prompt}}\left[\mathbb{E}_{\widehat{y}^{1:L}\sim\pi_\theta(\cdot|\text{prompt})}\left[\underbrace{\sum_{m=1}^{L} -\log \mathbf{1}\{\widehat{y}_{m+1} = y_{m+1}\}}_{\text{stepwise supervision}}\right]\right].$$

RLVR objective with outcome reward:

$$\max_{\theta} \mathcal{J}_L(\theta) = \mathbb{E}_{\text{prompt}}\left[\mathbb{E}_{\widehat{y}^{1:L}\sim\pi_\theta(\cdot|\text{prompt})}\left[\underbrace{\mathbf{1}\{\widehat{y}_L = y_L\}}_{\text{terminal reward}}\right]\right]$$

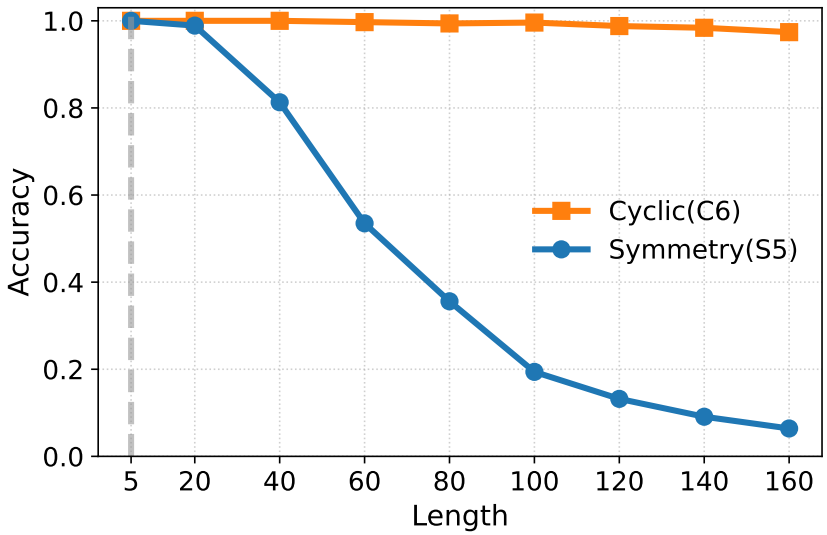
How do different training recipes impact the learning dynamics of attention?

Length generalization from SFT training

Theorem (Huang et al, 2026)

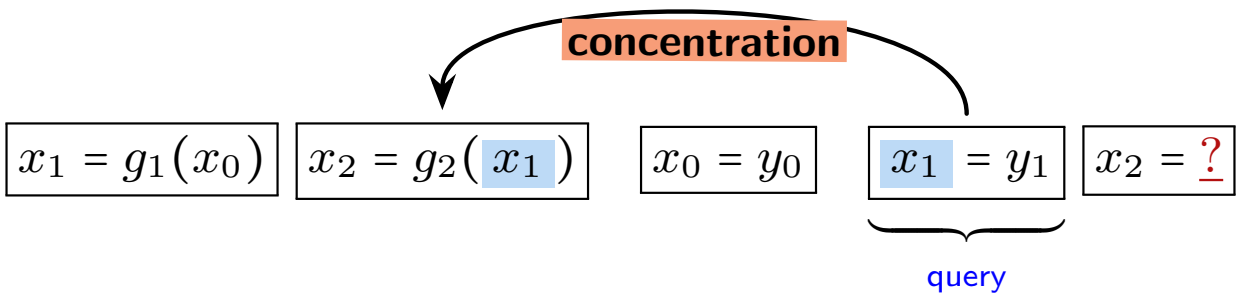
Suppose the one-layer transformer model is trained via GD for next-answer *prediction on the LEGO task of* $L = 1$ *and* $L = 2$.

- *For simply transitive actions, it* **can** *directly solve tasks of length* $L \leq O(d^c)$ *with some constant* $0 < c < 1$.
- *For symmetry actions, it* **cannot** *solve task beyond some constant length* $L = O(1)$.



Attention concentration governs length generalization

Simply transitive actions: unique action to focus on from context.



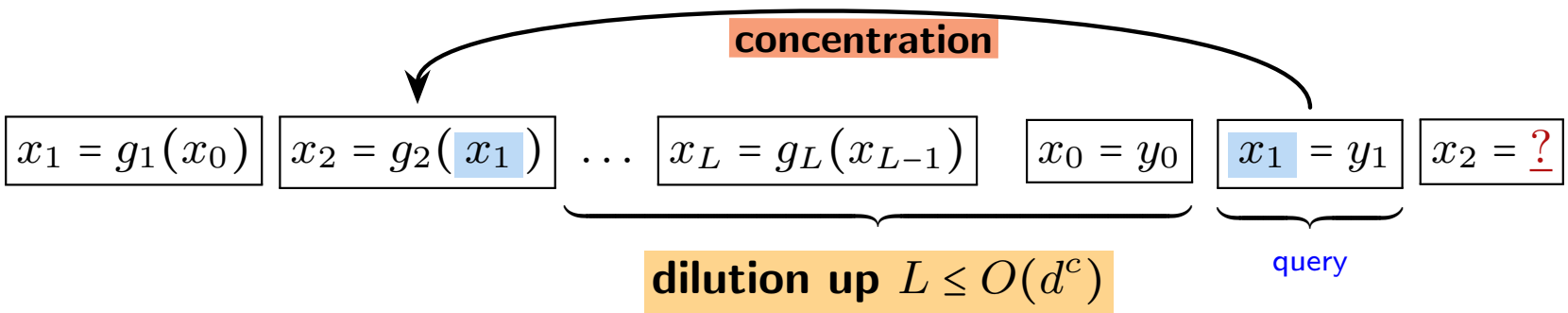
Sharp attention concentration at the end of training:

$$\mathbf{Attn}_{\text{correct}} \geq 1 - O(d^{-c})$$

direct length generalization!

Attention concentration governs length generalization

Simply transitive actions: unique action to focus on from context.



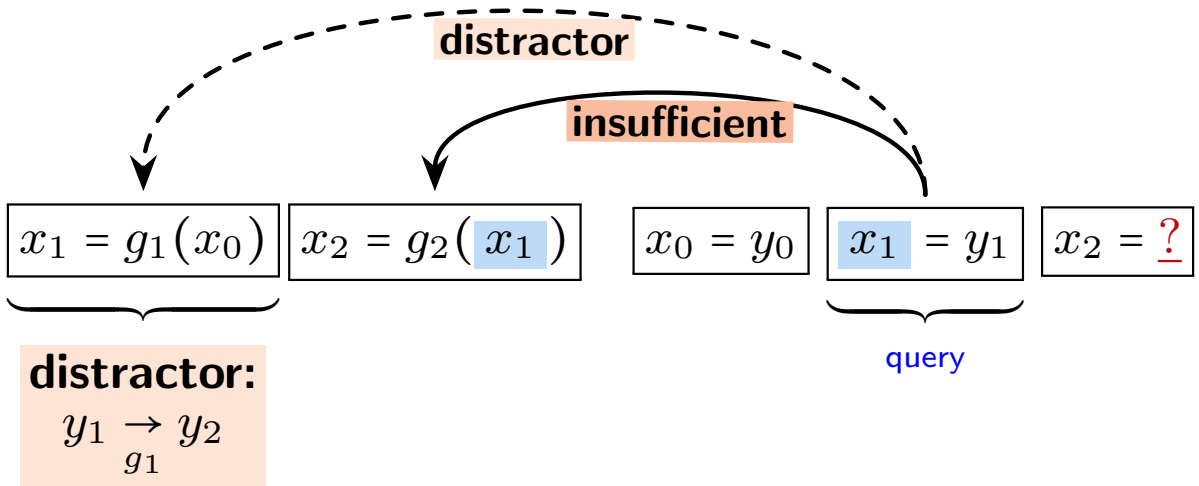
Sharp attention concentration at the end of training:

$$\mathbf{Attn}_{\text{correct}} \ge 1 - O(d^{-c})$$

direct length generalization!

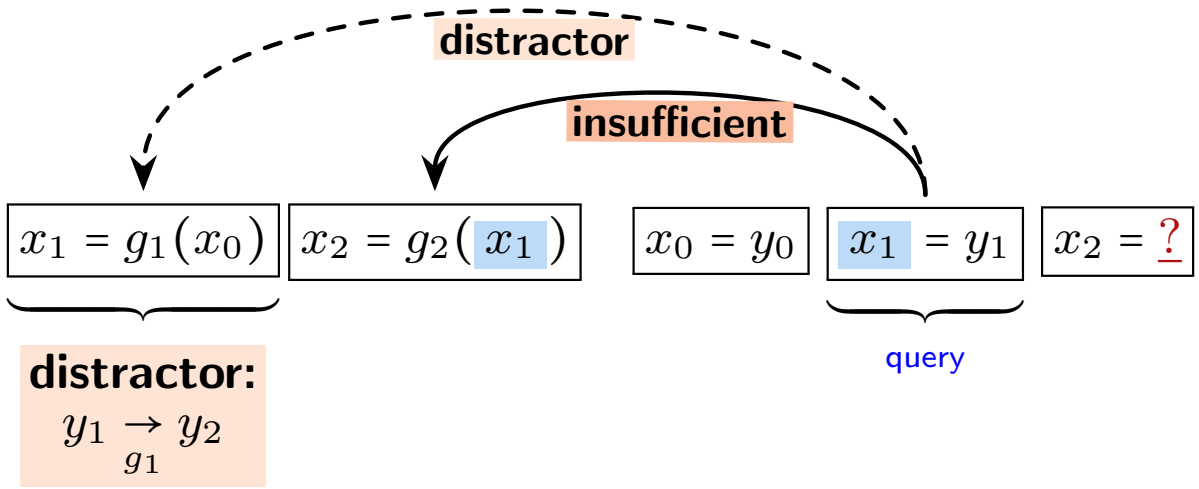
Attention concentration governs length generalization

Symmetry action: many distractors from context can make prediction



Attention concentration governs length generalization

Symmetry action: many distractors from context can make prediction

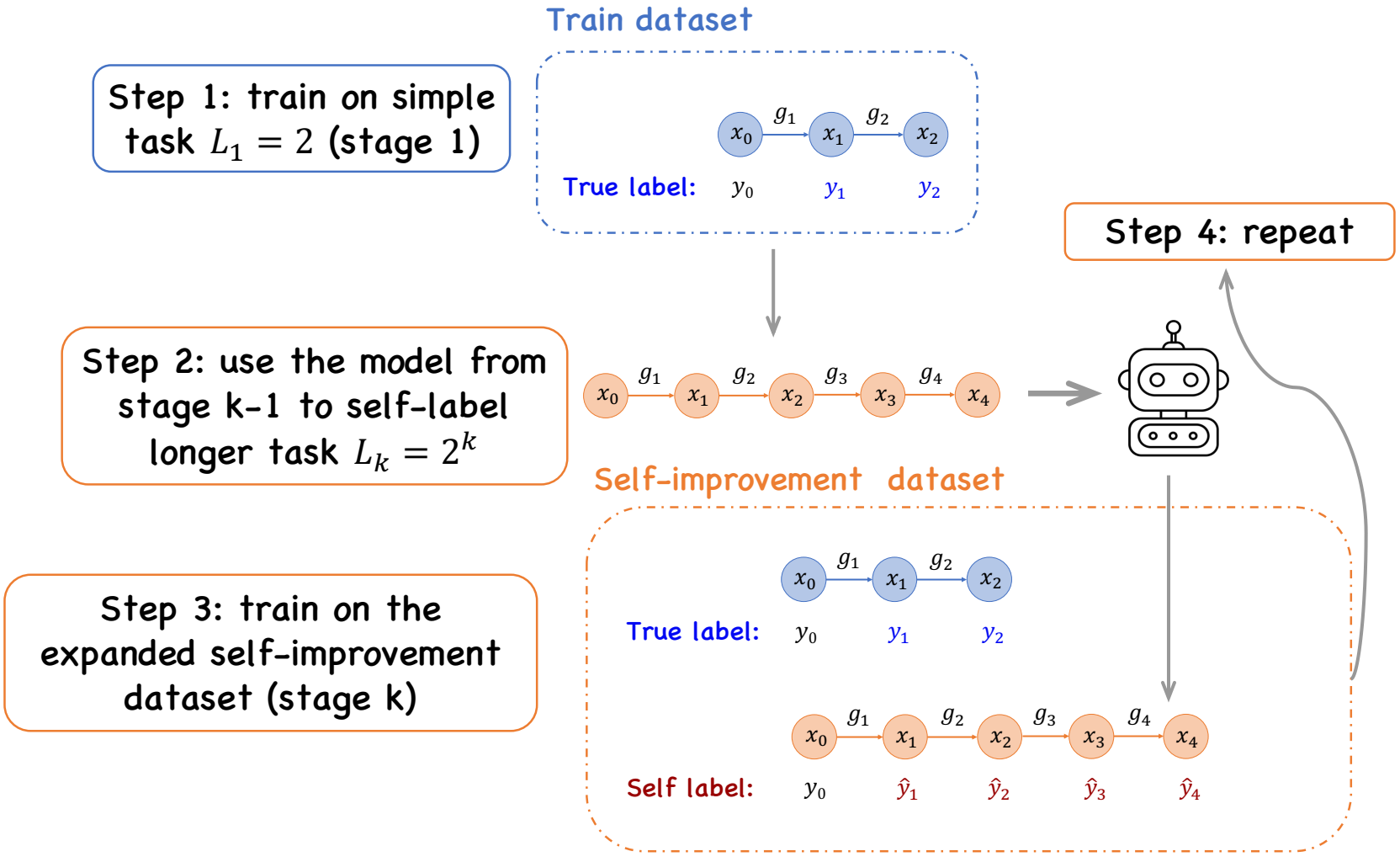


Limited attention concentration at the end of training:

$$\mathbf{Attn}_{\text{correct}} = 1 - \text{small constant}$$

no length generalization beyond constant length!

Self-improvement via curriculum training

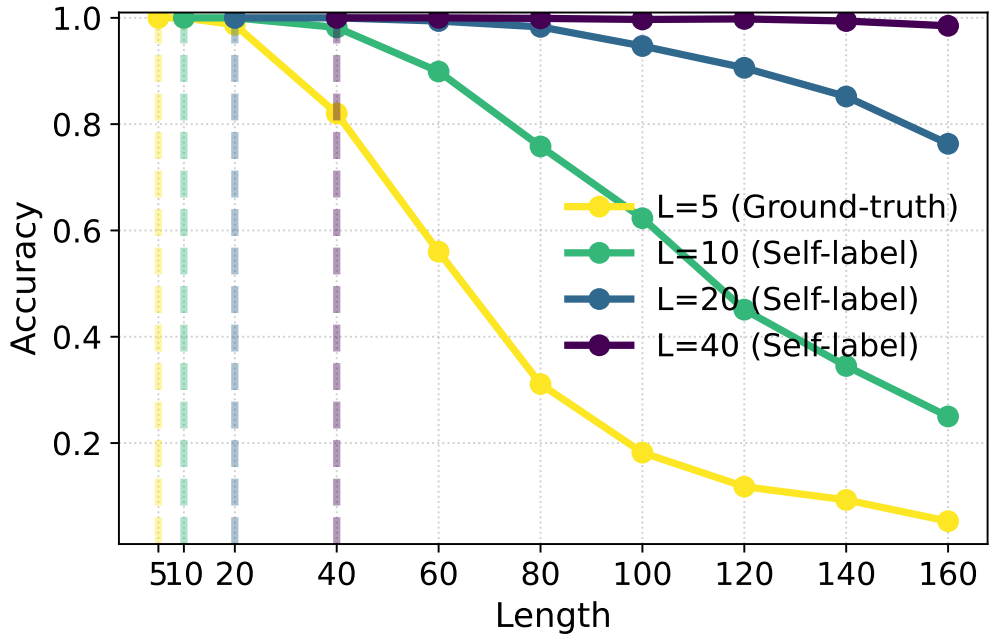


Inspired by (Lee et al., 2025)

Self-improvement provably helps

Theorem (Self-improvement for length generalization)

For symmetry actions, the one-layer transformer model trained via self-training after $\Theta(\log d)$ stages can solve tasks of $L \leq d$.



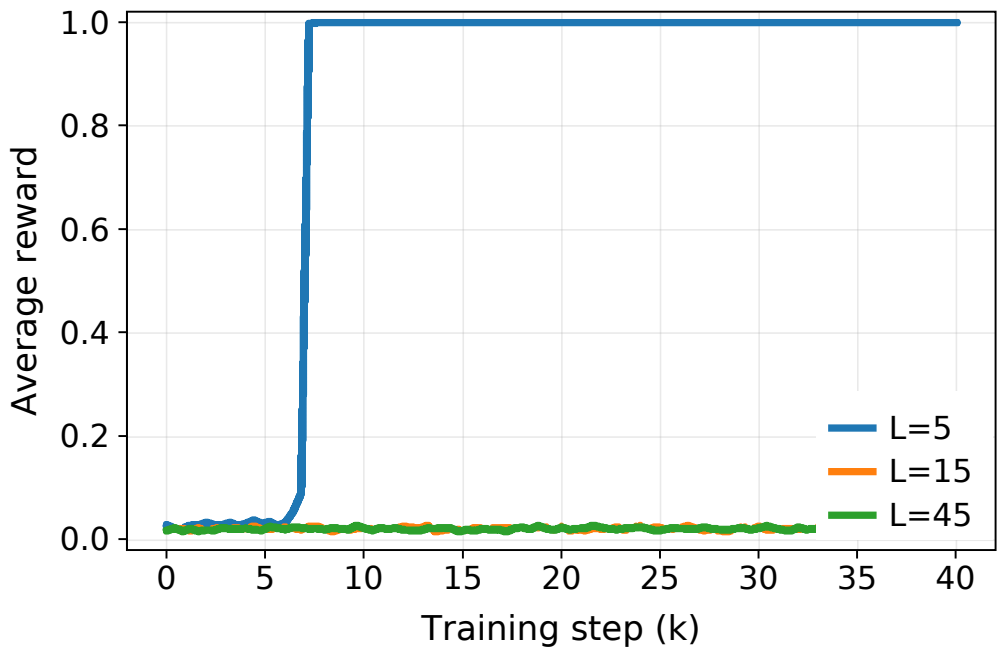
Self-improvement improves length generalization for harder tasks

Barrier to length generalization from RLVR

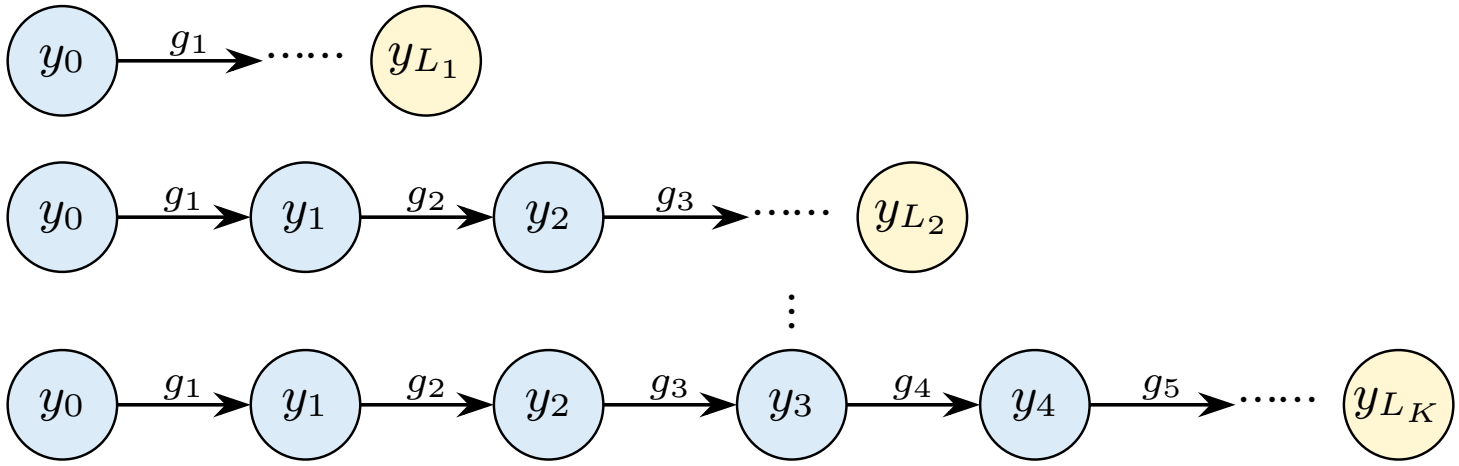
Theorem (Huang et al., 2026, informal)

Assume $\mathcal{G}$ is simply transitive, non-abelian and avoids shortcuts. There exists some constant length L_{critical} such that:

- *When $L < L_{\text{critical}}$, RL with policy gradient succeeds in a polynomial number of iterations $T = \widetilde{O}(\text{poly}(d))$.*
- *When $L \geq 2L_{\text{critical}}$, it suffers exponentially small gradient.*



Mixed-length RLVR: an implicit curriculum



Mixed-difficulty data recipe: we uniformly mix the horizons

$$\mathcal{L}_R = \{L_1, L_2, \ldots, L_K\},$$

where $R = L_{k+1}/L_k$ controls the *curriculum smoothness*.

Mixed-difficulty training objective:

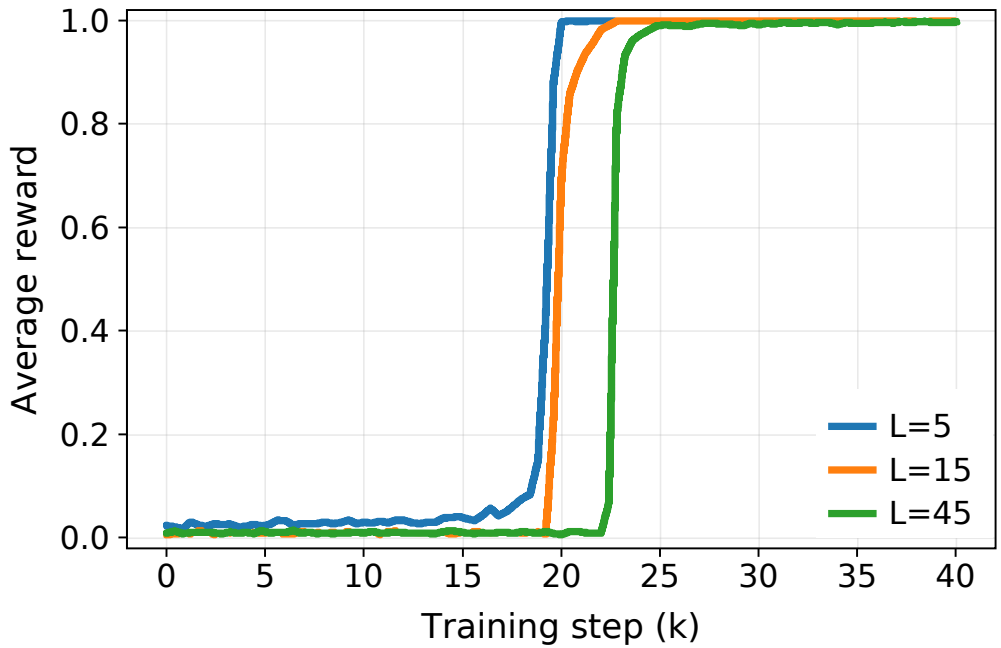
$$\mathcal{J}_{\text{mix},R}(\theta) = \mathbb{E}_{L\sim\text{Unif}(\mathcal{L}_R)}[\mathcal{J}_L(\theta)]$$

This creates an implicit curriculum as RL will first learn on shorter chains.

Breaking the barrier via mixed-length RLVR

Theorem (Huang et al., 2026)

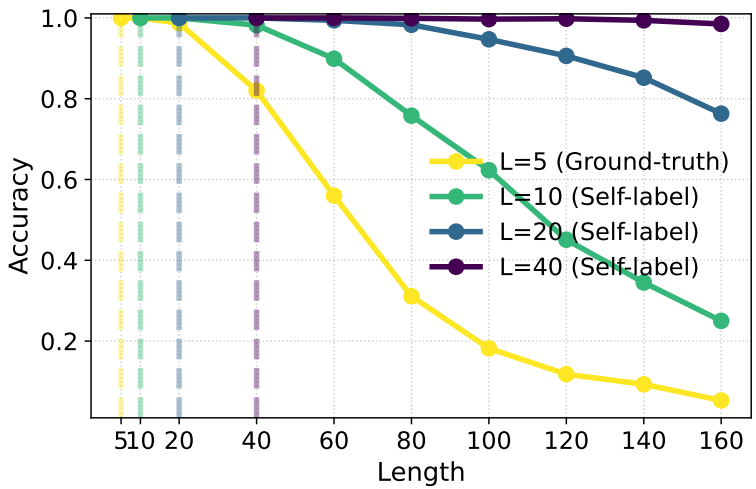
Assume $\mathcal{G}$ is simply transitive, non-abelian and avoids shortcuts. Suppose $L_1 \leq L_{\text{critical}}$, then mixed-length RL training succeeds in polynomial time for all lengths as long as $R \leq O(1)$.



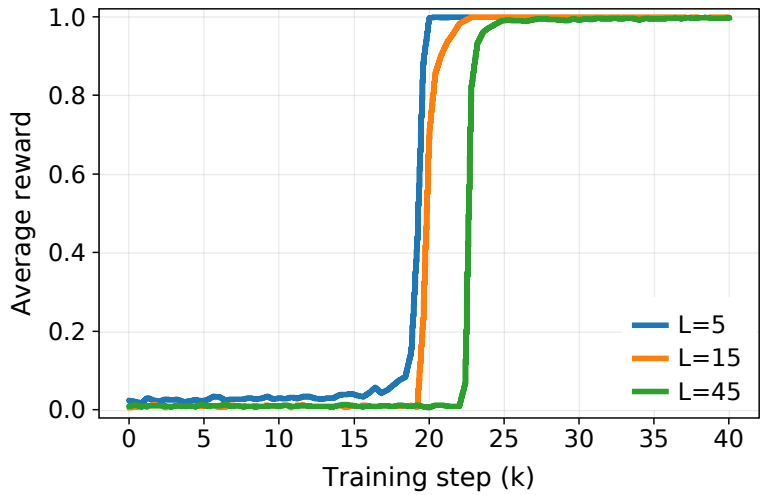
Smooth curricula facilitate relay dynamics across horizons.

Summary: what we learned from state tracking

Curriculum improves **attention concentration**
for predictable length generalization.



Enables harder tasks in SFT



Overcomes weaker feedback in RL

Mathematics of diffusion models: Provably efficient diffusion-based posterior sampling



Ivan Lee

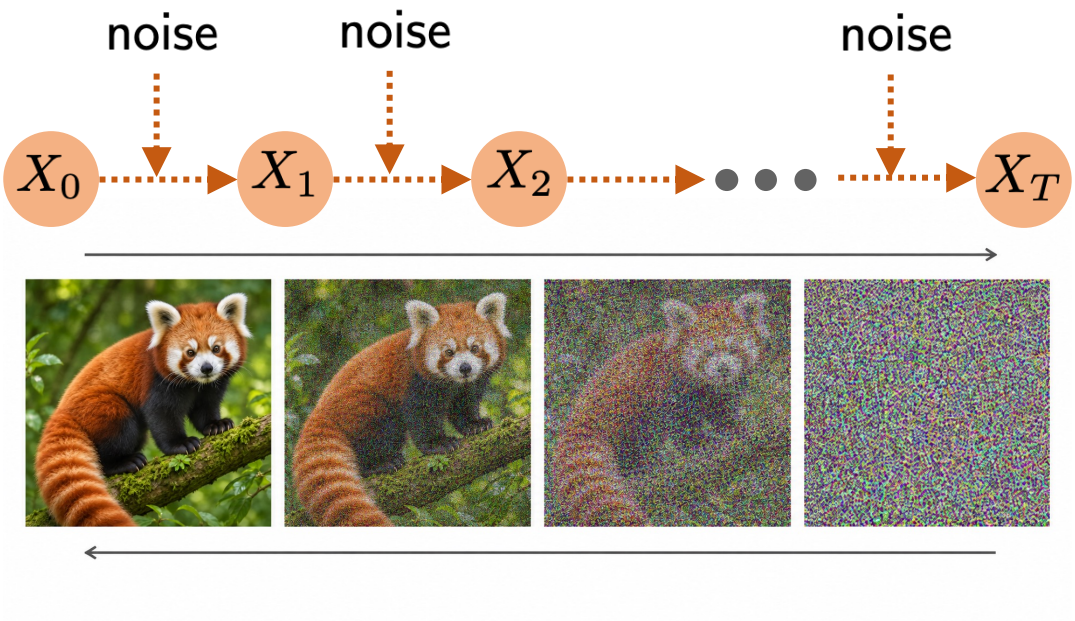
Yale



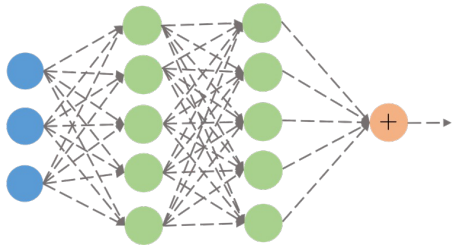
Xingyu Xu

CMU

Score-based diffusion models

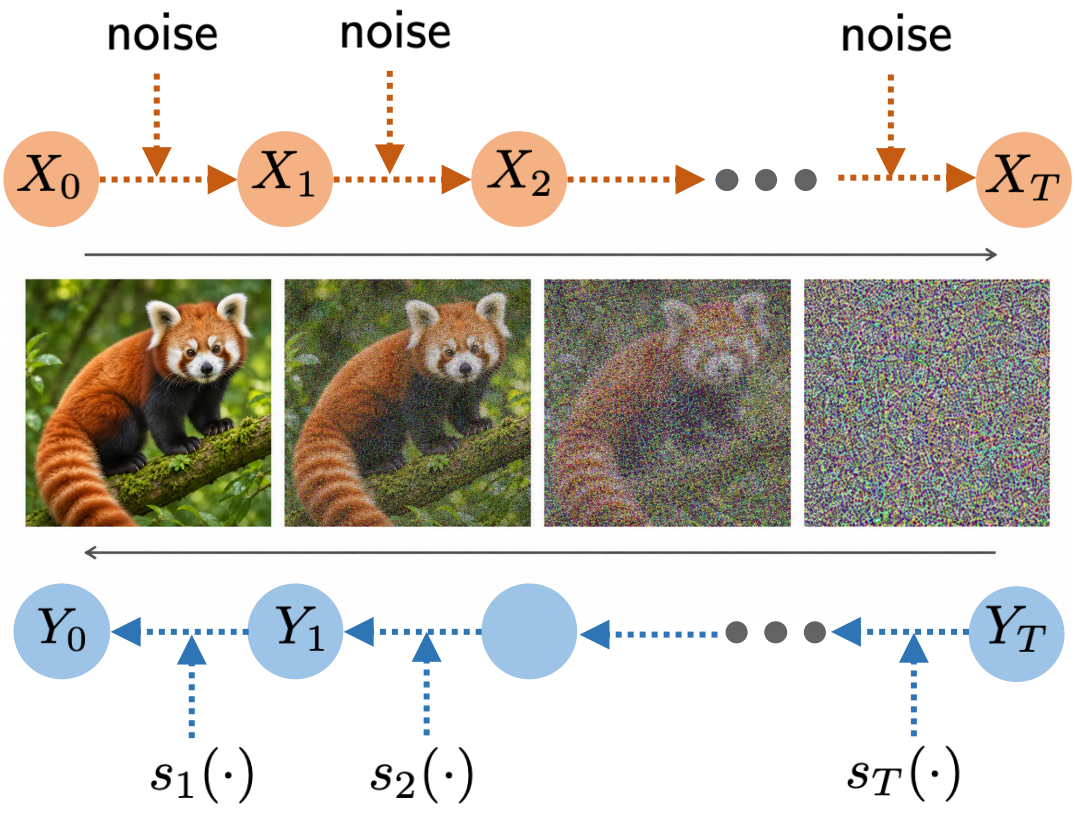


learn $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$

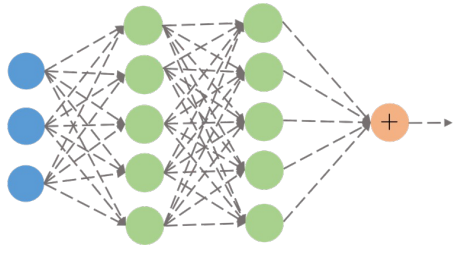


1. **score learning:** learn from the forward process $s_t = \nabla \log p_{X_t}(\cdot)$

Score-based diffusion models

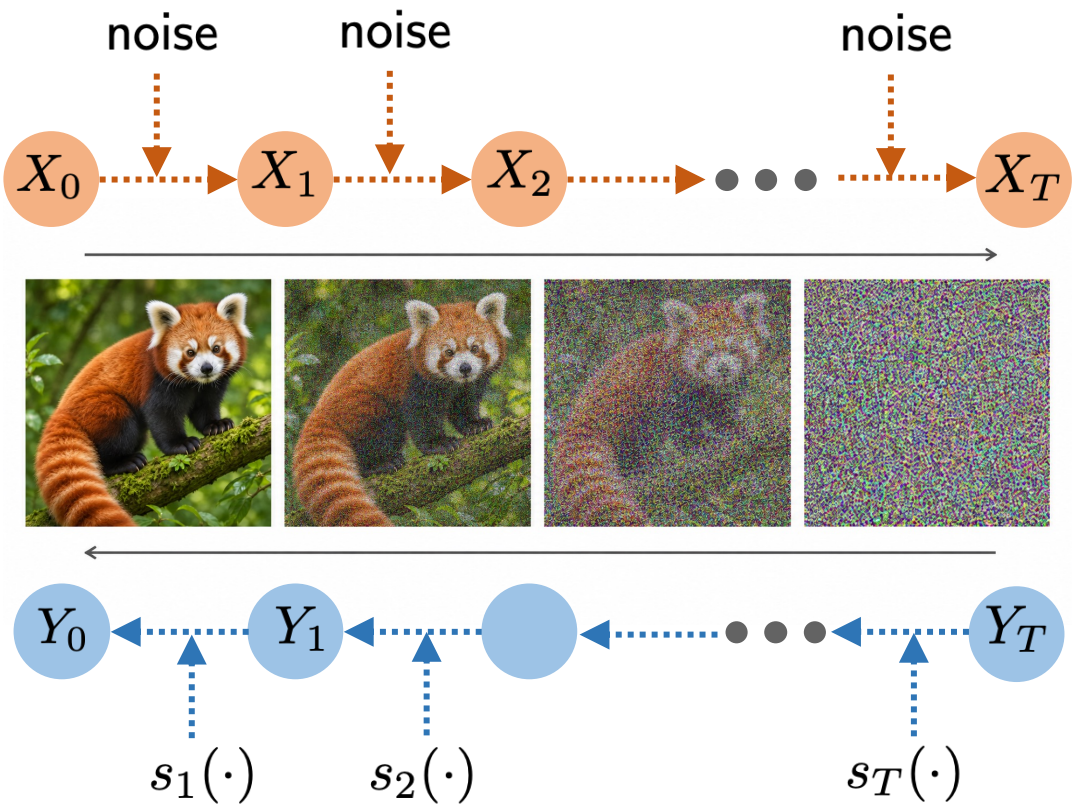


learn $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$

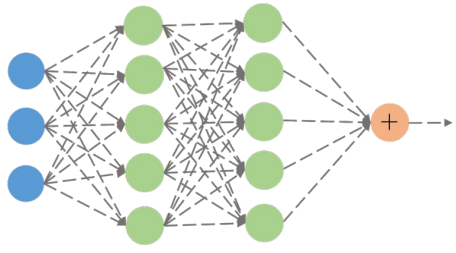


1. **score learning:** learn from the forward process $s_t = \nabla \log p_{X_t}(\cdot)$
2. **data generation:** sampling w/ the aid of score estimates $\{s_t(\cdot)\}$

Score-based diffusion models



learn $s_t(\cdot) = \nabla \log p_{X_t}(\cdot)$

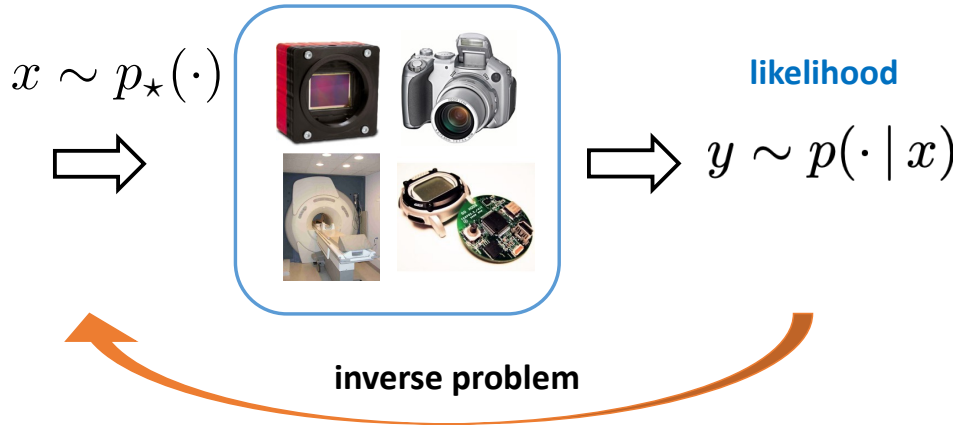


1. **score learning:** learn from the forward process $s_t = \nabla \log p_{X_t}(\cdot)$
2. **data generation:** sampling w/ the aid of score estimates $\{s_t(\cdot)\}$

Reliable scores are sufficient to generate from the data distribution.

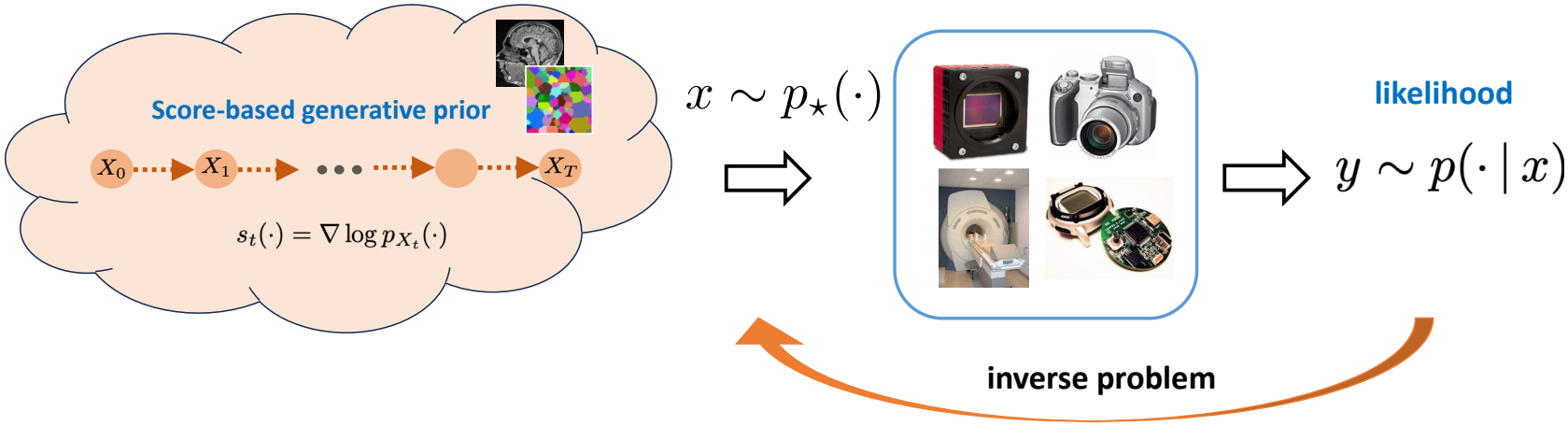
Posterior sampling for ill-posed inverse problems

Ill-posed inverse problem: recover the signal of interest x from y.



Posterior sampling for ill-posed inverse problems

Ill-posed inverse problem: recover the signal of interest x from y.

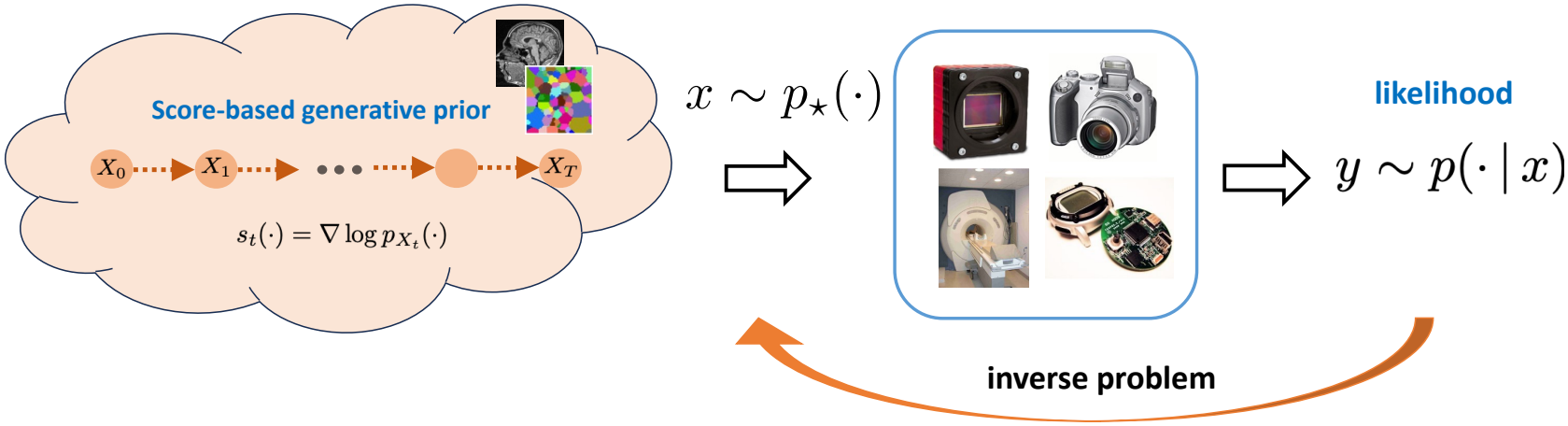


Posterior sampling: sample from

$$p(x|y) \propto \underbrace{p_\star(x)}_{\texttt{prior}} \cdot \underbrace{p(y \,|\, x)}_{\texttt{likelihood}}$$

Posterior sampling for ill-posed inverse problems

Ill-posed inverse problem: recover the signal of interest x from y.

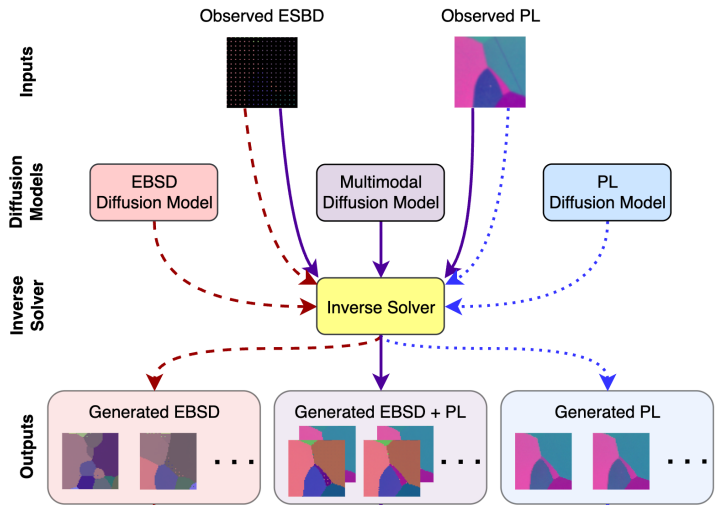


Posterior sampling: sample from

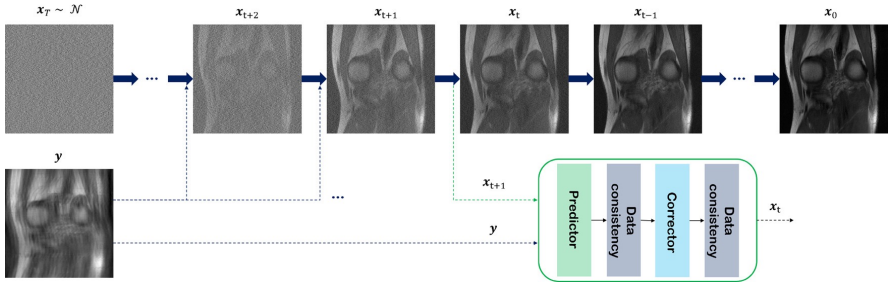
$$p(x|y) \propto \underbrace{p_\star(x)}_{\texttt{prior}} \cdot \underbrace{p(y \mid x)}_{\texttt{likelihood}}$$

Can we use diffusion models as an **implicit** prior for inverse problems?

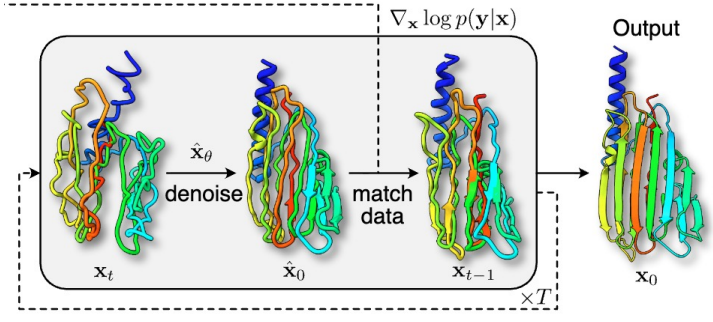
Diffusion-based inverse solvers are everywhere



Electron Back-Scattered Diffraction
[Dong et al, 2026]



Magnetic Resonance Imaging
[Chung and Ye, 2022]



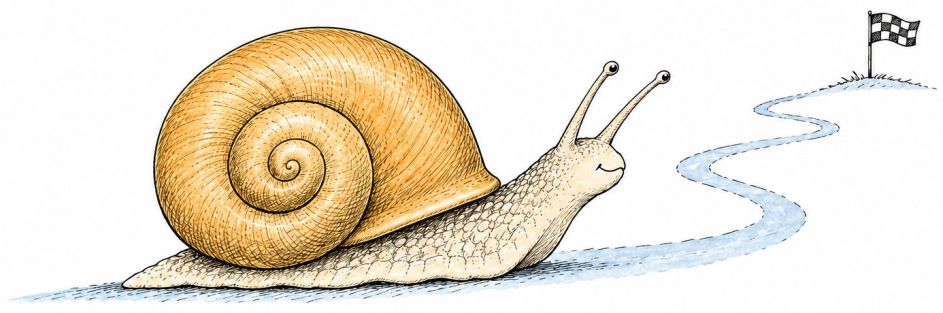
Protein Structures
[Levy et al, 2025]

Plug-and-play: reuse the same pretrained diffusion model for arbitrary forward models, without task-specific retraining.

What about its theory?

Posterior sampling can take exponential time (Gupta et. al., 2024; Bruna and Han, 2024): under mild *cryptography/hardness* assumption, even

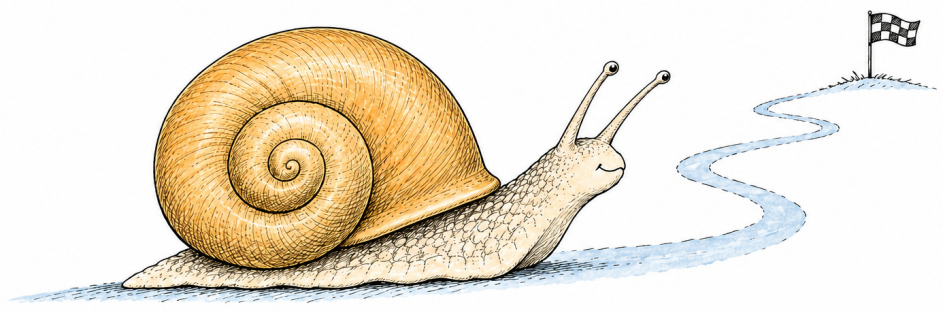
- for *linear* inverse problems,
- when *unconditional* sampling is provably fast.



What about its theory?

Posterior sampling can take exponential time (Gupta et. al., 2024; Bruna and Han, 2024): under mild *cryptography/hardness* assumption, even

- for *linear* inverse problems,
- when *unconditional* sampling is provably fast.



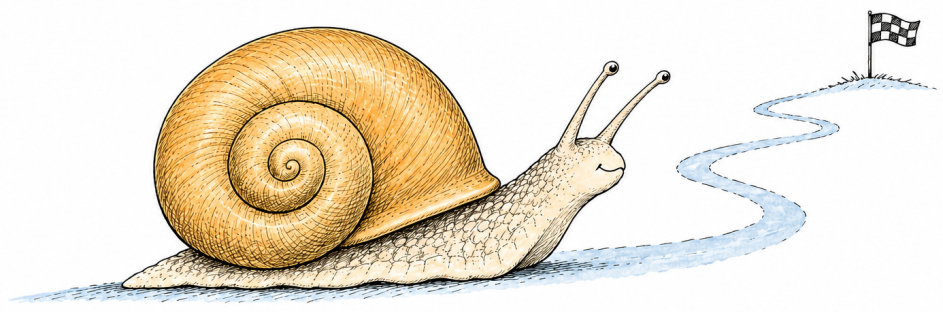
Existing guarantees lack efficiency. They

- either only admit asymptotic guarantees to posterior sampling,
- or contain unspecified parameters that can grow exponentially.

What about its theory?

Posterior sampling can take exponential time (Gupta et. al., 2024; Bruna and Han, 2024): under mild *cryptography/hardness* assumption, even

- for *linear* inverse problems,
- when *unconditional* sampling is provably fast.

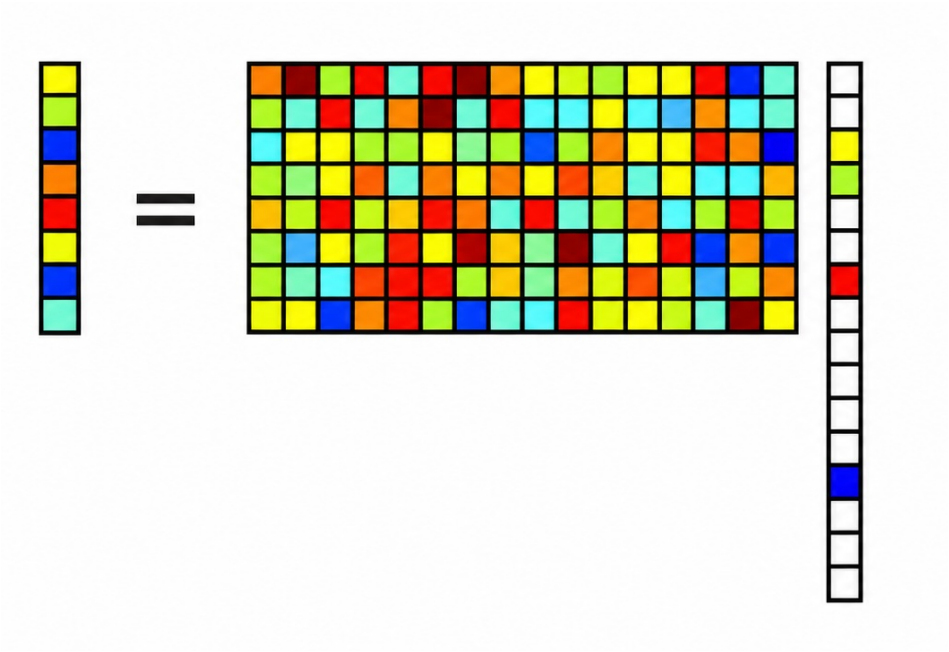


Existing guarantees lack efficiency. They

- either only admit asymptotic guarantees to posterior sampling,
- or contain unspecified parameters that can grow exponentially.

How do we reconcile the theory and practice gap?

Sandbox: sparse recovery



Compressed sensing (Candes, Tao, Romberg, 2004; Donoho, 2004):

$$y = AX_\star + \sigma\zeta, \qquad A \in \mathbb{R}^{m\times n}, \quad \zeta \sim \mathcal{N}(0, I_m).$$

- A beloved problem with great success in theory and in practice.
- Sweet spot of nontrivial prior and tractability.

Bernoulli–Gaussian prior for sparse recovery

Bernoulli–Gaussian prior: $X_\star \sim p_\star$, where $X_\star \sim \mathsf{BG}(k, n)$

$$(X_\star)_i \overset{\text{i.i.d.}}{\sim} \left(1 - \frac{k}{n}\right)\delta_0 + \frac{k}{n}\mathcal{N}(0, 1).$$

Here k, is the expected support size.

Bernoulli–Gaussian prior for sparse recovery

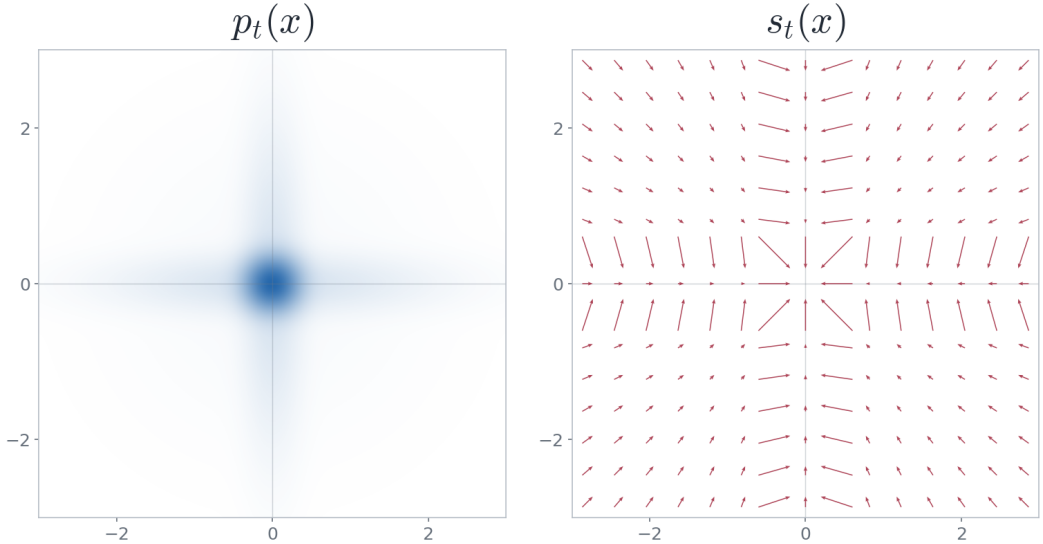
Bernoulli–Gaussian prior: $X_\star \sim p_\star$, where $X_\star \sim \mathrm{BG}(k, n)$

$$(X_\star)_i \overset{\text{i.i.d.}}{\sim} \left(1 - \frac{k}{n}\right)\delta_0 + \frac{k}{n}\mathcal{N}(0, 1).$$

Here k, is the expected support size.

Access to the prior: estimated *unconditional* scores

$$\widehat{s}_t(x) \approx \nabla \log p_t(x), \qquad p_t = p_\star * \mathcal{N}(0, tI_n), \quad t > 0.$$

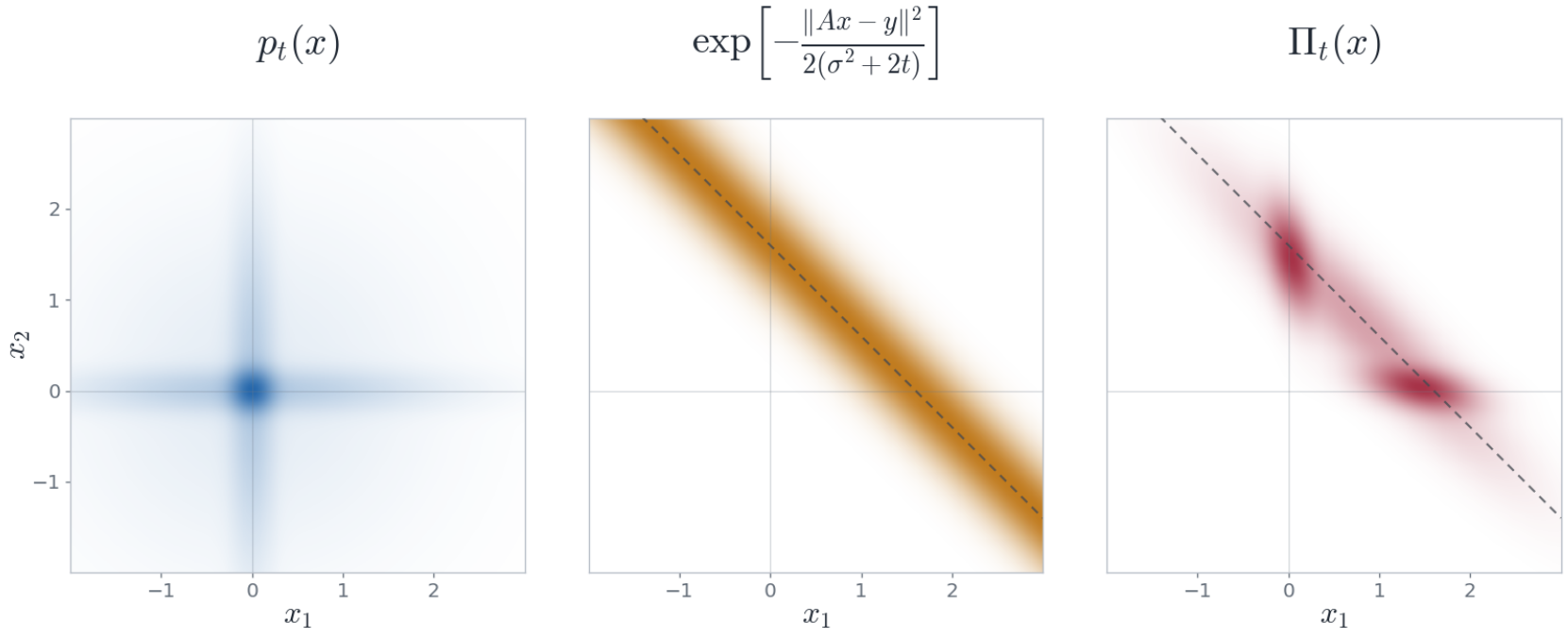


A tilted target towards the posterior

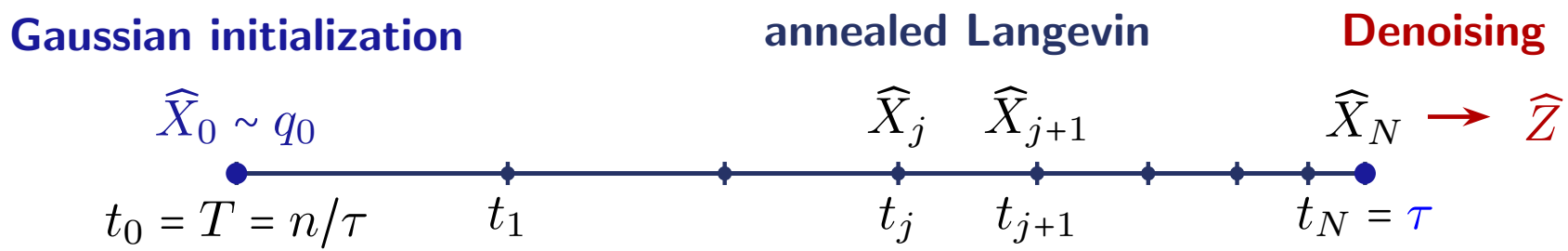
Tilted target: at each level t

$$\Pi_t(x) \propto p_t(x) \exp\left(-\frac{\|Ax - y\|^2}{2(\sigma^2 + 2t)}\right).$$

- Decreasing t sharpens the prior and strengthens guidance.
- When $t = 0$, this is the posterior $p_\star(x \mid y) \propto p_\star(x) \exp\left(-\frac{\|Ax-y\|^2}{2\sigma^2}\right)$.



Score-based diffusion inverse solver



$$q_0(x) \propto \exp\left(-\frac{\|x\|^2}{2T} - \frac{\|Ax-y\|^2}{2(\sigma^2+2T)}\right)$$

$$\widehat{Z} \sim \mathcal{L}(Z \mid Z + \sqrt{\tau}\,\xi = \widehat{X}_N)$$
$$Z \sim p_\star, \quad \xi \sim \mathcal{N}(0, I_n)$$

Likelihood-guided annealed Langevin sampler (Jalal et al, 2021): one Langevin step per level to track the evolving tilted target

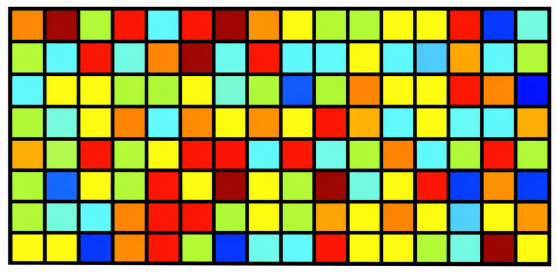
$$\widehat{X}_{j+1} = \widehat{X}_j + h\Bigg[\underbrace{\widehat{s}_{t_j}(\widehat{X}_j) + \frac{A^\top(y - A\widehat{X}_j)}{\sigma^2 + 2t_j}}_{\text{score of the tilted target}}\Bigg] + \underbrace{\sqrt{2h}\,\xi_j}_{\text{injected noise}}, \quad \xi_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_n),$$

where $h = N^{-2/3}$ is the step size, $t_j = T\left(\frac{\tau}{T}\right)^{j/N}$ is the schedule, $0 \le j < N$.

Assumptions

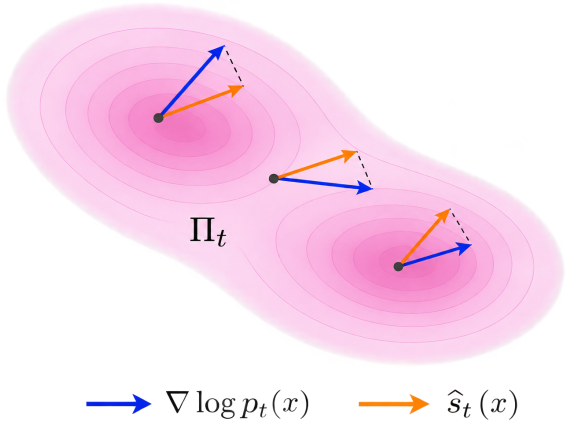
Sensing matrix. We cover a few families of popular random matrices:

- *Sub-Gaussian matrices.*
- *Randomly subsampled orthonormal bases*.



Score estimation error. The score error is accumulated along the measurement-dependent tilted targets:

$$\mathcal{E}_{\mathrm{sc}} := \mathbb{E}\left[\sum_{j=0}^{N-1} h \int \left\|\widehat{s}_{t_j}(x) - \nabla \log p_{t_j}(x)\right\|^2 \Pi_{t_j}(dx)\right].$$



Polynomial-time diffusion-based posterior sampling

Theorem (Lee, Xu, and Chi, 2026)

Fix $0 < \varepsilon < 1$. *Suppose that the sampler uses*

$$0 < \tau \lesssim \min\left\{\varepsilon^2, \frac{\varepsilon\sigma^2}{n^2(1 + k/\sigma^2)}\right\} \qquad \textit{(early stopping)},$$

and the sample size and iteration count satisfy

$$m \gtrsim \widetilde{\mathcal{O}}\big(k + \log(1/\varepsilon)\big), \quad N \gtrsim \widetilde{\mathcal{O}}\left(\frac{n^4}{m\varepsilon^6}\left(1 + \frac{n}{k\sigma\max\{1,\sigma\}}\right)^2\right).$$

Then the output $\widehat{Z}$ *satisfies*

$$\mathbb{E}\big[\mathsf{TV}\big(\mathcal{L}(\widehat{Z} \mid y), p_\star(\cdot \mid y)\big)\big] \lesssim \varepsilon + \sqrt{\mathcal{E}_{\mathrm{sc}}}.$$

- **Statistical and computational efficiency:** $\widetilde{\mathcal{O}}(k)$ samples and $\mathcal{O}(\mathsf{poly}(n))$ score evaluations suffice.

Polynomial-time diffusion-based posterior sampling

Theorem (Lee, Xu, and Chi, 2026)

Fix $0 < \varepsilon < 1$. *Suppose that the sampler uses*

$$0 < \tau \lesssim \min\left\{\varepsilon^2, \frac{\varepsilon\sigma^2}{n^2(1 + k/\sigma^2)}\right\} \qquad \textit{(early stopping)},$$

and the sample size and iteration count satisfy

$$m \gtrsim \widetilde{\mathcal{O}}\big(k + \log(1/\varepsilon)\big), \quad N \gtrsim \widetilde{\mathcal{O}}\left(\frac{n^4}{m\varepsilon^6}\left(1 + \frac{n}{k\sigma\max\{1,\sigma\}}\right)^2\right).$$

Then the output $\widehat{Z}$ *satisfies*

$$\mathbb{E}\big[\mathsf{TV}\big(\mathcal{L}(\widehat{Z} \mid y), p_\star(\cdot \mid y)\big)\big] \lesssim \varepsilon + \sqrt{\mathcal{E}_{\mathrm{sc}}}.$$

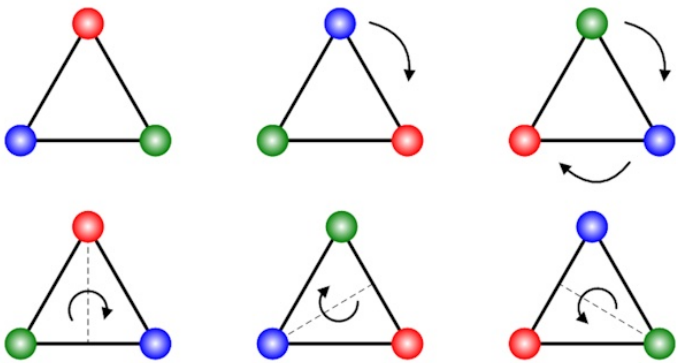
- **Statistical and computational efficiency:** $\widetilde{\mathcal{O}}(k)$ samples and $\mathcal{O}(\mathsf{poly}(n))$ score evaluations suffice.

Breaking the computational barrier in worst-case analysis!

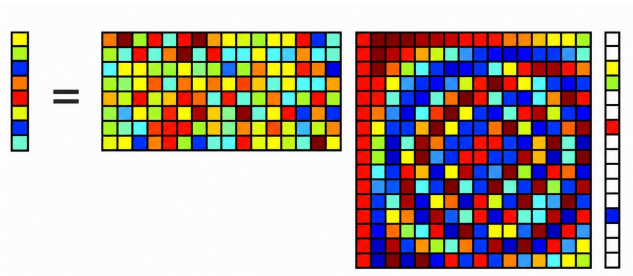
Summary

Studying structured problems to understand the dynamics of information flow to inform and predict practice.

reasoning models



diffusion models



state tracking
long-horizon reasoning
(Huang et al., 2026)

compressed sensing
posterior sampling
(Lee et al., 2026)

Thanks!

- Transformers Provably Learn Chain-of-Thought Reasoning with Length Generalization, arXiv:2511.07378.
- On the Emergence of Implicit Curriculum in RLVR Learning Dynamics, arXiv:2602.14872.
- Diffusion Models are Provably Efficient Priors for Sparse Recovery, arXiv:2611.24602.







https://yuejiechi.github.io