On the Learning Dynamics of RLVR at the Edge of Competence

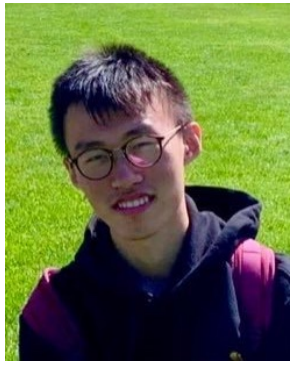
Yuejie Chi

Yale Statistics & Data Science

IMSI Workshop
Chicago, April 2026

Yu Huang*

Penn



Zixin Wen*

CMU

Yuting Wei

Penn

Aarti Singh

CMU

Yingbin Liang

OSU

Yuxin Chen

Penn

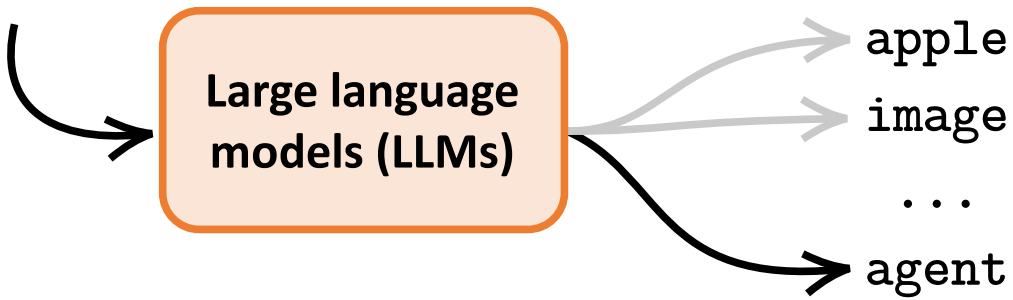
"On the Learning Dynamics of RLVR at the Edge of Competence," Y. Huang*, Z. Wen*, Y. Chi, Y. Wei, A. Singh, Y. Liang, Y. Chen, *arXiv: 2602.14872*

Large language models

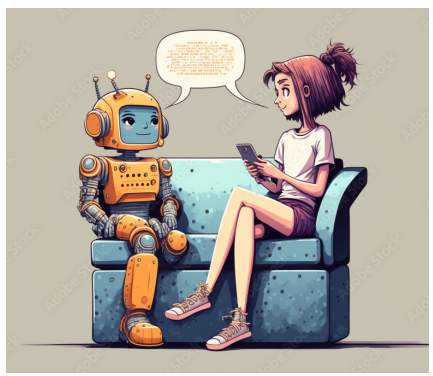
The language models predict the next word based on previous words.

Prompt: Explain reinforcement learning (RL).

Answer: Reinforcement learning (RL) is a type of machine learning where an...



Applications in translation, Q&A, summarization...

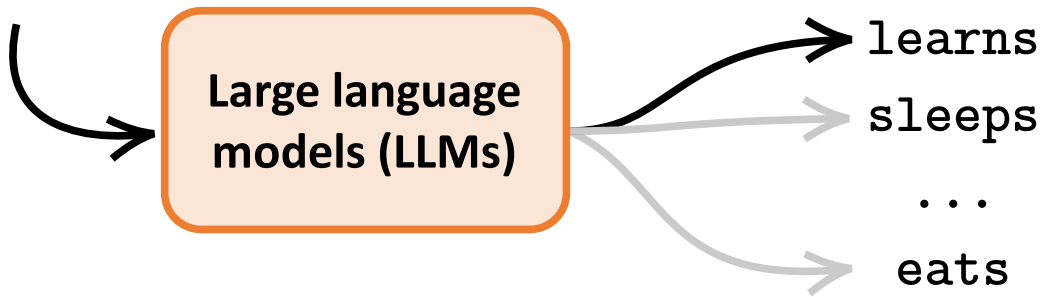


Large language models

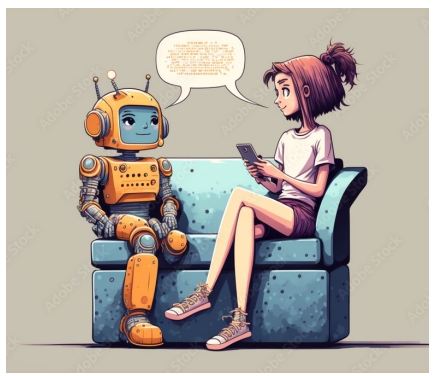
The language models predict the next word based on previous words.

Prompt: Explain reinforcement learning (RL).

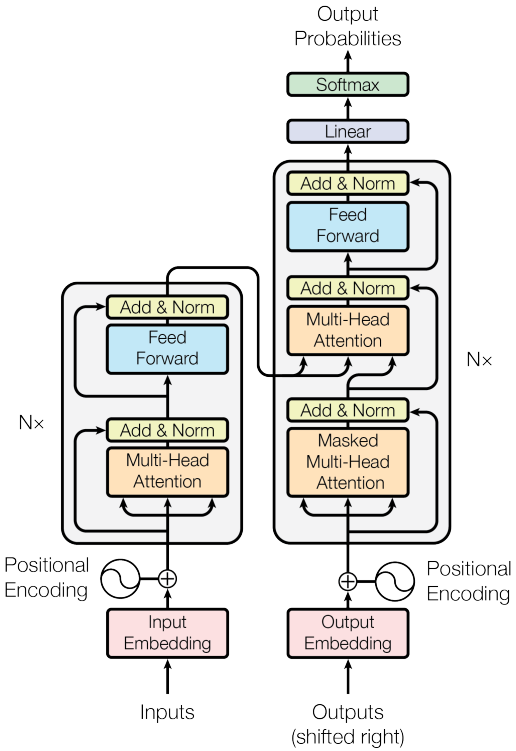
Answer: Reinforcement learning (RL) is a type of machine learning where an agent



Applications in translation, Q&A, summarization...



Transformers as the backbone architecture



Attention is all you need

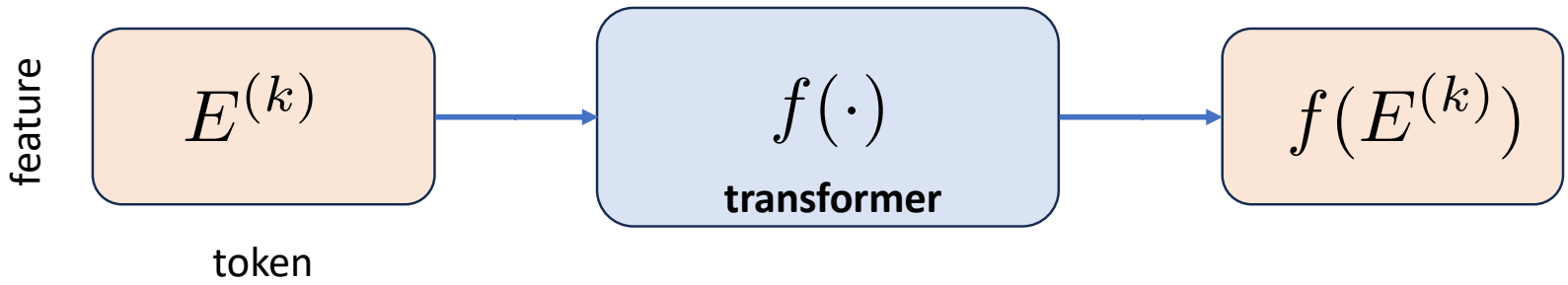
A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc

The dominant sequence transduction models are based on complex recurrent orconvolutional neural networks in an encoder and decoder configuration. The best …

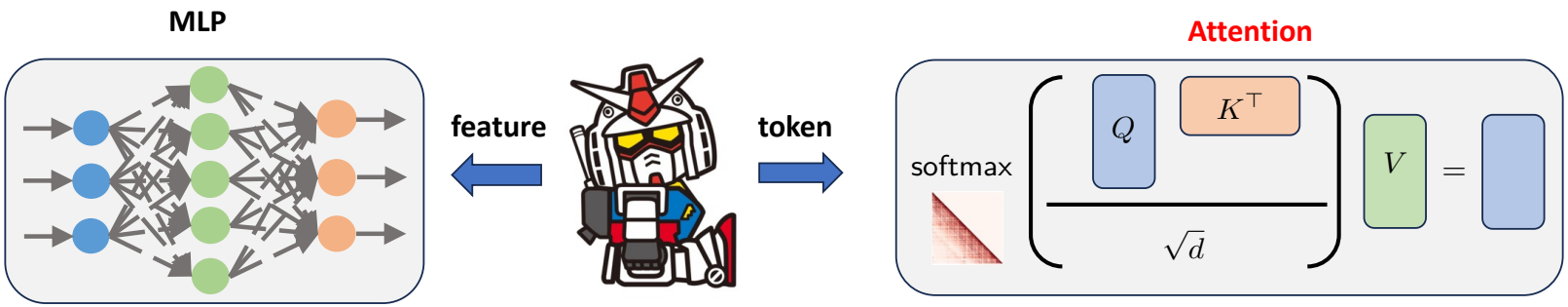
☆ Save 99 Cite Cited by 188409 Related articles »

Transformers as the backbone architecture

Sequence-to-sequence:

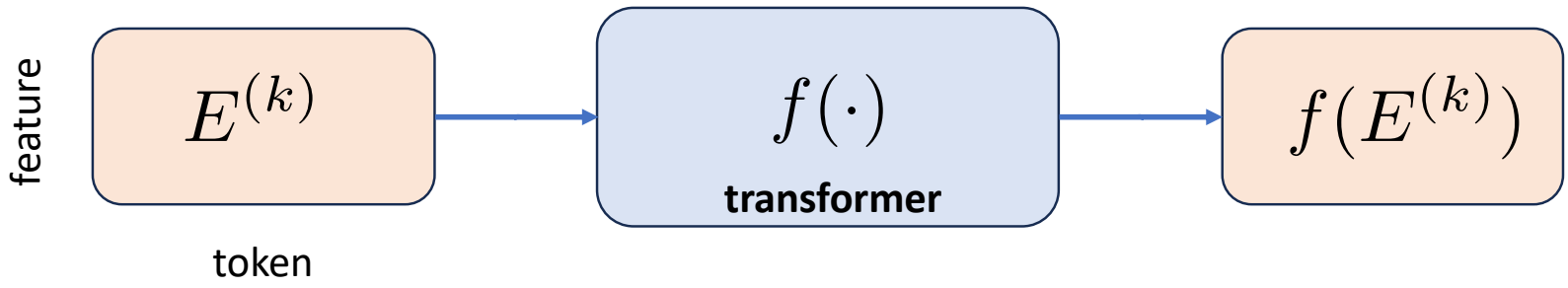


Transformer block = attention + MLP

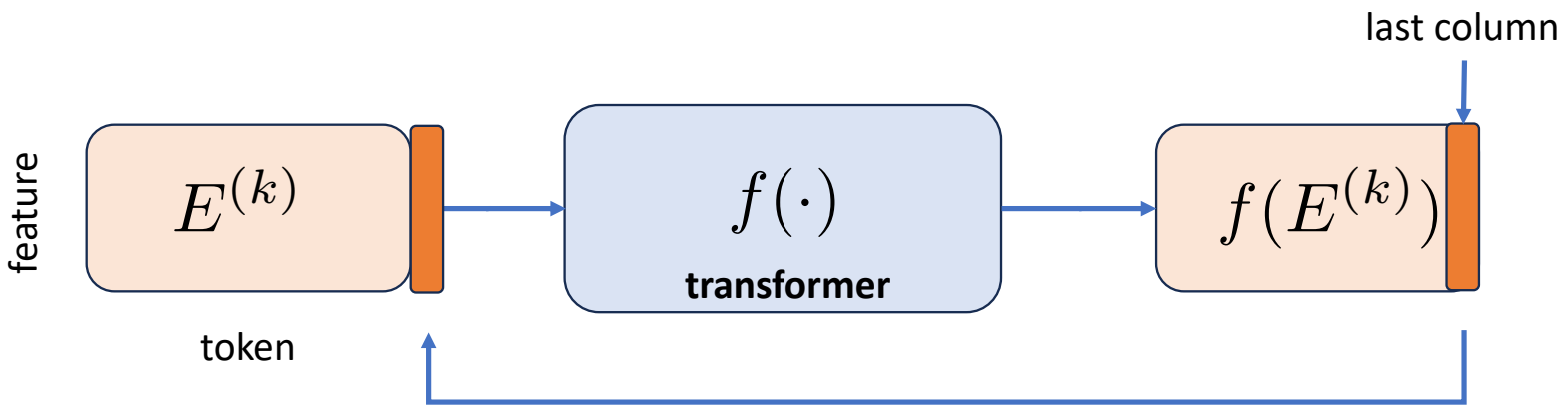


Transformers as the backbone architecture

Sequence-to-sequence:



Autoregressive decoding:



Chain-of-thought (CoT)

LLMs solve harder questions by reasoning **step-by-step**: generate intermediate reasoning steps before inferring an answer.*

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

<Question→Answer>

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

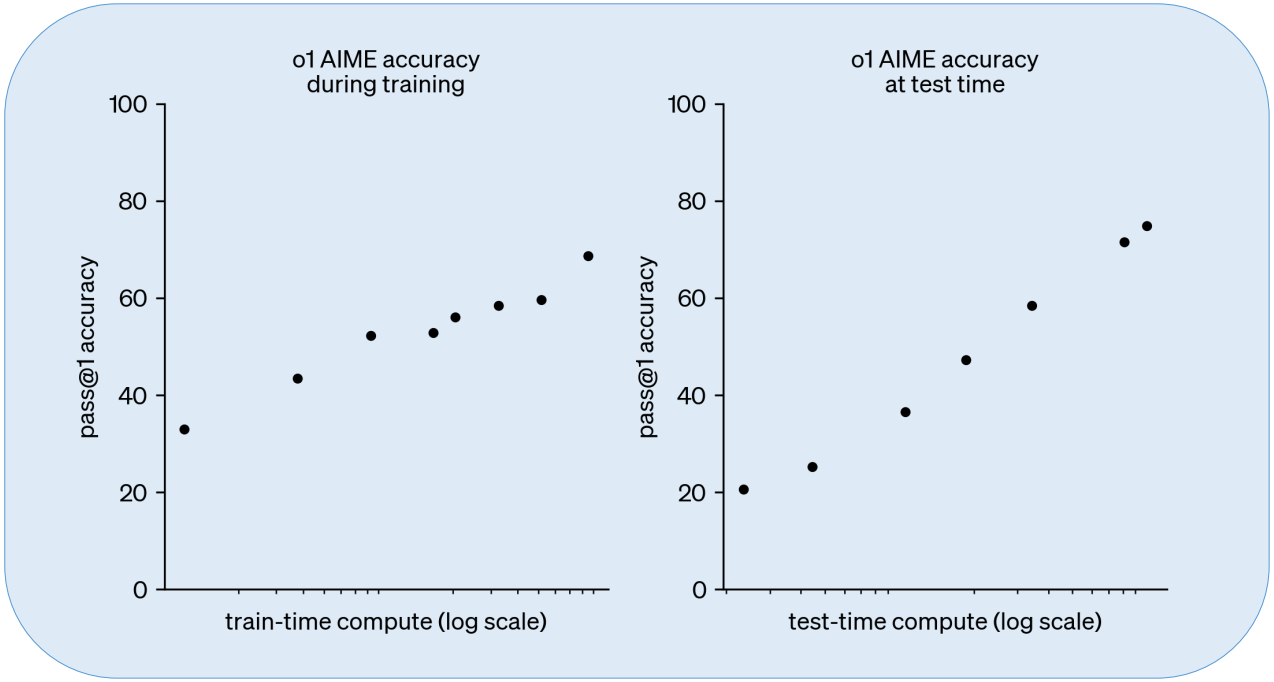
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

<Question→Rationale→Answer>

*Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.

The rise of reasoning models and test-time scaling

Reasoning models: OpenAI released o1 in 2024, which was trained via reinforcement learning.



(Figure credit: o1)

Test-time scaling: reasoning performance improves with more test-time compute (e.g., thinking longer).

RL with verifiable rewards

RL with verifiable rewards (RLVR): In early 2025, Deepseek released R1, and revealed the training recipe called RLVR.

- The model is only rewarded for accuracy and format.
- No outcome or process neural reward model.

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\cdots$

$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2$.

Rearrange to isolate the inner square root term:

$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be $\cdots$

We started with the equation:

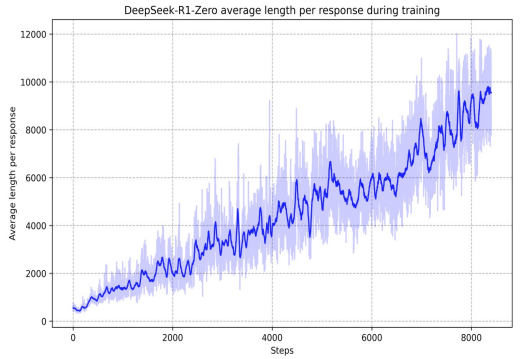
$\sqrt{a - \sqrt{a + x}} = x$

First, let's square both sides:

$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$

Next, I could square both sides again, treating the equation: $\cdots$

$\cdots$



Does RLVR learn new capabilities?

"Reasoning patterns of larger models can be distilled into smaller models, resulting in **better performance** *compared to the reasoning patterns* **discovered through RL** *on small models."*

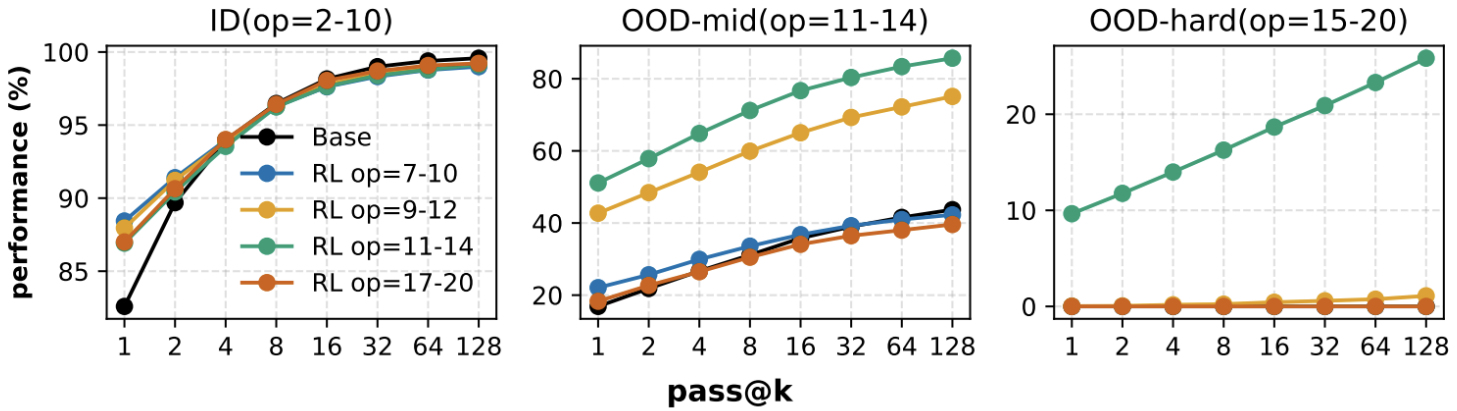
— DeepSeek-R1 [2025]

Model	AIME 2024		MATH	GPQA Diamond	LiveCode Bench
	pass@1	cons@64	pass@1	pass@1	pass@1
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9
Qwen2.5-32B-Zero	47.0	60.0	91.6	55.0	40.2
DeepSeek-R1-Distill-Qwen-32B	**72.6**	**83.3**	**94.3**	**62.1**	**57.2**

(Figure credit: Deepseek-R1)

The ceiling of RLVR seems to depend on the **capability** of base LLMs.

When is RL most effective?



RL is most effective at the **edge of competence** [Yue et al. 2025]:

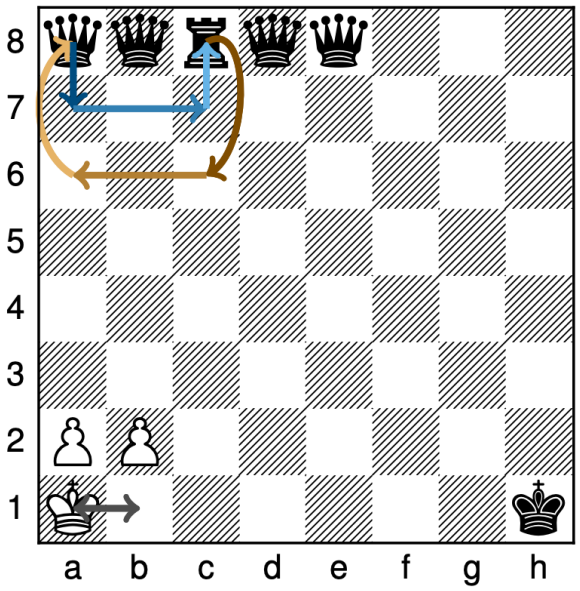
- No saturation: the task is not heavily covered during pre-training;
- "Just-right" difficulty: neither too easy nor too hard.

Our question: How does RLVR improve model's capabilities at the edge of competence?

State tracking

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

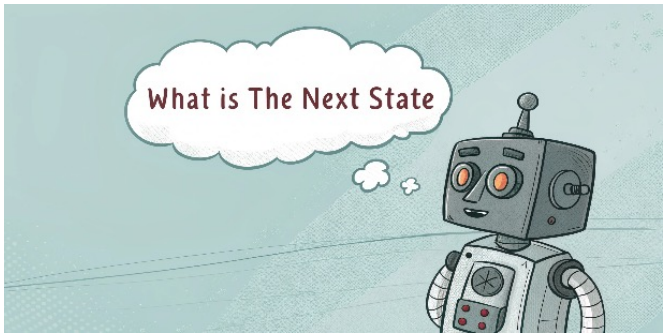
Playing Chess



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 as compositional reasoning

Given a group $\mathcal{G}$ acting on a finite value set $\mathcal{Y}$:

$$y_0 \xrightarrow{g_1} y_1 \xrightarrow{g_2} y_2 \xrightarrow{g_3} \cdots \xrightarrow{g_L} y_L.$$

Goal: calculate the last value $y_L = g_L(g_{L-1}(\cdots g_1(y_0)))$ given the sequence of transitions $g_i \in \mathcal{G}$ with an initial value $y_0 \in \mathcal{Y}$.

State tracking as compositional reasoning

Given a group $\mathcal{G}$ acting on a finite value set $\mathcal{Y}$:

$$y_0 \xrightarrow{g_1} y_1 \xrightarrow{g_2} y_2 \xrightarrow{g_3} \cdots \xrightarrow{g_L} y_L.$$

Goal: calculate the last value $y_L = g_L(g_{L-1}(\cdots g_1(y_0)))$ given the sequence of transitions $g_i \in \mathcal{G}$ with an initial value $y_0 \in \mathcal{Y}$.

Example: modulo sum

$$y_1 = y_0 \oplus_5 4, \qquad y_2 = y_1 \oplus_5 2, \qquad y_0 = 0$$

where

$$y_2 = y_1 \underbrace{\oplus_5 2}_{g_2} = (y_0 \underbrace{\oplus_5 4}_{g_1}) \oplus_5 2 = (\underbrace{0}_{y_0} \oplus_5 4) \oplus_5 2 \equiv 1 \pmod 5.$$

Reasoning data and autoregressive decoding

Reasoning data: for length-L reasoning, $Z^L = (Z_p^L, Z_a^L)$ is coded as

Prompt Z_p^L: $\boxed{x_{p,1}, g_1}, \boxed{x_{p,2}, g_2}, \ldots, \boxed{x_{p,L}, g_L}$

Answer Z_a^L: $\boxed{x_{a,0}, y_0}, \boxed{x_{a,1}, y_1}, \ldots, \boxed{x_{a,L}, y_L}$

where $x_{p,\ell}, x_{a,\ell} \in \mathcal{X}$ and $\mathcal{X} \subset \mathbb{R}^d$ is a set of orthonormal position vectors.

- *position alignment:* a fixed unknown permutation $\mathfrak{s} : \mathcal{X} \to \mathcal{X}$ s.t.,

$$x_{a,\ell-1} = \mathfrak{s}(x_{p,\ell}),$$

  which encodes the order of the relevant actions in action.
- *orthonormal embeddings:* $g \in \mathcal{G}$ and $y \in \mathcal{Y}$ are orthonormal vectors.

How does the transformer reason $\{y_\ell\}_{\ell=1}^L$ sequentially?

How does the transformer reason sequentially?

Model mechanism: TF = MLP (executor) ∘ Attention (retrieval)

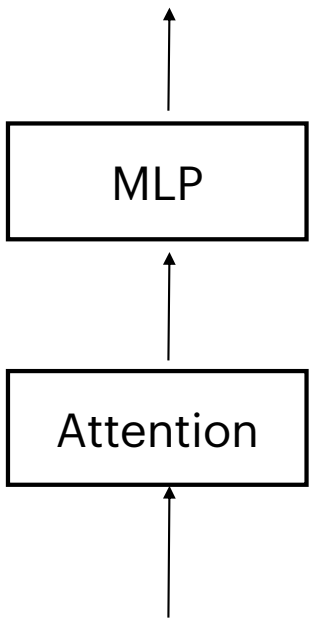
Transformer as autoregressive policy: given $Z^{L,0} = [Z_{a,0}, Z_p^L]$,

$$\pi_\theta(\hat{y}^\ell \mid Z^{L,0}) = \prod_{k=1}^{\ell} \pi_\theta(\hat{y}_k \mid \hat{Z}_{a,k-1}, Z_p^L), \qquad \ell = 1, \ldots, L,$$

where $\hat{y}^\ell = (\hat{y}_1, \ldots, \hat{y}_\ell)$.

Recursion of L-step reasoning: at each step ℓ, recall the target $y_\ell = g_\ell(y_{\ell-1})$.

- **Attention** = retriever: find correct g_ℓ from prompt at step ℓ
- **MLP** = executor: $(g_\ell, y_{\ell-1}) \mapsto g_\ell(y_{\ell-1})$ with near-perfect accuracy



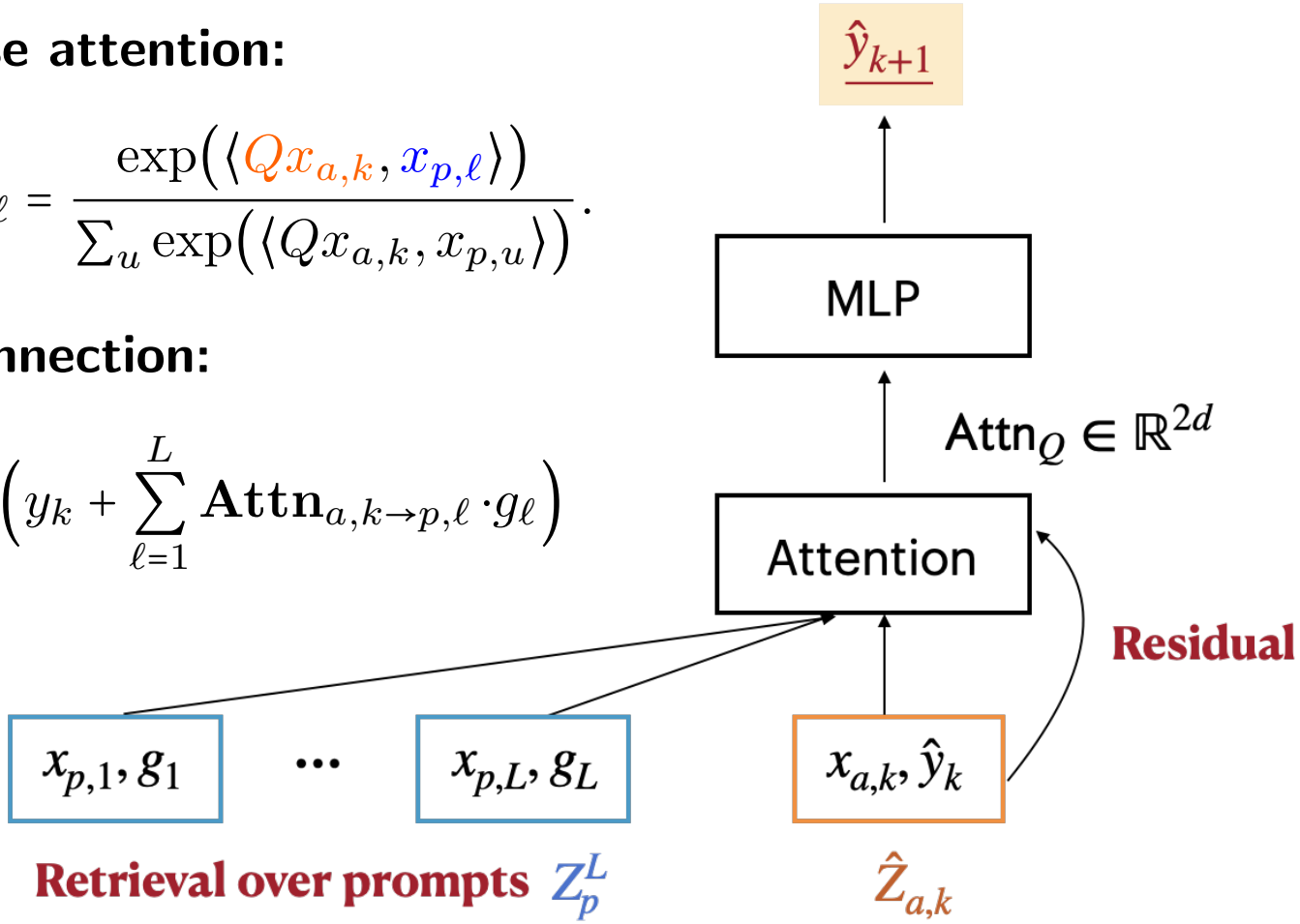
Attention layer

Position-wise attention:

$$\mathbf{Attn}_{a,k\to p,\ell} = \frac{\exp\big(\langle Q x_{a,k}, x_{p,\ell}\rangle\big)}{\sum_u \exp\big(\langle Q x_{a,k}, x_{p,u}\rangle\big)}.$$

Residual connection:

$$\mathrm{Attn}_Q = \frac{1}{2}\Big(y_k + \sum_{\ell=1}^{L} \mathbf{Attn}_{a,k\to p,\ell}\cdot g_\ell\Big)$$



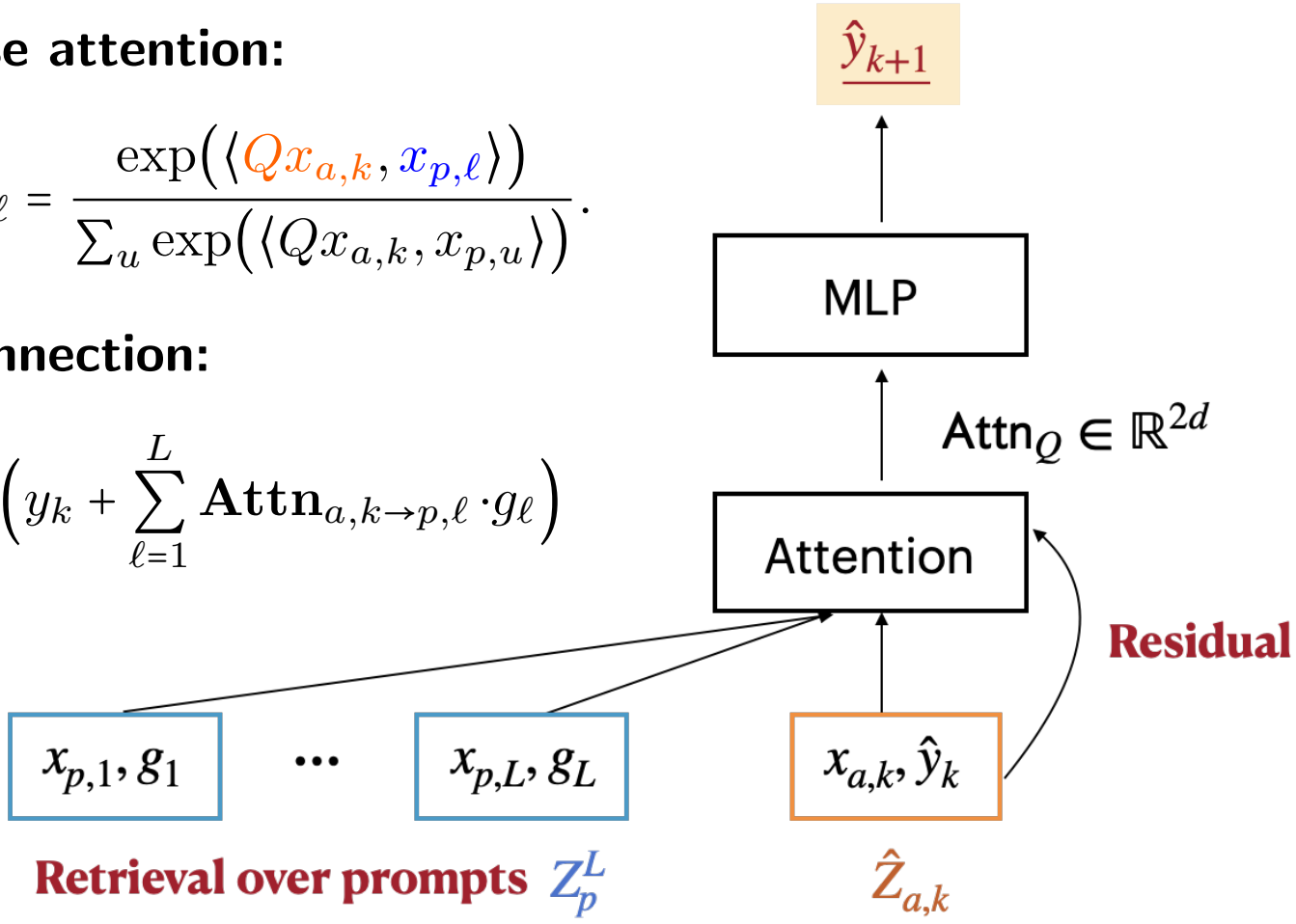
Attention layer

Position-wise attention:

$$\mathbf{Attn}_{a,k\to p,\ell} = \frac{\exp\big(\langle Qx_{a,k}, x_{p,\ell}\rangle\big)}{\sum_u \exp\big(\langle Qx_{a,k}, x_{p,u}\rangle\big)}.$$

Residual connection:

$$\mathrm{Attn}_Q = \frac{1}{2}\Big(y_k + \sum_{\ell=1}^{L} \mathbf{Attn}_{a,k\to p,\ell}\cdot g_\ell\Big)$$



When the attention **concentrates**, it outputs $\frac{1}{2}(y_k + g_{k+1})$.

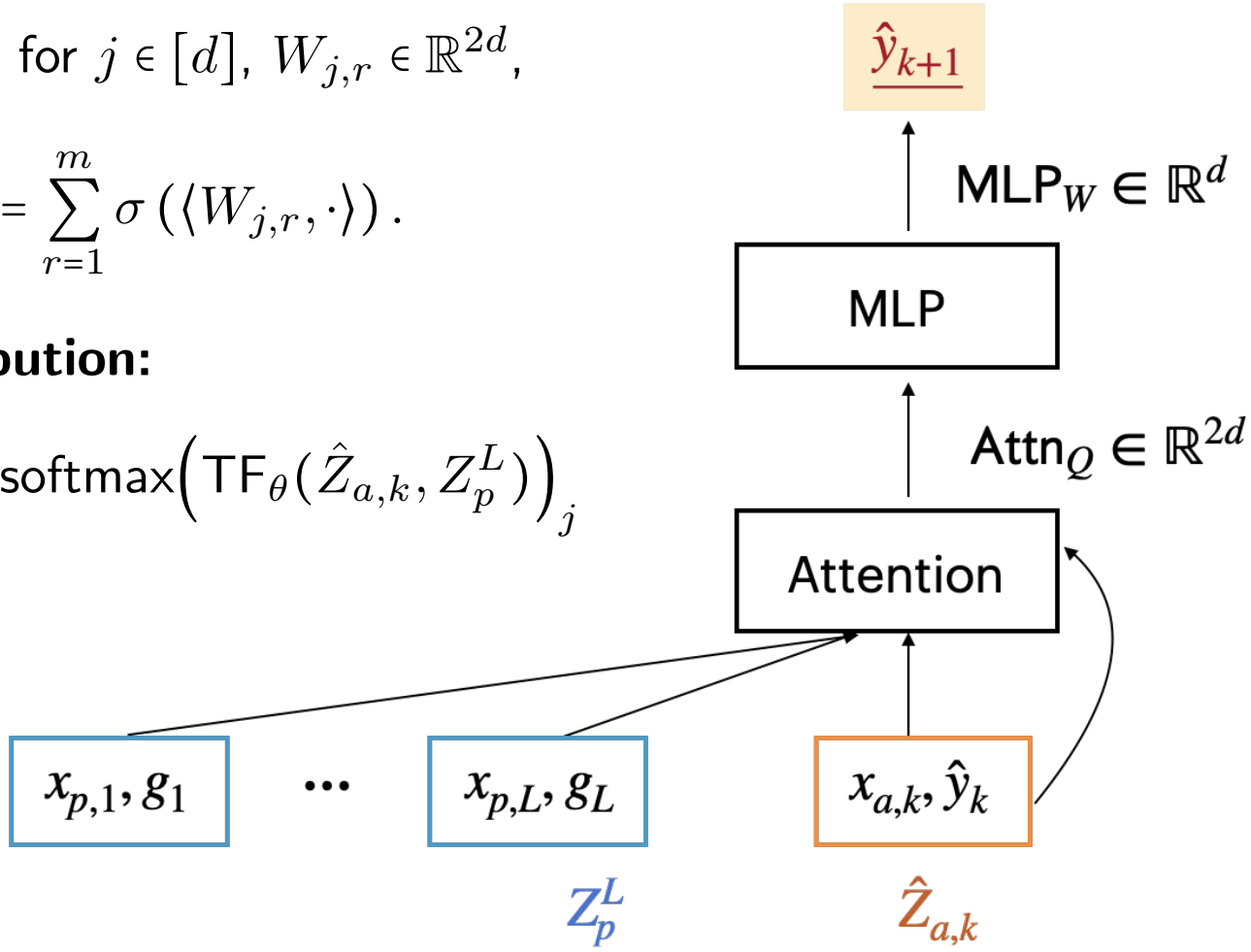
MLP layer

ReLU activation: for $j \in [d]$, $W_{j,r} \in \mathbb{R}^{2d}$,

$$\left[\mathsf{MLP}_W(\cdot)\right]_j = \sum_{r=1}^{m} \sigma\left(\langle W_{j,r}, \cdot \rangle\right).$$

Next-state distribution:

$$\pi_\theta\left(j \mid \hat{Z}_{a,k}, Z_p^L\right) \triangleq \mathsf{softmax}\left(\mathsf{TF}_\theta(\hat{Z}_{a,k}, Z_p^L)\right)_j$$



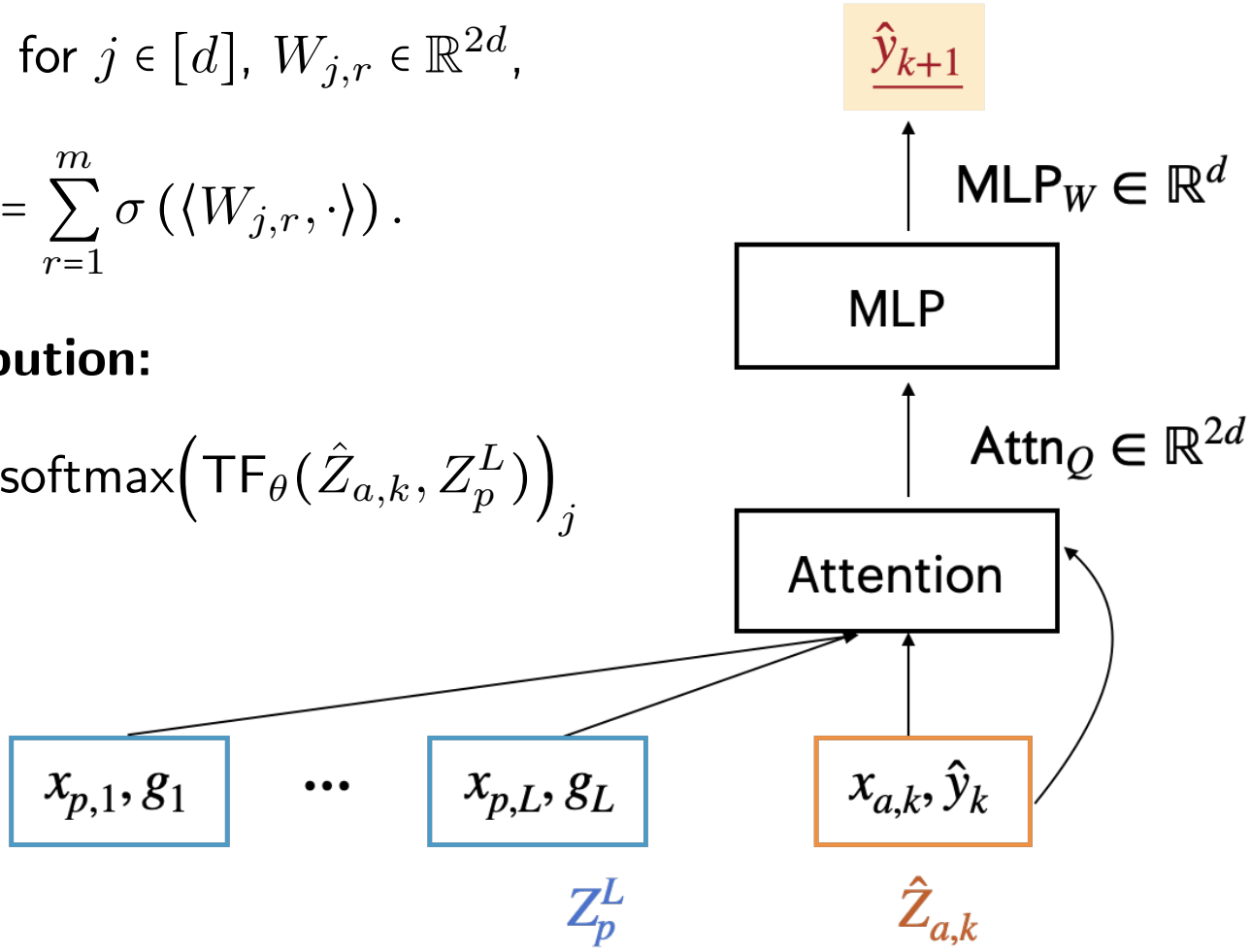
MLP layer

ReLU activation: for $j \in [d]$, $W_{j,r} \in \mathbb{R}^{2d}$,

$$\left[\mathsf{MLP}_W(\cdot)\right]_j = \sum_{r=1}^{m} \sigma\left(\langle W_{j,r}, \cdot \rangle\right).$$

Next-state distribution:

$$\pi_\theta\left(j \mid \hat{Z}_{a,k}, Z_p^L\right) \triangleq \mathsf{softmax}\left(\mathsf{TF}_\theta(\hat{Z}_{a,k}, Z_p^L)\right)_j$$



The MLP outputs highest probability on the index of $g_{k+1}(y_k)$.

Training distribution and assumptions

Data distribution $Z^L \sim \mathcal{D}^L$: we train on the population loss

- $y_0 \overset{\text{unif}}{\sim} \mathcal{Y}$;
- $g_\ell \overset{\text{unif, w/o rep}}{\sim} \mathcal{G}$;
- $x_{p,\ell} \overset{\text{unif, w/o rep}}{\sim} \mathcal{X}$;
- $x_{a,\ell-1} = \mathfrak{s}(x_{p,\ell})$ and $x_{a,L} \overset{\text{unif}}{\sim} \mathcal{X} \setminus \{\mathfrak{s}(x_{p,\ell})\}$.

Training distribution and assumptions

Data distribution $Z^L \sim \mathcal{D}^L$: we train on the population loss

- $y_0 \overset{\text{unif}}{\sim} \mathcal{Y}$;
- $g_\ell \overset{\text{unif, w/o rep}}{\sim} \mathcal{G}$;
- $x_{p,\ell} \overset{\text{unif, w/o rep}}{\sim} \mathcal{X}$;
- $x_{a,\ell-1} = \mathfrak{s}(x_{p,\ell})$ and $x_{a,L} \overset{\text{unif}}{\sim} \mathcal{X} \setminus \{\mathfrak{s}(x_{p,\ell})\}$.

Group structure and action:

- *non-abelian simple:* different sequences of operations rarely lead to the same state.
- *simply transitively:* for any $y, y' \in \mathcal{Y}$, there exists a unique $g \in \mathcal{G}$ s.t., $g(y) = y' \Rightarrow |\mathcal{G}| = |\mathcal{Y}| = d$.

Training distribution and assumptions

Data distribution $Z^L \sim \mathcal{D}^L$: we train on the population loss

- $y_0 \overset{\text{unif}}{\sim} \mathcal{Y}$;
- $g_\ell \overset{\text{unif, w/o rep}}{\sim} \mathcal{G}$;
- $x_{p,\ell} \overset{\text{unif, w/o rep}}{\sim} \mathcal{X}$;
- $x_{a,\ell-1} = \mathfrak{s}(x_{p,\ell})$ and $x_{a,L} \overset{\text{unif}}{\sim} \mathcal{X} \setminus \{\mathfrak{s}(x_{p,\ell})\}$.

Group structure and action:

- *non-abelian simple:* different sequences of operations rarely lead to the same state.
- *simply transitively:* for any $y, y' \in \mathcal{Y}$, there exists a unique $g \in \mathcal{G}$ s.t., $g(y) = y' \Rightarrow |\mathcal{G}| = |\mathcal{Y}| = d$.

Asymptotics: we assume $|\mathcal{X}| = \Theta(d^{c_x})$ for some $c_x \in (0.1, 1)$. The longest length of the chain is $L_{\max} = |\mathcal{X}| - 1$.

Outcome-based RL objective

Outcome-based RL objective: $r(\hat{y}^L \mid Z^{L,0}) \triangleq \mathbf{1}\{\hat{y}_L = y_L\}$

$$\mathcal{J}_L(\theta) = \mathbb{E}_{Z^L}\left[\mathbb{E}_{\hat{y}^L \sim \pi_\theta^L(\cdot|Z^{L,0})}\left[r(\hat{y}^L \mid Z^{L,0})\right]\right].$$

Outcome-based RL objective

Outcome-based RL objective: $r(\hat{y}^L \mid Z^{L,0}) \triangleq \mathbf{1}\{\hat{y}_L = y_L\}$

$$\mathcal{J}_L(\theta) = \mathbb{E}_{Z^L}\Big[\mathbb{E}_{\hat{y}^L \sim \pi_\theta^L(\cdot|Z^{L,0})}\big[r(\hat{y}^L \mid Z^{L,0})\big]\Big].$$

Pretrained MLP: we assume MLP has already learned the atomic skills of group action $(g, y) \mapsto g(y)$ with weight W fixed during RL training.

Outcome-based RL objective

Outcome-based RL objective: $r(\hat{y}^L \mid Z^{L,0}) \triangleq \mathbf{1}\{\hat{y}_L = y_L\}$

$$\mathcal{J}_L(\theta) = \mathbb{E}_{Z^L}\Big[\mathbb{E}_{\hat{y}^L \sim \pi_\theta^L(\cdot|Z^{L,0})}\big[r(\hat{y}^L \mid Z^{L,0})\big]\Big].$$

Pretrained MLP: we assume MLP has already learned the atomic skills of group action $(g, y) \mapsto g(y)$ with weight W fixed during RL training.

Policy gradient for attention: length-normalized policy gradient following REINFORCE:

$$Q^{(t+1)} = Q^{(t)} + \eta \nabla_Q \widetilde{\mathcal{J}}_L(\theta)$$
$$\nabla_Q \widetilde{\mathcal{J}}_L(\theta) = \frac{1}{L}\mathbb{E}_{Z^L, \hat{y}^L}\Big[r(\hat{y}^L \mid \hat{Z}^{L,0}) \nabla \log \pi_\theta(\hat{y}^L \mid \hat{Z}^{L,0})\Big],$$

where $\widetilde{\mathcal{J}}_L(\theta) = \frac{1}{L}\mathcal{J}_L(\theta)$, $Q^{(0)}$ is initialized as a zero matrix, and η is the learning rate.

Pretrained atomic skills

When MLP receives $\frac{1}{2}(g+y)$, then for $j=\tau(g(y))$

$$\text{softmax}\Big(\text{MLP}_W\Big(\frac{1}{2}(g+y)\Big)\Big)_j = 1-\frac{1}{\text{poly}(d)}$$

Here, $\tau:\mathcal{Y}\to[d]$ index the states for prediction.

Atomic structure: for each pair (g,y) with correct output $j=\tau(g(y))$, there exists a unique neuron r being activated:

$$\langle W_{j,r},g\rangle = B,\quad \langle W_{j,r},y\rangle = B,\quad \langle W_{j,r},s\rangle = -B \text{ (other } s\neq g,y)$$
$$\langle W_{j,r'},s\rangle = 0 \quad (\text{other } r'\neq r, s\in\mathcal{G}\cup\mathcal{Y})$$

where $B=C_B\log d$ and C_B is a constant.

- Prior work [Huang et al, 2025] provides guarantees for learning such atomic structures.

Horizon barrier of compositional reasoning

Our first result reveals that RLVR is efficient for short-horizon reasoning, but suffers from flat landscape beyond a critical horizon.

Theorem (Horizon Barrier)

Suppose the transformer is initialized from uniform attention, then:

- **Success of RL within short horizon** $L < C_B$**:** *for large enough iterations* $T = \widetilde{O}(\frac{\mathrm{poly}(d)}{\eta})$ *with* $\eta = \frac{1}{\mathrm{poly}(d)}$*, the reward*

$$\mathcal{J}_L^{(T)} \geq 1 - \frac{1}{\mathrm{poly}(d)}.$$

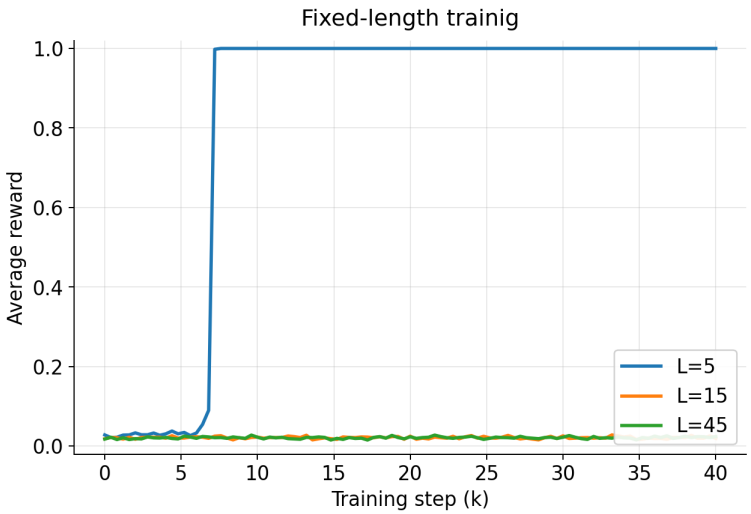
- **Exponentially flat region when** $L \geq 2C_B$**:** *the gradients of* $\mathcal{J}_L$ *and rewards* $\mathcal{J}_L^{(t)}$ *satisfy*

$$\left|\left\langle \left[\nabla_Q \mathcal{J}_L^{(t)}\right] x, x' \right\rangle\right| \leq d^{-\Omega(L)}, \qquad \mathcal{J}_L^{(t)} = \frac{1}{d}(1 + o(1)),$$

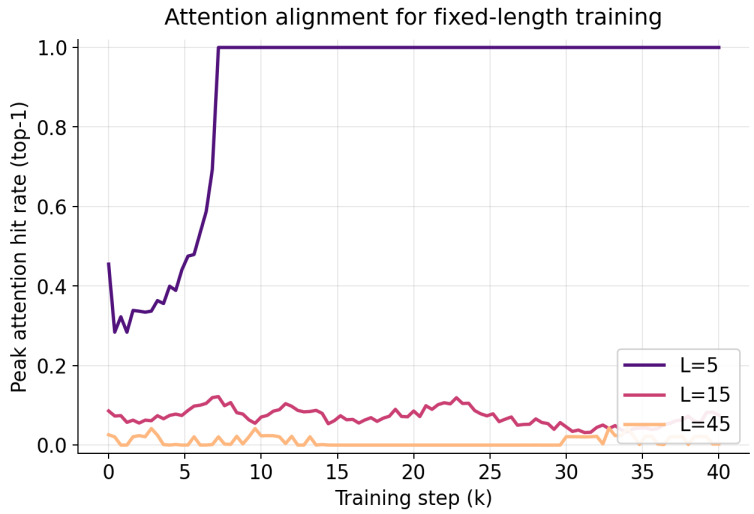
whenever $\max_{x,x' \in \mathcal{X}} \langle Q^{(t)} x, x' \rangle \leq 0.01$.

Numerical illustration

Experiment setup: we assume a cyclic group $\mathbb{Z}_{96}$ for reasoning length $L = 5, 15, 45$.



Average reward



Attention concentration

RL rapidly learns **short-horizon compositions**, whereas longer horizons exhibit a **near-flat reward plateau.**

Mixed-difficulty RLVR

The previous result showed that RL cannot operate beyond a critical horizon, even when the reward is non-zero by random guesses.

What if we **mix different horizons** together?

Mixed-difficulty distribution: we uniformly mix the horizons

$$\mathcal{L}_R = \{L_1, L_2, \ldots, L_K\}, \quad L_k = \min\{\lceil RL_{k-1} \rceil, L_{\max}\}.$$

Here, $R = L_{k+1}/L_k$ is the difficulty ratio, which controls how **smooth** the curriculum structure in the distribution.

Mixed-difficulty training objective:

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

$y_0 \xrightarrow{g_1} \cdots \xrightarrow{g_{L_1}} y_{L_1}$

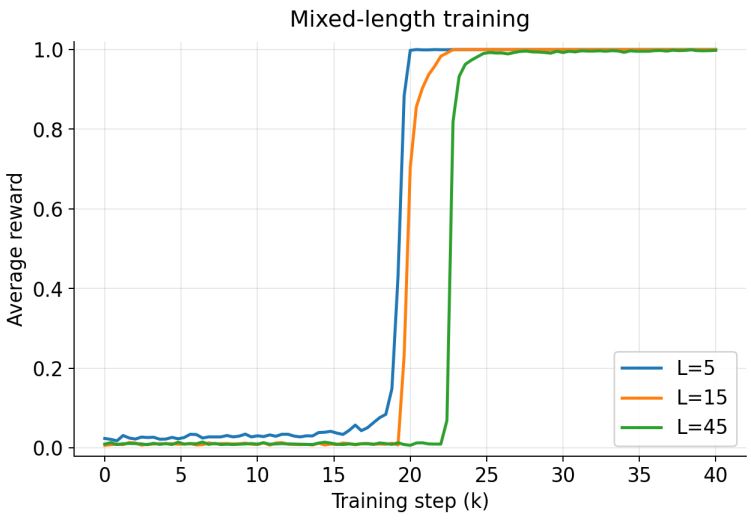
$y_0 \xrightarrow{g_1} y_1 \xrightarrow{g_2} y_2 \xrightarrow{g_3} \cdots \xrightarrow{g_{L_2}} y_{L_2}$

$\vdots$

$y_0 \xrightarrow{g_1} y_1 \xrightarrow{g_2} y_2 \xrightarrow{g_3} y_3 \xrightarrow{g_4} y_4 \xrightarrow{g_5} \cdots \xrightarrow{g_{L_K}} y_{L_K}$

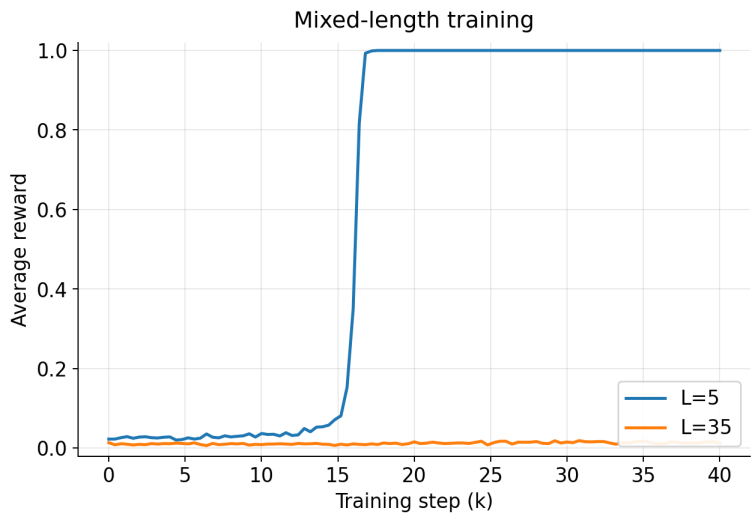
Learning behavior at different difficulty ratios

Relay



$R = 3$: a smoother difficulty spectrum facilitates relay dynamics across horizons.

Grokking



$R = 7$: a larger difficulty ratio yields a prolonged plateau at longer horizons.

The shared reasoning pattern generalizes across horizons through an implicit curriculum provided by the data mix.

Grokking under large difficulty ratio

Stopping times: define the following stopping times for

- *Visible return*: $T_{\mathrm{vis},L} := \min_t\{\mathcal{J}_L^{(t)} \geq 0.01\}$;
- *Mastery*: $T_{\mathrm{mas},L} := \min_t\{\mathcal{J}_L^{(t)} \geq 0.99\}$.

Theorem (Grokking)

Suppose $L_1 \leq C_B$, *and* $R \geq \omega(1)$, *then during training, for each* $k \in [1, K-2]$*:*

- **(Long plateaus)** *Before the next horizon* L_{k+1} *sees visible returns, there is a long plateau:* $T_{\mathrm{vis},k+1} - T_{\mathrm{mas},k} \gtrsim \frac{L_{\max}}{\eta} \cdot d^{C_B-1}$.
- **(Phase transitions)** *Once horizon* L_{k+1} *reaches visible return, it arrives at mastery time quickly:* $T_{\mathrm{mas},k} - T_{\mathrm{vis},k} \lesssim \frac{L_{\max}}{\eta} \cdot L_{k+1}$.

Relay under moderate difficulty ratios

Theorem (Relay)

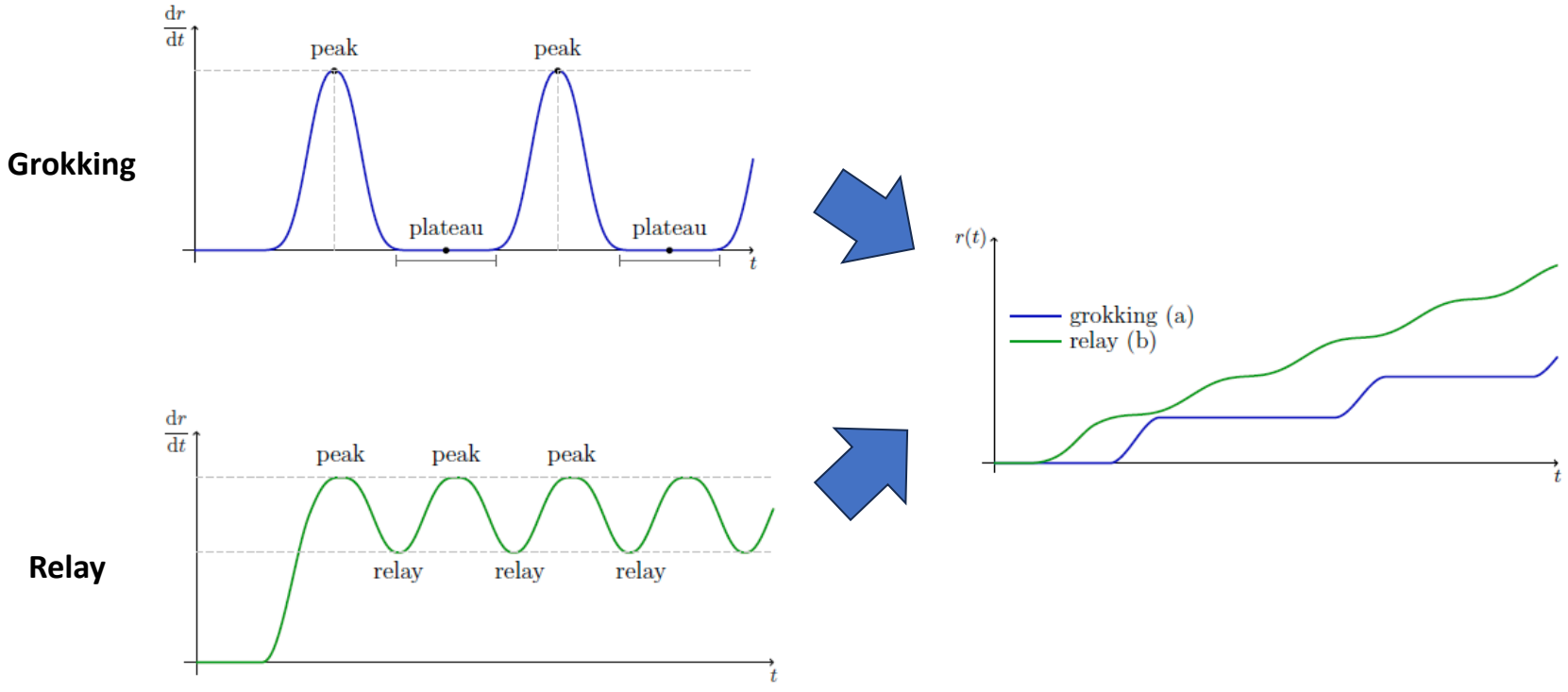
Suppose $L_1 \le C_B$, *and* $R \le O(1)$, *then during training, for each* $k \in [1, K-2]$:

- **(Short plateaus)** *Before* L_{k+1} *sees visible returns, the plateau period is short:* $T_{\mathrm{vis},k+1} - T_{\mathrm{mas},k} \lesssim \frac{L_{\max}}{\eta} \cdot d^{C_B(1-\frac{C_B}{C_B+R})-1}$.
- **(Phase transitions)** $T_{\mathrm{mas},k} - T_{\mathrm{vis},k} \lesssim \frac{L_{\max}}{\eta} \cdot L_{k+1}$.

Denote the first time that the longest horizon $L_{\max}$ reaches mastery state as T_{relay} and T_{grok}, then

$$T_{\mathrm{relay}} \le \tilde{O}(d^{-\Omega(1)}) \cdot T_{\mathrm{grok}}.$$

Mechanisms of RLVR dynamics



- **Grokking**: shorter-horizon saturates long before longer-horizon escapes random guessing.
- **Relay**: at the **edge of competence**, shorter-horizon still improves while longer-horizon already escapes randomness.

Gradient characterization

Step-wise probability margin: At each step, the policy assigns probability to:

- $p_{L,1}$: correct next state (target)
- $p_{L,2}$: wrong state from context (distractor)
- $p_{L,3}$: other wrong states

Define margin $\Delta_L := p_{L,1} - p_{L,3}$ (how much target dominates).

A key policy gradient characterization:

$$\nabla_q \widetilde{\mathcal{J}}_L \propto \underbrace{(\Delta_L)^L}_{\text{success over } L \text{ steps}} \cdot \underbrace{(1-\Delta_L)}_{\text{room to improve}},$$

where $q = \langle Q^t \mathfrak{s}(x), x \rangle$ for $x \in \mathcal{X}$.

- Δ_L small $\Rightarrow (\Delta_L)^L$ exponentially suppresses gradient (cold start)
- $\Delta_L \approx 1 \Rightarrow 1 - \Delta_L$ vanishes (saturated)

Regime comparison

$$\nabla_q \widetilde{\mathcal{J}}_L \propto \underbrace{(\Delta_L)^L}_{\text{success over } L \text{ steps}} \cdot \underbrace{(1-\Delta_L)}_{\text{room to improve}},$$

where $\Delta_L := p_{L,1} - p_{L,3}$.

Handover between horizons happens when

$$(\Delta_{L_{k+1}})^{L_{k+1}} \approx 1 - \Delta_{L_k}$$

- When $R = \omega(1)$: the happens when $1 - \Delta_{L_k}$ is already vanishingly small (saturated), so that $\nabla_q \mathcal{J}_{L_{k+1}}$ stays negligible over a long plateau – grokking.
- When $R \leq O(1)$: this happens when $1 - \Delta_{L_k}$ is still away from full saturation, so the non-zero periods of $\nabla_q \mathcal{J}_{L_k}$ and $\nabla_q \mathcal{J}_{L_{k+1}}$ overlap, and jointly drive the process – relay.

Fourier analysis on groups

Policy gradient requires computing

$$\nabla_q \mathcal{J}_L \propto \sum\nolimits_{\ell \in [L]} \left(\mathbb{P}(u_\ell = g_\ell \mid \hat{y}_L = y_L) - \mathbb{P}(u_\ell = g_\ell) \right),$$

where u_ℓ is the one-step group action.

The challenge: the terminal success $\hat{y}_L = y_L$ couples all L steps.

Fourier analysis on groups

Policy gradient requires computing

$$\nabla_q \mathcal{J}_L \propto \sum_{\ell \in [L]} \left(\mathbb{P}(u_\ell = g_\ell \mid \hat{y}_L = y_L) - \mathbb{P}(u_\ell = g_\ell) \right),$$

where u_ℓ is the one-step group action.

The challenge: the terminal success $\hat{y}_L = y_L$ couples all L steps.

Key insight: Success probability is an L-fold convolution on group $\mathcal{G}$:

$$\mathbb{P}(\hat{y}_L = y_L) = (\mu_L * \cdots * \mu_1)(G_*),$$

where $G_* = g_L \circ \ldots g_1$, and μ_ℓ is the one-step action law of u_ℓ on $\mathcal{G}$.

Fourier analysis on groups

Policy gradient requires computing

$$\nabla_q \mathcal{J}_L \propto \sum_{\ell \in [L]} \left(\mathbb{P}(u_\ell = g_\ell \mid \hat{y}_L = y_L) - \mathbb{P}(u_\ell = g_\ell) \right),$$

where u_ℓ is the one-step group action.

The challenge: the terminal success $\hat{y}_L = y_L$ couples all L steps.

Key insight: Success probability is an L-fold convolution on group $\mathcal{G}$:

$$\mathbb{P}(\hat{y}_L = y_L) = (\mu_L * \cdots * \mu_1)(G_*),$$

where $G_* = g_L \circ \ldots g_1$, and μ_ℓ is the one-step action law of u_ℓ on $\mathcal{G}$.

Fourier trick: Convolution $\xrightarrow{\text{Fourier}}$ multiplication

$$\widehat{\mu_L * \cdots * \mu_1} = \widehat{\mu}_L \cdot \widehat{\mu}_{L-1} \cdots \widehat{\mu}_1$$

$\Rightarrow$ Reduces L-step coupling to product of per-step operators!

Take-away messages

Message 1: the first end-to-end learning dynamics analysis for outcome-based RL with transformer-based policies.

Message 2: grokking-like phase transitions and the relay effect under mixed difficulty, explaining how RLVR learns at the edge of competence through implicit curriculum.

Technical side: a novel Fourier analysis on groups that makes long-horizon conditioning and compositional structure tractable.

Thanks!

- On the Learning Dynamics of RLVR at the Edge of Competence, arXiv 2602.14872.

https://yuejiechi.github.io