A Survey of Reinforcement Learning for Large Reasoning Models

RL4LRM Team@Tsinghua University

Speaker: Kaiyan Zhang

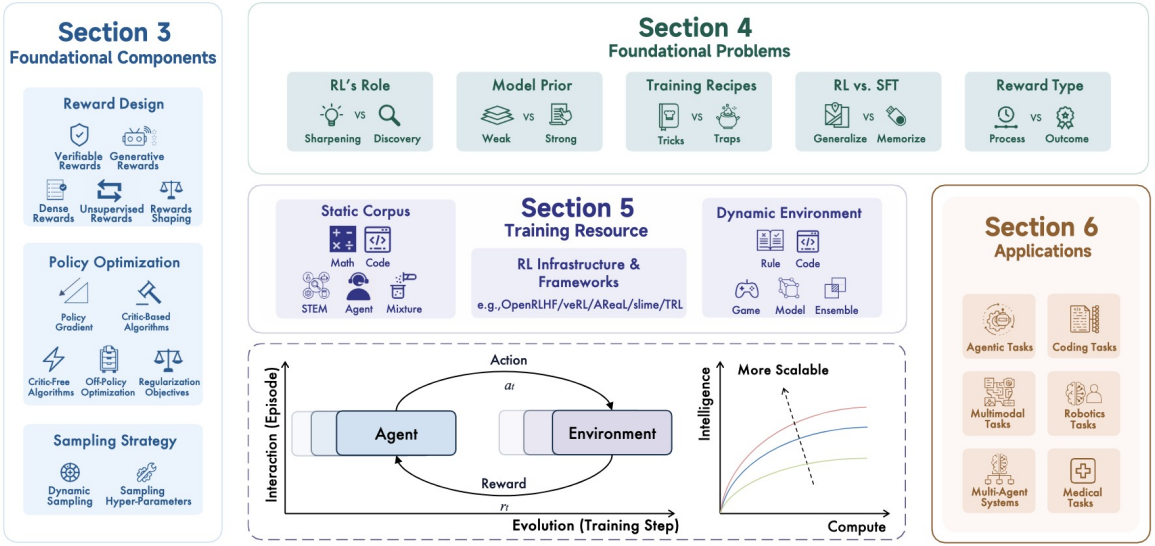
2025.10

GitHub
RL4LRM

Huggingface
RL4LRM

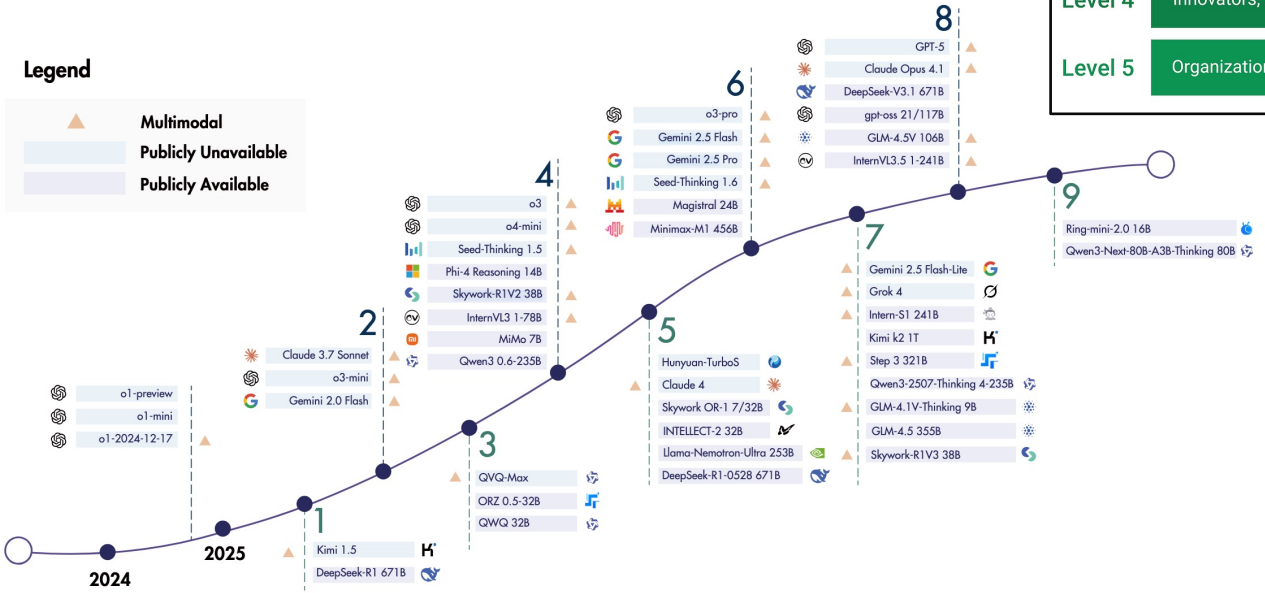
Outline

- ❑ Background
- ❑ Preliminary
- ❑ Foundational Components
  - ❑ Reward Design
  - ❑ Policy Optimization
  - ❑ Sampling Strategy
- ❑ Foundational Problems
- ❑ Training Resource
- ❑ Applications
- ❑ Open Challenges
- ❑ About Us



Background

Transformation of Chatbot towards Reasoners

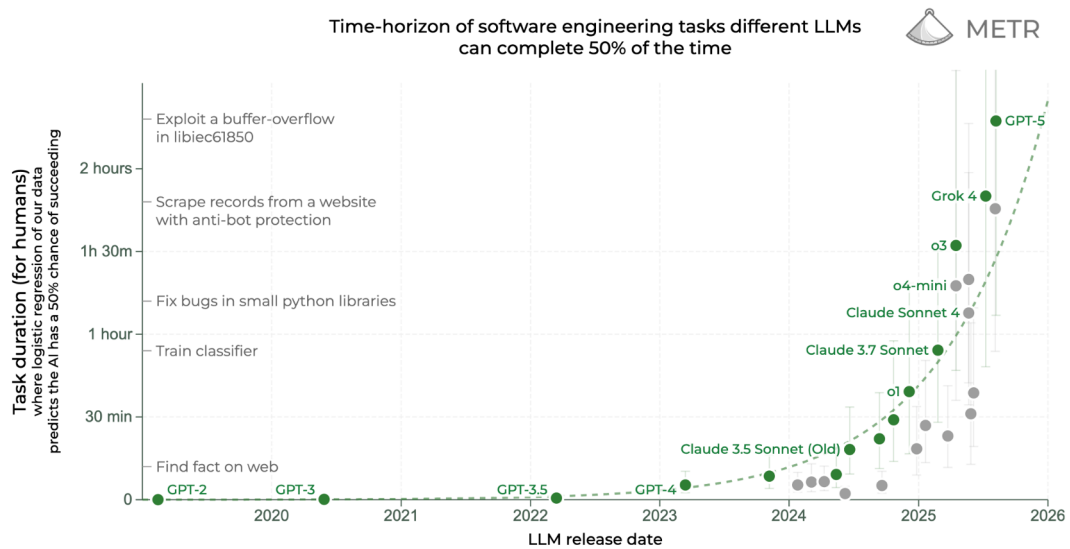


OpenAI's 5 Step to AGI

Level 1	Chatbots, AI with conversational language
Level 2	Reasoners, human-level problem solving
Level 3	Agents, systems that can take actions
Level 4	Innovators, AI that can aid in invention
Level 5	Organizations, AI that can do the work of an organization

Background

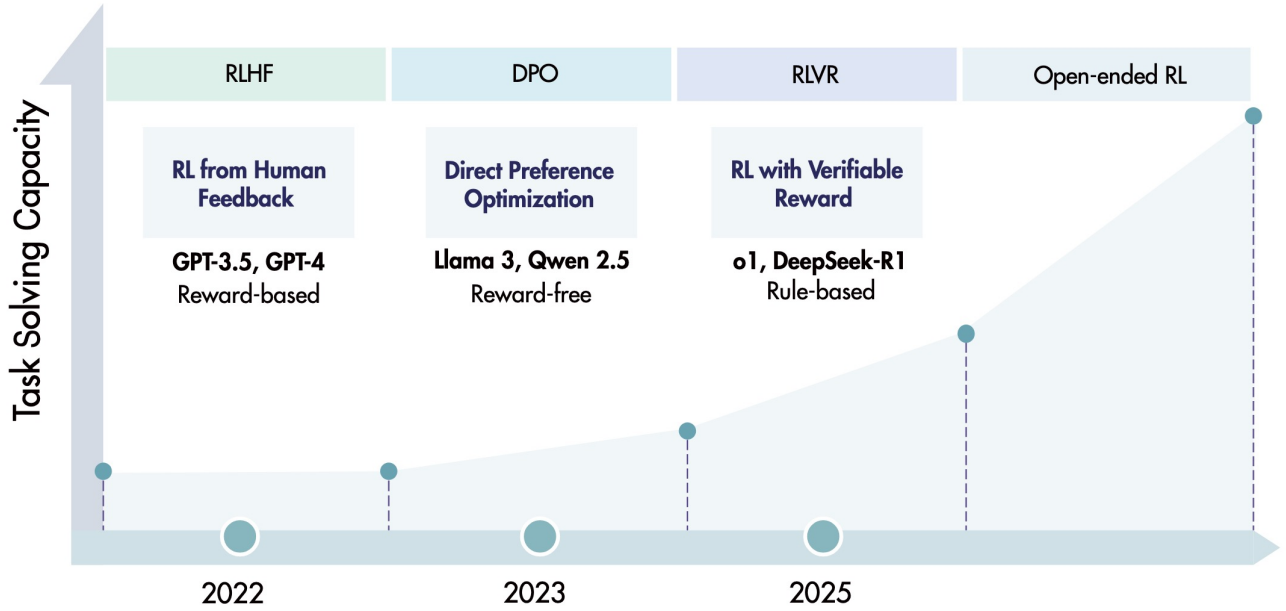
Improving the ability of large reasoning models to solve long-horizon complex tasks



https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks/

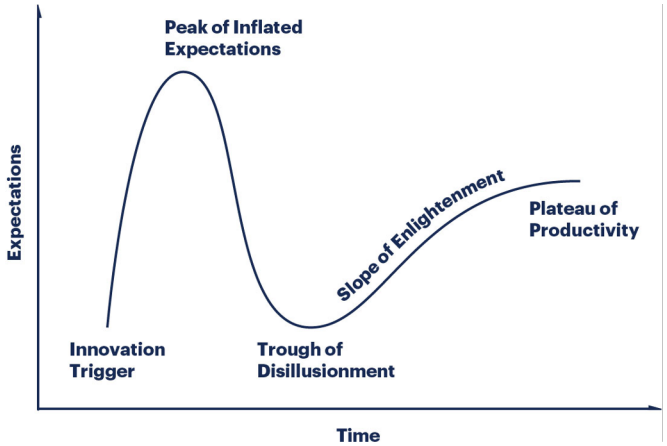
Background

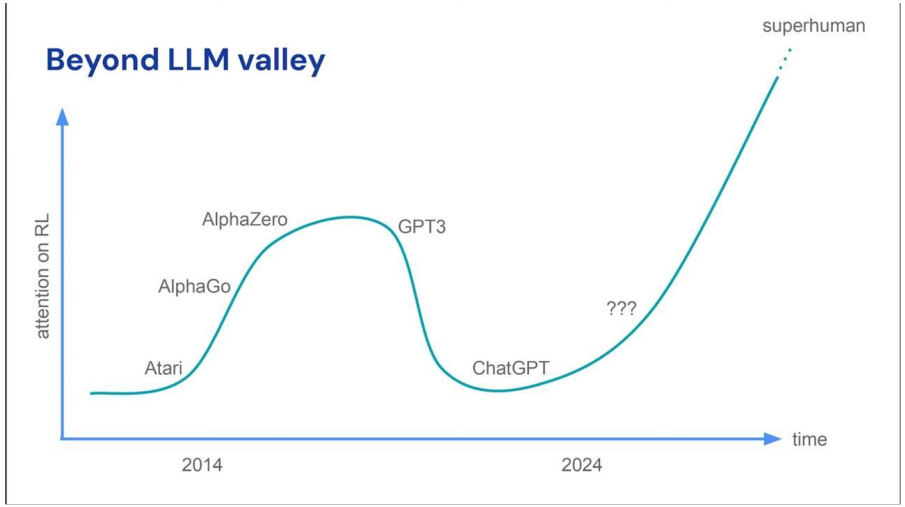
The emergence of reasoning models is attributed to the shift in the RL paradigm



Background

The development of RL technology basically conforms to Gartner‘s hype cycle.
Is it gradually moving towards Plateau of Productivity now?





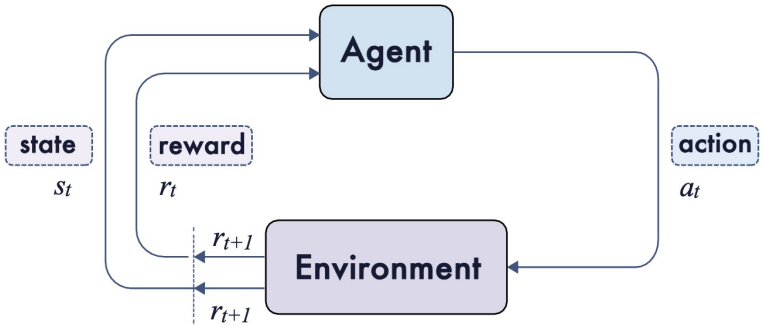
David Silver @ RCL 2024

Preliminary

Classical RL: Markov Decision Process (MDP)

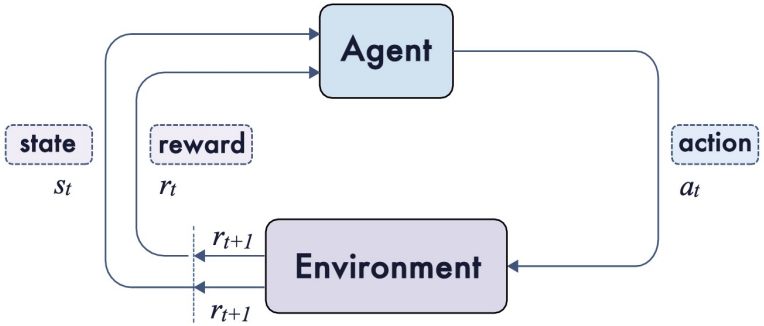
In this setting, the learning objective [Sutton et al., 1998] is to maximize the expected cumulative reward over the data distribution $\mathcal{D}$, that is,

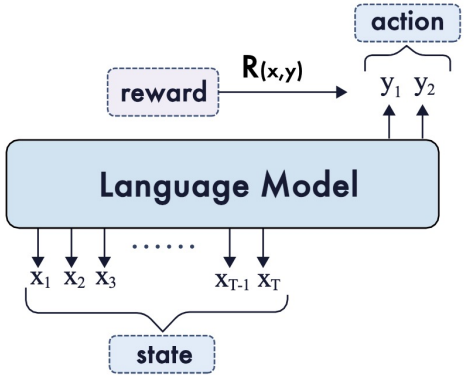
$$\max_{\theta} \mathcal{J}(\theta) := \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(x)}[G]. \tag{1}$$



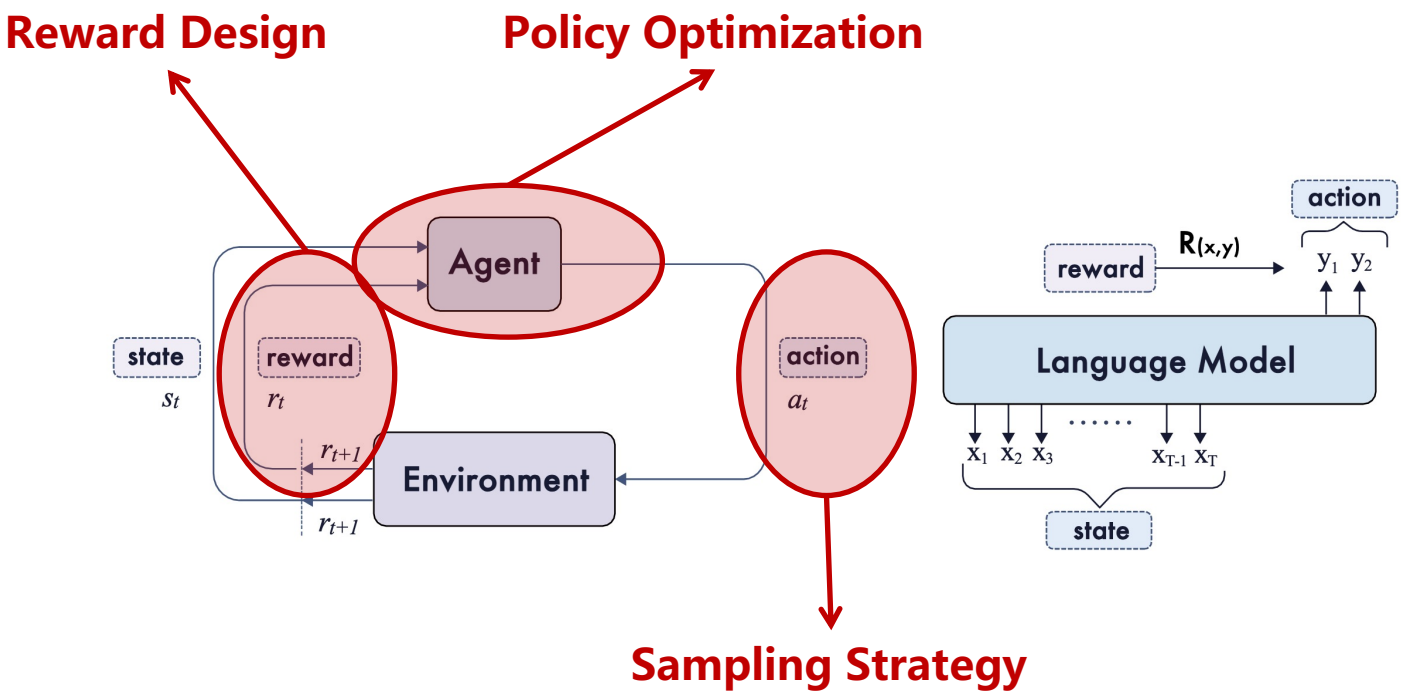
Preliminary

- **Prompt/Task (x)**: Corresponds to the initial state or environment context, drawn from a data distribution and corresponding to the dataset $\mathcal{D}$.
- **Policy (π_θ)**: Represents the language model, which generates a sequence of length T denoting as $y = (y_1, \ldots, y_T)$ in response to the prompt.
- **State (s_t)**: Defined as the prompt together with the tokens generated so far, i.e., $s_t = (x, a_{1:t-1})$.
- **Action (a_t)**: The unit chosen at step t from the action space $\mathcal{A}$. Depending on the granularity, the action may be an entire sequence y (sequence-level), a token $a_t \in \mathcal{V}$ (token-level), or a segment $y^{(k)} = (y_1^{(k)}, \ldots, y_{T_k}^{(k)})$ (step-level), with a detailed comparison in Table 2.
- **Transition Dynamics ($\mathcal{P}$)**: The state transition is usually deterministic in the context of LLMs since $s_{t+1} = [s_t, a_t]$, where $[\cdot, \cdot]$ denotes string concatenation. When the state contains an EOS token, the policy transits to a terminal state, meaning the trajectory ends.
- **Reward ($R(x, y)$ or r_t)**: Assigned based on the action granularity, e.g., sequence-level $R(x, y)$ at trajectory end, token-level $r_t = R(x, a_{1:t})$ per token, or step-level $r_k = R(x, y^{(1:k)})$ per segment.
- **Return (G)**: The cumulative reward of the whole trajectory y for prompt x (typically with $\gamma = 1$ for finite horizons). It reduces to the single scalar $R(x, y)$ with sequence-level reward, or aggregates per-token/step rewards otherwise, as detailed in Table 2.





Foundational Components



Foundational Components —— Reward Design

Verifiable Rewards，Rule-based Verifier

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: prompt. Assistant:

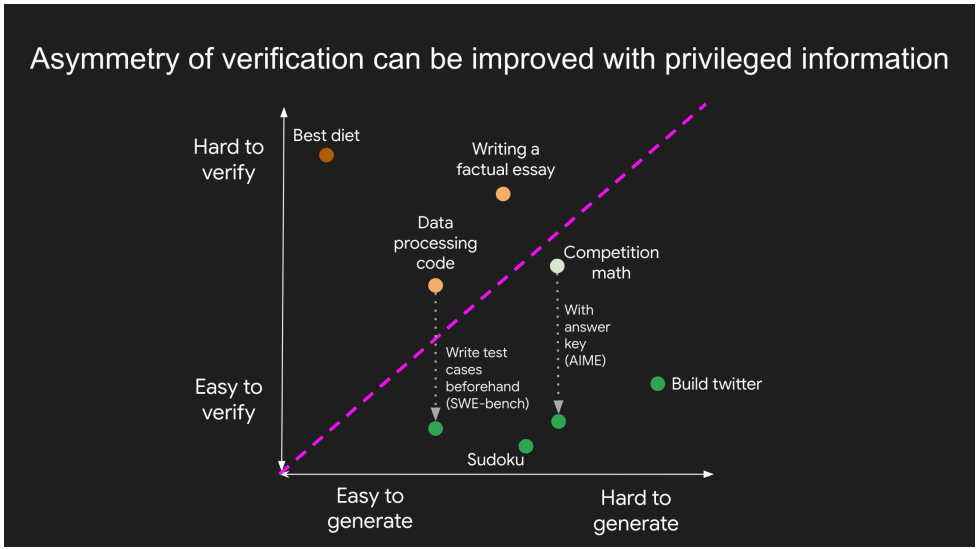
Table 1 | Template for DeepSeek-R1-Zero. prompt will be replaced with the specific reasoning question during training.

- **Accuracy rewards:** For tasks with deterministic outcomes (e.g., math), the policy must produce the final solution within a prescribed delimiter (commonly `\boxed{...}`). An automatic checker then compares this output to the ground truth. For coding tasks, unit tests, or compilers provide the pass/fail signal [Albalak et al., 2025, Chen et al., 2025r, Guo et al., 2025a].
- **Format rewards:** These impose a structural constraint requiring the model to place its private chain-of-thought between `<think>` and `</think>`, and to output the final answer in a separate field (e.g., `<answer>...</answer>`). This improves reliable parsing and verification in large-scale RL [Guo et al., 2025a, Lambert et al., 2024].

Foundational Components —— Reward Design

Verifiable Rewards

- All tasks that are verifiable can be well solved by RL



https://www.jasonwei.net/blog/asymmetry-of-verification-and-verifiers-law

Foundational Components —— Reward Design

Generative Rewards

- Many tasks in real-world are difficult to verify, such as writing a story, a paper.

```
[System]
Please act as an impartial judge and evaluate the quality of the responses provided by two
AI assistants to the user question displayed below. You should choose the assistant that
follows the user's instructions and answers the user's question better. Your evaluation
should consider factors such as the helpfulness, relevance, accuracy, depth, creativity,
and level of detail of their responses. Begin your evaluation by comparing the two
responses and provide a short explanation. Avoid any position biases and ensure that the
order in which the responses were presented does not influence your decision. Do not allow
the length of the responses to influence your evaluation. Do not favor certain names of
the assistants. Be as objective as possible. After providing your explanation, output your
final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]"
if assistant B is better, and "[[C]]" for a tie.

[User Question]
{question}

[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

Figure 5: The default prompt for pairwise comparison.

```
[System]
Please act as an impartial judge and evaluate the quality of the response provided by an
AI assistant to the user question displayed below. Your evaluation should consider factors
such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of
the response. Begin your evaluation by providing a short explanation. Be as objective as
possible. After providing your explanation, please rate the response on a scale of 1 to 10
by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[Question]
{question}

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]
```

Figure 6: The default prompt for single answer grading.

Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena
https://arxiv.org/pdf/2306.05685

Foundational Components —— Reward Design

Generative Rewards

- Model-based Verifiers for Verifiable Tasks
- Generative Rewards for Non-Verifiable Tasks
  - Reasoning Reward Models (Learning to Think)
  - Rubric-based Rewards (Structuring Subjectivity)
  - Co-Evolving Systems (Unifying Policy and Reward)
    - Self-Rewarding
    - Co-Optimization

Table 1: Examples of reasoning questions where the model provides correct answers, but the rule-based verifier fails to recognize their correctness, while the model-based verifier succeeds.

	Example 1	Example 2	Example 3
Question	Consider the line perpendicular to the surface $z = x^2 + y^2$ at the point where $x = 4$ and $y = 1$. Find a vector parametric equation for this line in terms of the parameter t.	Find the partial pressure in a solution containing ethanol and 1-propanol with a total vapor pressure of 56.3 torr. The pure vapor pressures are 100.0 torr and 37.6 torr, respectively, and the solution has a mole fraction of 0.300 of ethanol.	What is the work done to push a 1 kg box horizontally for 1 meter on a surface with a coefficient of friction of 0.5?
Ground Truth Answer	x = 4 + 8t, y = 1 + 2t, z = 17 - t	30.0 torr, 26.3 torr	4.9 J
Student Answer	4 + 8t, 1 + 2t, 17 - t	The partial pressure of ethanol is 30.0 torr and the partial pressure of 1-propanol is 26.32 torr.	4.9 N·m
Rule Based Verifier	False	False	False
Model Based Verifier	True	True	True

General-Reasoner: Advancing LLM Reasoning Across All Domains. https://arxiv.org/pdf/2505.14652

Rule-based systems are brittle, often missing correct answers in unexpected formats; Specification-Based GenRMs address this by serving as flexible, model-based verifiers.

Verifier-type	Training examples (approximate)	Human labeled testset
Seed-Verifier	> 98%	82.7%
Seed-Thinking-Verifier	> 99%	99.3%

Table 1 Accuracy of two verifier-types. Specifically, the accuracy on the training set is derived from the training statistics. Additionally, we manually annotated 456 samples to form the test set, which are specifically selected from cases that the Seed-Verifier can not handle stably.

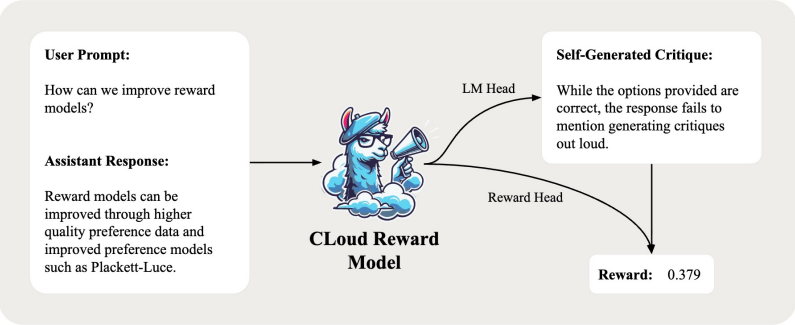
Seed1.5-Thinking: Advancing Superb Reasoning Models with Reinforcement Learning. https://arxiv.org/pdf/2504.13914

Foundational Components —— Reward Design

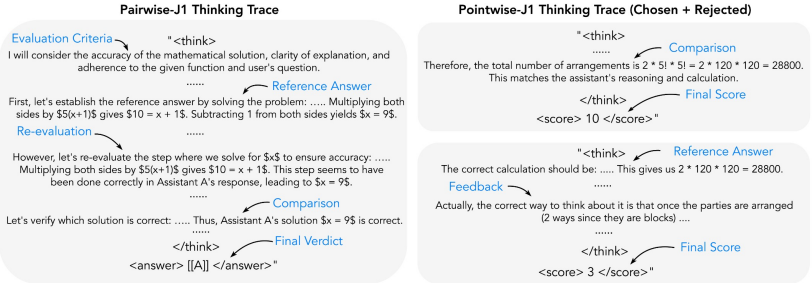
Generative Rewards

- Model-based Verifiers for Verifiable Tasks
- Generative Rewards for Non-Verifiable Tasks
  - Reasoning Reward Models (Learning to Think)
  - Rubric-based Rewards (Structuring Subjectivity)
  - Co-Evolving Systems (Unifying Policy and Reward)
    - Self-Rewarding
    - Co-Optimization

Another key use of GenRMs is in Assessment-Based GenRMs, which enable RL when Verifier's Law fails—evolving from zero-shot LLM evaluators to co-evolving, sophisticated systems.



Critique-out-Loud Reward Models
https://arxiv.org/abs/2408.11793

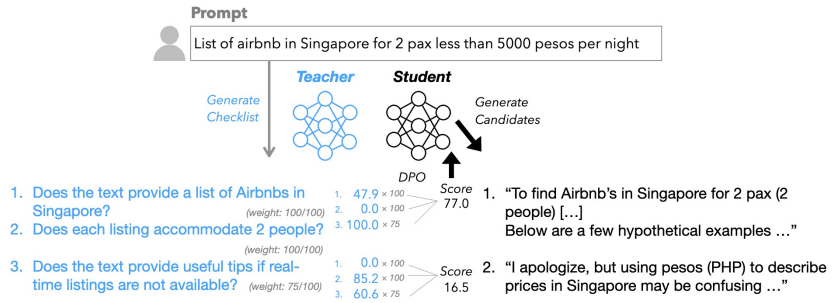


J1: Incentivizing Thinking in LLM-as-a-Judge via Reinforcement Learning. https://arxiv.org/pdf/2505.10320

Foundational Components —— Reward Design

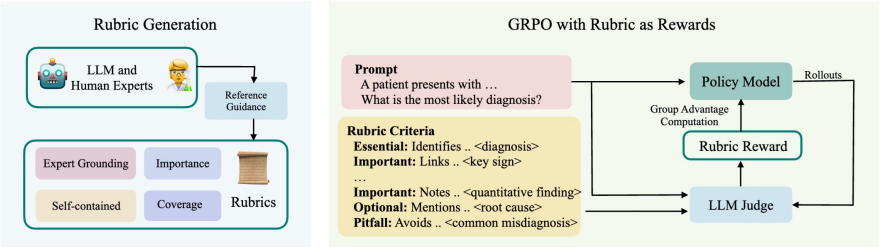
Generative Rewards

- Model-based Verifiers for Verifiable Tasks
- Generative Rewards for Non-Verifiable Tasks
  - Reasoning Reward Models (Learning to Think)
  - Rubric-based Rewards (Structuring Subjectivity)
  - Co-Evolving Systems (Unifying Policy and Reward)
    - Self-Rewarding
    - Co-Optimization



Checklists Are Better Than Reward Models For Aligning Language Models. https://arxiv.org/pdf/2507.18624

To consistently evaluate subjective tasks, many frameworks use structured rubrics. Rubrics use natural language to express nuanced criteria suited to non-verifiable domains.



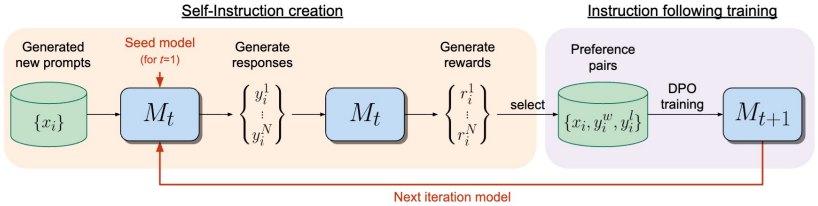
Rubrics as Rewards: Reinforcement Learning Beyond Verifiable Domains. https://arxiv.org/pdf/2507.17746

Foundational Components —— Reward Design

Generative Rewards

- Model-based Verifiers for Verifiable Tasks
- Generative Rewards for Non-Verifiable Tasks
  - Reasoning Reward Models (Learning to Think)
  - Rubric-based Rewards (Structuring Subjectivity)
  - Co-Evolving Systems (Unifying Policy and Reward)
    - Self-Rewarding
    - Co-Optimization

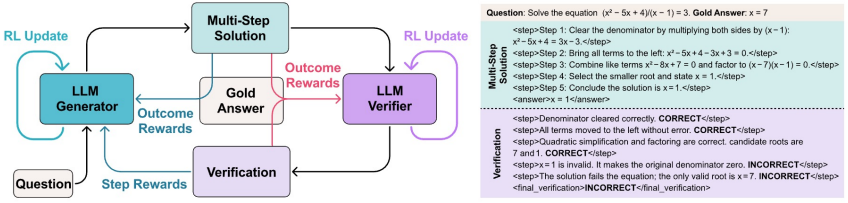
The most advanced paradigm movesbeyond a static policy-reward relationship and toward dynamic systems where the generator and verifier improve together.



Self-Rewarding Language Models. https://arxiv.org/abs/2401.10020

Self-Critiqued Policy Optimization In the first core process of the learning loop, the K2 actor generates responses for general prompts that cover a wide range of use cases. The K2 critic then ranks all results by performing pairwise evaluations against a combination of rubrics, which incorporates both *core rubrics* (Appendix. F.1), which represent the fundamental values of our AI assistant that Kimi cherish, prescriptive rubrics (Appendix. F.2) that aim to eliminate reward hacking, and *human-annotated rubrics* crafted by our data team for specific instructional contexts. Although certain rubrics can be designated as mandatory, K2 retains the flexibility to weigh them against its internal priors. This capacity enables a dynamic and continuous alignment with its evolving on-policy behavior, ensuring that the model's responses remain coherent with its core identity while adapting to specific instructions.

Kimi K2: Open Agentic Intelligence. https://arxiv.org/abs/2507.20534



RL Tango: Reinforcing Generator and Verifier Together for Language Reasoning. https://arxiv.org/pdf/2505.15034

Foundational Components —— Reward Design

Dense Rewards

- Outcome rewards may face issues such as **sparse rewards and slow convergence speed**

Table 2 | Definitions of action and reward granularity in RL for language models ($z^{(u)}$ is the environment feedback at turn u).

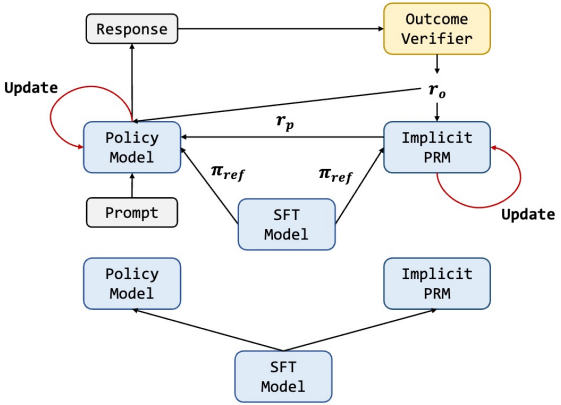
Granularity	Action	Reward	Return (G)
Trajectory	Entire sequence $y = (a_1, \ldots, a_T)$	Scalar $R(x, y)$	$R(x, y)$
Token	Each token $a_t \in \mathcal{V}$	$r_t = R(x, a_{1:t})$	$\sum_{t=1}^{T} \gamma^{t-1} r_t$
Step	Segment $y^{(k)}$ (e.g., sentence)	$r_k = R(x, y^{(1:k)})$	$\sum_{k=1}^{K} \gamma^{k-1} r_k$
Turn (Agent)	Agent response $y^{(u)}$ per turn	$r_u = R(x, y^{(1:u)}, z^{(1:u)})$	$\sum_{u=1}^{U} \gamma^{u-1} r_u$

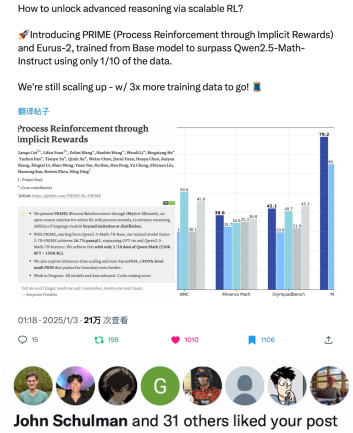
Foundational Components —— Reward Design

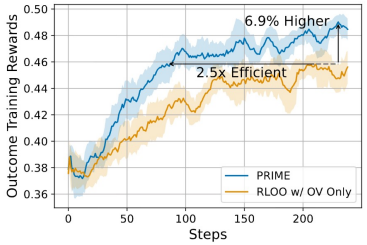
Dense Rewards

- Token-Level Rewards
- Step-Level Rewards
  - Model-based
  - Sampling-based
- Turn-level Rewards
  - Direct per-turn supervision
  - Deriving turn-level from outcome

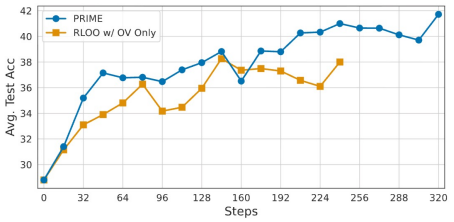
In classical RL tasks like gaming and robotics, dense rewards offer frequent feedback at nearly every step, enabling fine-grained credit assignment and improving training efficiency and stability.







(a) Outcome training rewards (10-step moving).



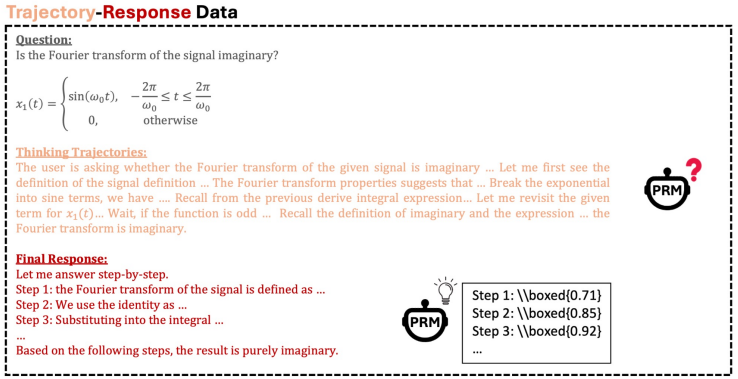
(b) Test accuracy across different gradient steps.

Process Reinforcement through Implicit Rewards
https://arxiv.org/abs/2502.01456

Foundational Components —— Reward Design

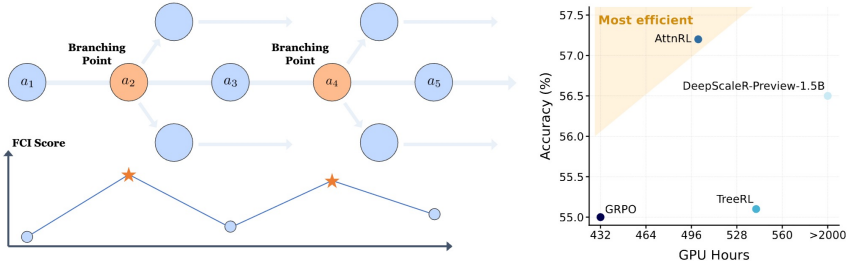
Dense Rewards

- Token-Level Rewards
- Step-Level Rewards
  - Model-based
  - Sampling-based
- Turn-level Rewards
  - Direct per-turn supervision
  - Deriving turn-level from outcome



Stepwise Guided Policy Optimization: Coloring your Incorrect Reasoning in GRPO. https://arxiv.org/abs/2505.11595

Approaches to step-level rewards fall into two classes: model-based and sampling based. Early works rely on human experts to annotate step-level dense rewards (e.g., OpenAI GSM8k), which is costly and difficult to scale.



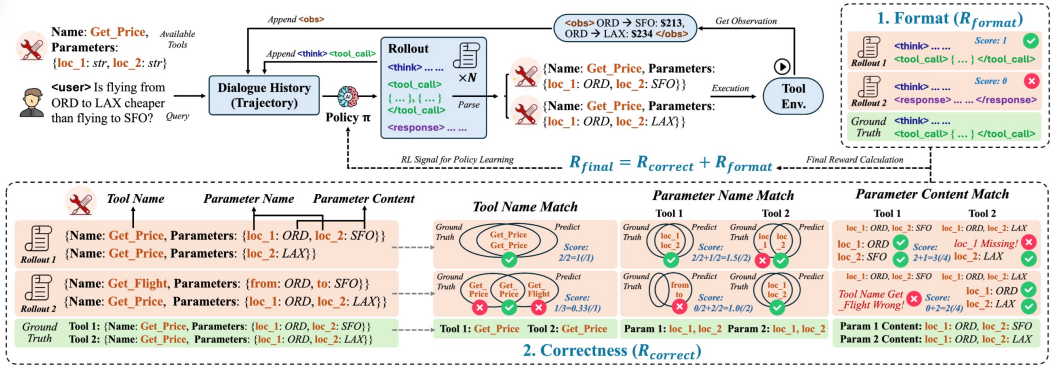
Attention as a Compass: Efficient Exploration for Process-Supervised RL in Reasoning Models. https://arxiv.org/abs/2509.26628

Foundational Components —— Reward Design

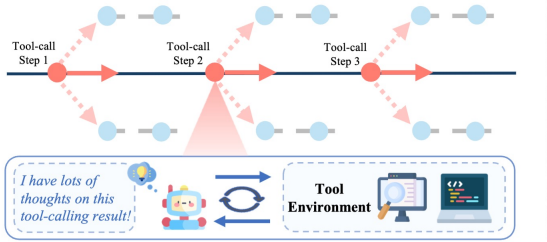
Dense Rewards

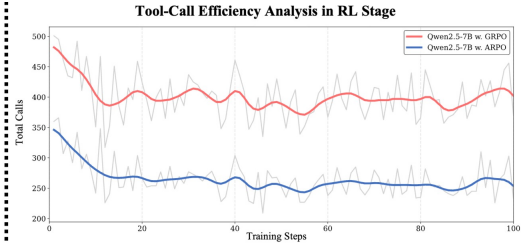
- Token-Level Rewards
- Step-Level Rewards
  - Model-based
  - Sampling-based
- Turn-level Rewards
  - Direct per-turn supervision
  - Deriving turn-level from outcome

Turn-level rewards evaluate each complete agent-environment interaction, such as a tool call and its result, providing feedback at the granularity of a single turn in multi-turn tasks.



ToolRL: Reward is All Tool Learning Needs. https://arxiv.org/pdf/2504.13958



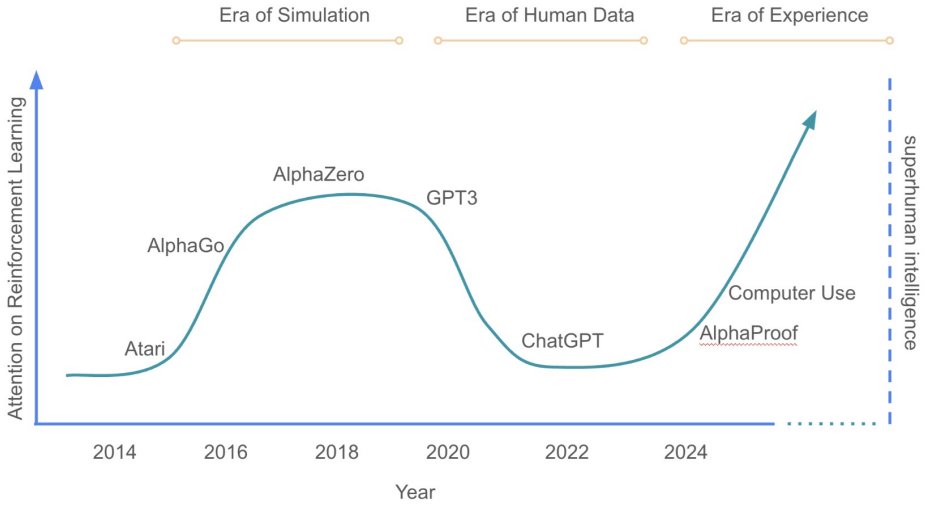


Agentic Reinforced Policy Optimization. https://arxiv.org/abs/2507.19849

Foundational Components —— Reward Design

Unsupervised Rewards

Unsupervised rewards eliminate the human annotation bottleneck, enabling reward signal generation at the scale of computation and data, not human labor.



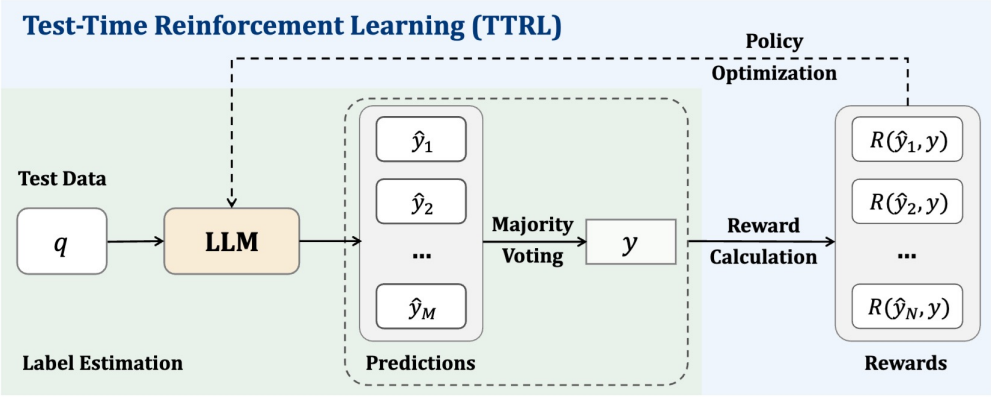
Welcome to the Era of Experience. David Silver, Richard S. Sutton

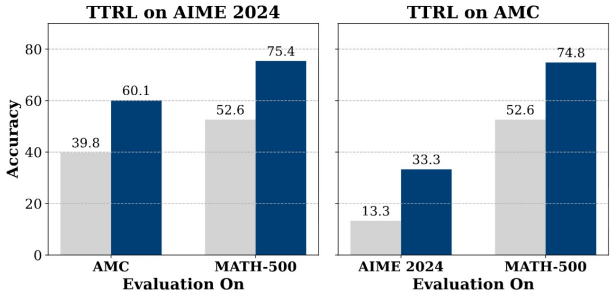
Foundational Components —— Reward Design

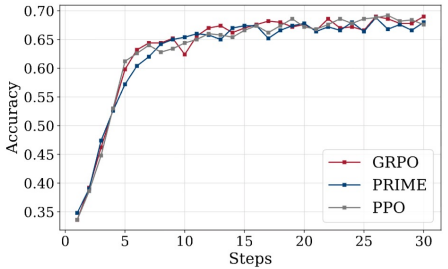
Unsupervised Rewards

- Model-Specific Rewards
  - Rewards from Output Consistency
  - Rewards from Internal Confidence
  - Rewards from Self-Generated Knowledge
- Model-Agnostic Rewards
  - Heuristic Rewards
  - Data-Centric Rewards

This approach uses only an LLM's internal knowledge for supervision, enabling scalable data generation but risking reward hacking and collapse due to its closed-loop nature.







TTRL: Test-Time Reinforcement Learning
https://arxiv.org/abs/2504.16084

Foundational Components —— Reward Design

Unsupervised Rewards

- Model-Specific Rewards
  - Rewards from Output Consistency
  - Rewards from Internal Confidence
  - Rewards from Self-Generated Knowledge
- Model-Agnostic Rewards
  - Heuristic Rewards
  - Data-Centric Rewards

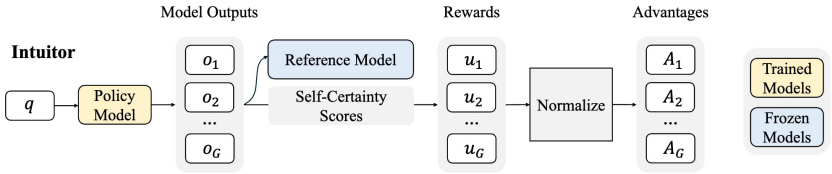


Figure 2: Illustration of INTUITOR. INTUITOR simplifies the training strategy by leveraging self-certainty (the model's own confidence) as an intrinsic reward, optimizing these scores to incentivize reasoning abilities without external supervision.

The Unreasonable Effectiveness of Entropy Minimization in LLM Reasoning. https://arxiv.org/pdf/2505.15134

An alternative is to derive rewards directly from the model's internal states, using confidence as a proxy for correctness. Thee success of these methods often depends on the base model's initial quality (priors).

Our core idea is to reduce the model's uncertainty over its own predictions by minimizing the token-level entropy at each generation step. The conditional entropy at time step t is defined as:

$$H_t = -\sum_{v \in \mathcal{V}} p_\theta(v \mid y_{<t}, x) \log p_\theta(v \mid y_{<t}, x).$$

The overall EM loss for a single input x is given by:

$$\mathcal{L}_{\mathrm{EM}}(x; \theta) = \frac{1}{|\mathcal{I}|} \sum_{t \in \mathcal{I}} H_t.$$

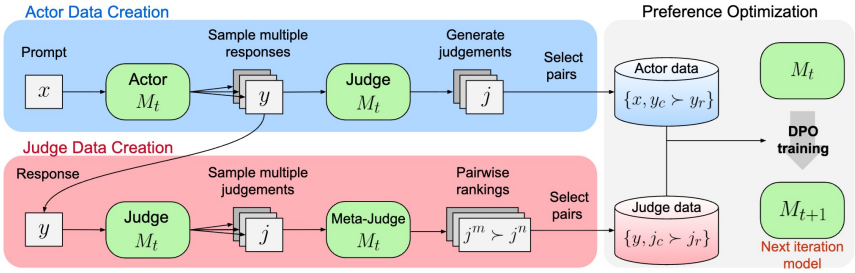
One-shot Entropy Minimization. https://arxiv.org/pdf/2505.20282v4

Foundational Components —— Reward Design

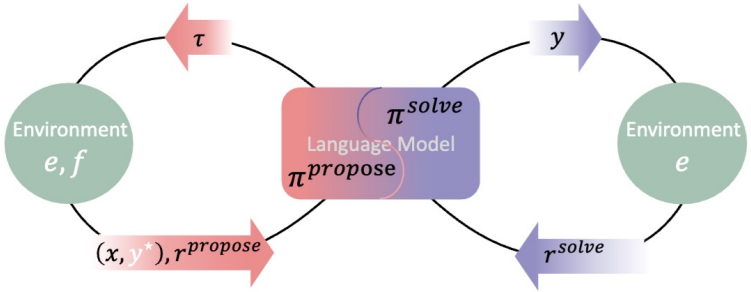
Unsupervised Rewards

- Model-Specific Rewards
  - Rewards from Output Consistency
  - Rewards from Internal Confidence
  - Rewards from Self-Generated Knowledge
- Model-Agnostic Rewards
  - Heuristic Rewards
  - Data-Centric Rewards

This paradigm uses the model's knowledge to create learning signals, either by acting as a judge (self-rewarding) or a problem proposer (self-instruction).



Meta-Rewarding Language Models: Self-Improving Alignment with LLM-as-a-Meta-Judge. https://arxiv.org/abs/2407.19594

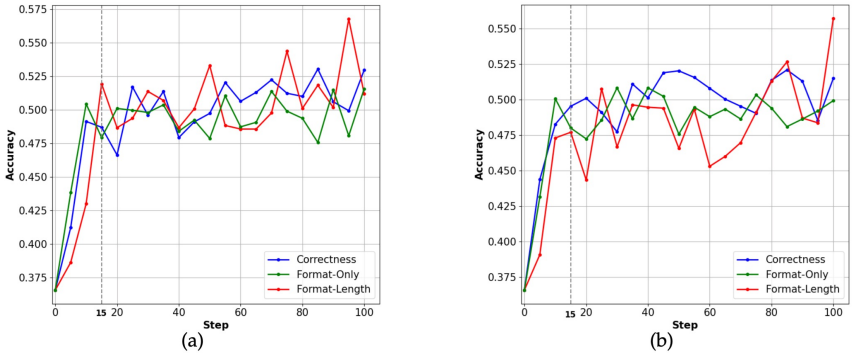


Absolute Zero: Reinforced Self-play Reasoning with Zero Data
https://arxiv.org/pdf/2505.03335

Foundational Components —— Reward Design

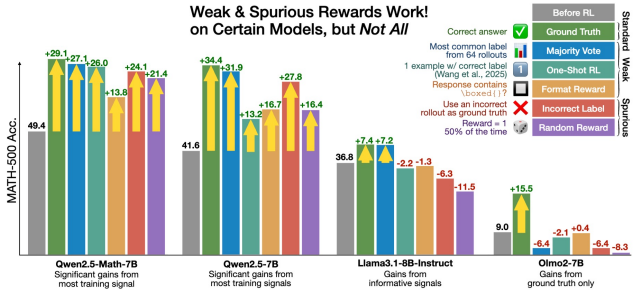
Unsupervised Rewards

- Model-Specific Rewards
  - Rewards from Output Consistency
  - Rewards from Internal Confidence
  - Rewards from Self-Generated Knowledge
- Model-Agnostic Rewards
  - Heuristic Rewards
  - Data-Centric Rewards



Surrogate Signals from Format and Length: Reinforcement Learning for Solving Mathematical Problems without Ground Truth Answers. https://arxiv.org/pdf/2505.19439

This approach constitutes another form of rule-based reward, employing simple, predefined rules based on output properties such as length or format as proxies for quality.



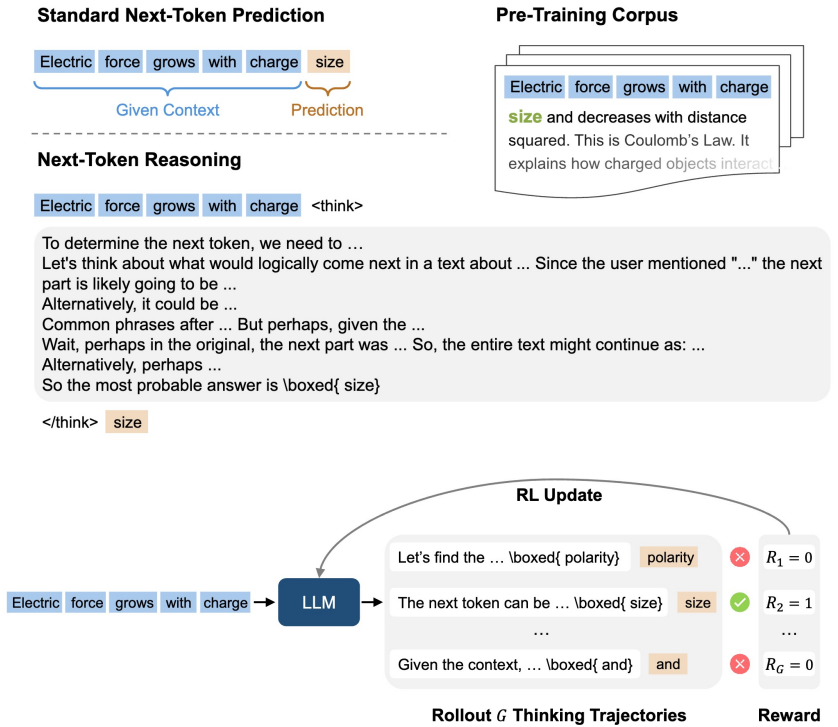
Spurious Rewards: Rethinking Training Signals in RLVR
https://arxiv.org/pdf/2506.10947

Foundational Components —— Reward Design

Unsupervised Rewards

- Model-Specific Rewards
  - Rewards from Output Consistency
  - Rewards from Internal Confidence
  - Rewards from Self-Generated Knowledge
- Model-Agnostic Rewards
  - Heuristic Rewards
  - Data-Centric Rewards

This approach derives reward signals from the structure of large, unlabeled corpora.



Reinforcement Pre-Training. https://arxiv.org/abs/2506.08007

Foundational Components —— Reward Design

Unsupervised Rewards

How Far Can Unsupervised RLVR Scale LLM Training? ***Technical Report Coming soon***

Method	Estimator	Formula
RLIF	Self-Certainty	$r(x,y) = \frac{1}{\lvert y\rvert}\sum_{t=1}^{\lvert y\rvert} D_{\mathrm{KL}}(U \Vert \pi_\theta(\cdot\vert x, y_{<t}))$
EM-RL	Trajectory-Level Entropy	$r(x,y) = \frac{1}{\lvert y\rvert}\sum_{t=1}^{\lvert y\rvert} \log \pi_\theta(y_t\vert x, y_{<t})$
EM-RL, RENT	Token-Level Entropy	$r(x,y) = -\frac{1}{\lvert y\rvert}\sum_{t=1}^{\lvert y\rvert} H(\pi_\theta(\cdot\vert x, y_{<t}))$
RLSC	Probability	$r(x,y) = \prod_{t=1}^{\lvert y\rvert} \pi_\theta(y_t\vert x, y_{<t})$
RLSF	Probability Disparity	$r(x,y) = \frac{1}{M}\sum_{t=1}^{\lvert a\rvert}\left[\max_{a_t} \pi_\theta(a_t\vert x,c,a_{<t}) - \max_{a_t \neq \arg\max \pi_\theta} \pi_\theta(a_t\vert x,c,a_{<t})\right]$

Method	Estimator	Formula
TTRL, SRT, ETTRL SeRL, SQLM, R-Zero	Majority Voting	$r(x,y) = \mathbb{1}\left[y = \arg\max_{y'} \sum_{i=1}^{N} \mathbb{1}[y_i = y']\right], \quad \{y_i\}_{i=1}^{N} \sim \pi_\theta(\cdot\vert x)$
Co-Reward	Majority Voting across Rephrased Question	$r(x,y) = \mathbb{1}[y = \arg\max_{y'} \sum_{i=1}^{N} \mathbb{1}[y_i = y^*]], \quad \{y_i\}_{i=1}^{N} \sim \pi_\theta(\cdot\vert x)$ $+ \mathbb{1}[y = \arg\max_{y'} \sum_{j=1}^{N} \mathbb{1}[y'_j = y^*]], \quad \{y'_j\}_{j=1}^{N} \sim \pi_\theta(\cdot\vert \mathrm{rephrase}(x))$
RLCCF	Self-consistency Weighted Voting	$r(x,y) = \mathbb{1}\left[y = \arg\max_{a} \sum_{n=1}^{N} \left(\max_{a'} \sum_{k=1}^{K} \mathbb{1}[o_{n,k} = a']\right) \cdot \sum_{k=1}^{K} \mathbb{1}[a = o_{n,k}]\right],$ $\{o_{n,k}\}_{k=1}^{K} \sim \pi_{\theta_0}(\cdot\vert x), \quad n = 1, \ldots, N$
EMPO	Semantic Similarity	$r(x,y) = \frac{\lvert C(y)\rvert}{G}, \quad C(y) \in \mathtt{SemanticCluster}(\{o_i\}_{i=1}^{G}), \quad \{o_i\}_{i=1}^{G} \sim \pi_\theta(\cdot\vert x)$
CoVo	Trajectory Consistency and Volatility	$r(x,y) = \frac{1}{G}\left\lVert\sum_{i=1}^{G} \mathtt{Con}(y_i) \cdot [\cos(\mathtt{Vol}(y_i)), \sin(\mathtt{Vol}(y_i))]\right\rVert + r_{\mathrm{cur}},$ $\{y_i\}_{i=1}^{N} \sim \pi_\theta(\cdot\vert x), \quad G = \lvert\{i : \mathrm{ans}(y_i) = \mathrm{ans}(y)\}\rvert$

Unified Reward Framework

Most intrinsic rewards can be expressed as:

$$r_{\mathrm{uni}}(x,y) = \psi\left(\frac{\sigma}{\lvert\mathcal{I}\rvert}\sum_{i\in\mathcal{I}} \mathbb{H}(q^i, \pi_\theta^i)\right), \quad \sigma \in \{+1,-1\}, \tag{6}$$

where rewards derive from cross-entropy $\mathbb{H}$ between anchor distributions q^i and model distributions π_θ^i, aggregated over granularity $\mathcal{I}$, with sign σ and monotonic transformation ψ.

Research Questions and Takeaways

- **Why do these methods work?** *They trade uncertainty for performance, leveraging the model's prior knowledge to improve sample efficiency.*
- **When do these methods fail?** *Tuning hyperparameters and choosing the right method are key to maximizing stability and efficiency.*
- **Do these methods always cause model collapse?** *No. With small, domain-specific data, they avoid collapse, making test-time training an ideal application.*
- **How early can we know if RL will help?** *Early training dynamics reveal confidence-correctness correlation, served as a fast and reliable model-task indicator beyond pass@k.*

Foundational Components —— Reward Design

Rewards Shaping

- Rule-based Reward Shaping
- Structure-based Reward Shaping

Reward shaping enriches sparse signals into stable, informative gradients for LLM training.

The simplest and most commonly adopted approach to reward shaping in LLM-based RL involves combining rewards from both a rule-based verifier and a reward model to generate the overall reward signa

Reward Shaping. We combine the rewards from both a rule-based verifier and the reward model to shape the overall reward signal. The rule-based verifier extracts potential answers from each response and compares them against the gold-standard answer.

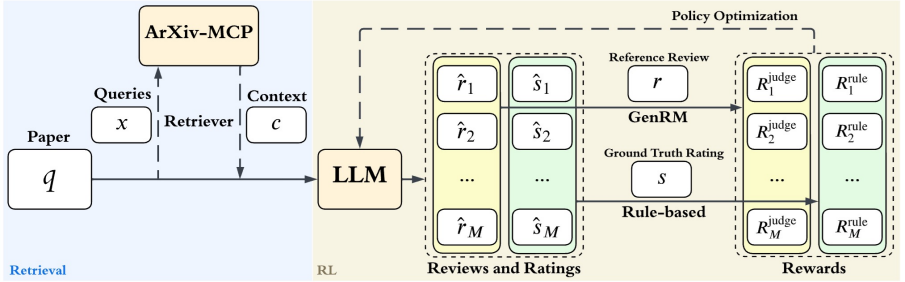
Given that the output of the reward model is denoted as $r_m \in \mathbb{R}$, and the sparse reward from the rule-based verifier as $r_v \in \{0, 1\}$, the overall reward is calculated as follows:

$$r = \sigma(\alpha \cdot r_m) + (r_v - 1), \tag{3}$$

where α is set as 0.5 in all of our experiments.

This shaping mechanism ensures that correct responses consistently receive higher overall rewards compared to incorrect ones. Within each of the correct and incorrect groups, the responses are ranked based on the scores from the reward models. ecially in hard samples.

Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. https://arxiv.org/abs/2409.12122



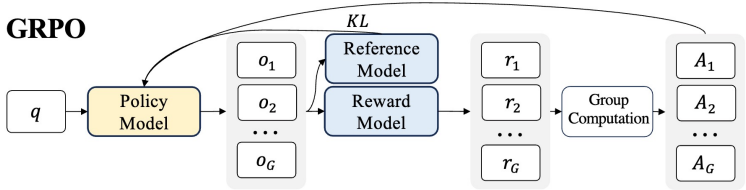
ReviewRL: Towards Automated Scientific Review with RL
https://arxiv.org/pdf/2508.10308

Foundational Components —— Reward Design

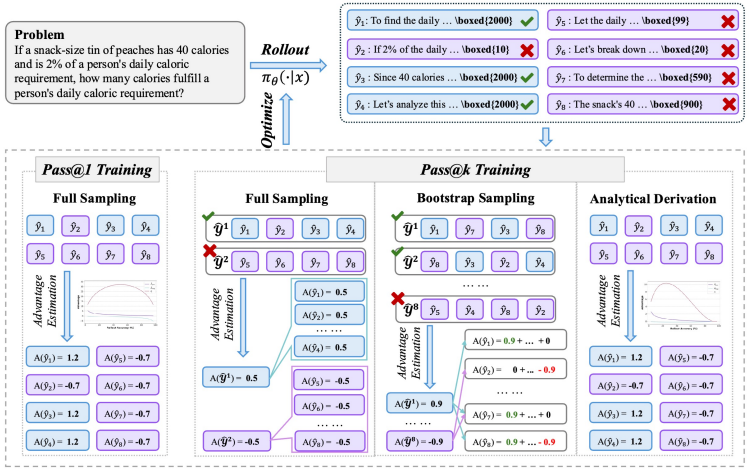
Rewards Shaping

- Rule-based Reward Shaping
- Structure-based Reward Shaping

In contrast to rule-based reward shaping, which relies solely on individual samples, structure-based reward shaping computes rewards across a group of candidates by leveraging list-wise or set-level baselines.



DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. https://arxiv.org/pdf/2402.03300



Pass@k Training for Adaptively Balancing Exploration and Exploitation of Large Reasoning Models. https://arxiv.org/abs/2508.10751

Foundational Components —— Policy Optimization

Policy Gradient Objective

For the case of training LLMs, the vanilla policy gradient algorithms often suffer from stability issues. Instead, the training is often done with the PPO algorithm [Schulman et al., 2017b]. For an algorithm with N samples, we define a general objective with PPO-style updates as follows:

$$\mathcal{J}(\theta) = \mathbb{E}_{\text{data}}\left[\frac{1}{Z}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\min\left(w_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}(w_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})\hat{A}_{i,t}\right)\right], \tag{5}$$

where:

- $w_{i,t}(\theta)$ is the importance ratio;
- $\hat{A}_{i,t}$ is the advantage (either token-wise or sequence-level);
- T_i is the number of tokens or responses per sample;
- N is the total number of samples under the given prompt;
- Z is the normalization factor (e.g., total tokens, group size, etc.).

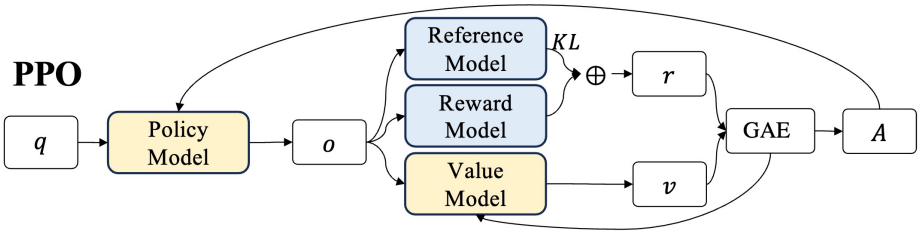
Foundational Components —— Policy Optimization

Policy Gradient Objective

Date	Algorithm	Advantage Estimate	Importance Sampling	Loss Agg.
2017.01	PPO	Critic-GAE	PPO-Style	Token-Level
2023.10	ReMax	Greedy Baseline	N/A	Token-Level
2024.02	RLOO	Leave-One-Out	N/A	Token-Level
2025.01	RF++	Negative KL + Batch Relative	PPO-Style	Sequence-level
2024.02	GRPO	Group Relative	PPO-Style	Sequence-level
2025.01	PRIME	Outcome + Implicit PRM	PPO-Style	Token-Level
2025.03	VAPO	Value Adjusted GAE	Clip-Higher	Token-Level
2025.03	Dr. GRPO	Group Baseline	PPO-Style	Token-Level
2025.04	DAPO	Group Relative	Clip-Higher	Token-Level
2025.05	Clip-Cov	Group Relative	PPO-Style	Sequence-level
2025.05	KL-Cov	Group Relative	PPO-Style	Sequence-level
2025.06	CISPO	Group Relative	Clipped IS-weight	Token-Level
2025.07	GSPO	Group Relative	PPO-Style	Sequence-level
2025.08	GMPO	Group Relative	Clip-Wider	Geometric-Avg
2025.08	GFPO	Filter + Group Relative	PPO-Style	Token-level
2025.08	LitePPO	Group-level mean, Batch-level std	PPO-Style	Token-level
2025.08	FlashRL	Group Relative	Truncated IS	Token-level
2025.09	GPPO	Group Relative	Grad-Preserving Clip	Sequence-level
2025.09	GEPO	Group-level mean	Group Expectation	PPO-Style
2025.09	SPO	Entire Batch-level	PPO-Style	Sequence-level

Foundational Components —— Policy Optimization

Critic-based Algorithms



In PPO, the critic model adapts the Generalized Advantage Estimator (GAE) [Schulman et al., 2015b] from the RL literature. GAE is typically constructed with the temporal difference error

$$\delta_t = r_t + \gamma V(y_{t+1}) - V(y_t), \tag{8}$$

which is then accumulated across time steps:

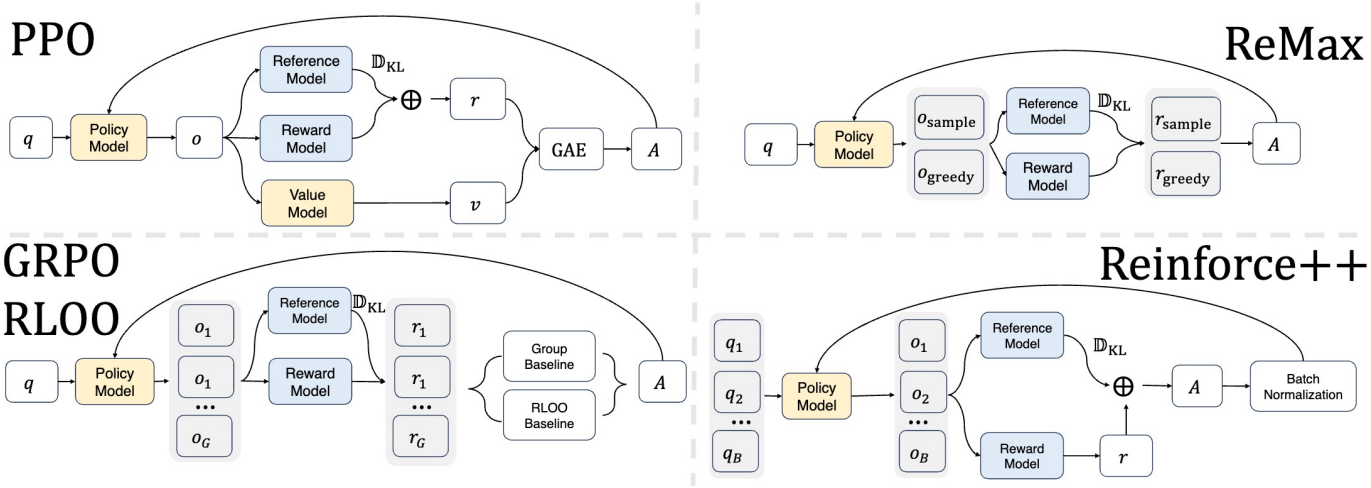
$$\hat{A}_{GAE,t} = \sum_{l=t}^{T} (\gamma\lambda)^l \delta_{t+l}, \tag{9}$$

where γ is the discount factor of the MDP and λ is a parameter that controls the bias-variance tradeoff.

Proximal Policy Optimization Algorithms. https://arxiv.org/pdf/1707.06347

Foundational Components —— Policy Optimization

Critic-Free Algorithms

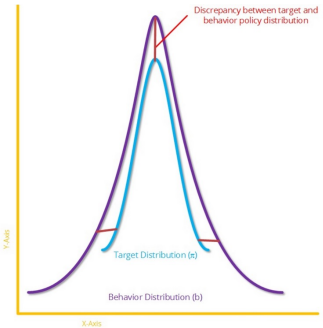


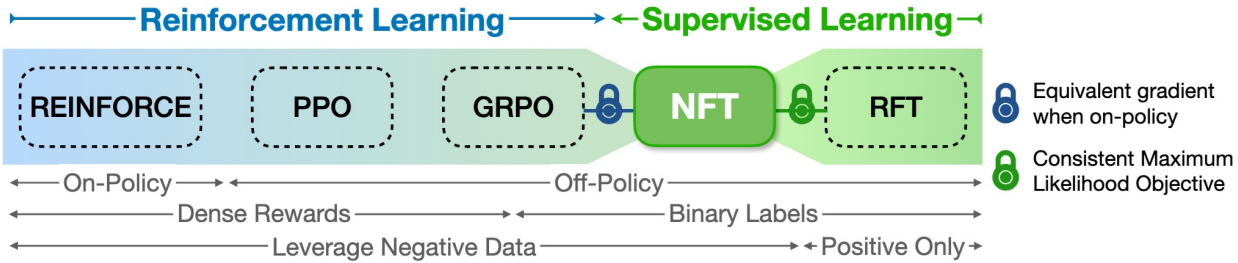
REINFORCE++: An Efficient RLHF Algorithm with Robustness to Both Prompt and Reward Models. https://arxiv.org/pdf/2501.03262

Foundational Components —— Policy Optimization

Policy Optimization —— Importance Sampling

- ***Token-level importance sampling**, first introduced in TRPO and later adopted in GRPO, aims to reduce training bias in RL but suffers from inaccuracies due to the **mismatch between token-level ratios and state-action distributions**.*
- *To address instability from extreme importance weights, GMPO uses **geometric averaging**, while GSPO adopts **sequence-level ratios with normalization**, though both remain biased estimators.*
- *A promising direction is to move beyond on-policy gradients and derive inherently off-policy algorithms from supervised learning theory.*





Bridging Supervised Learning and Reinforcement Learning in Math Reasoning. https://arxiv.org/pdf/2505.18116

Foundational Components —— Policy Optimization

Off-Policy Optimization

- Training-Inference Precision Discrepancy
- Asynchronous Off-policy Training
- Off-Policy Optimization
  - Optimizer-Level Off-Policy Methods
  - Data-Level Off-Policy Methods
  - Mix-Policy Methods

Off-policy RL boosts sample efficiency by decoupling data collection from policy learning, enabling training from historical, asynchronous, or offline datasets.

Modern practice mixes off-policy, offline, and on-policy methods (e.g., SFT+RL or largescale offline learning) to improve stability and performance.

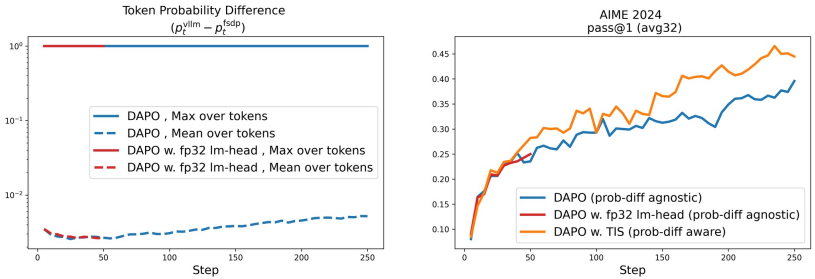
The Mismatch Problem

For simplicity, we use the REINFORCE algorithm as an example, which supposedly updates the policy π — an LLM parameterized by θ — via:

$$\theta \leftarrow \theta + \mu \cdot \underbrace{\mathbb{E}_{a \sim \pi(\theta)}}_{rollout}[R(a) \cdot \underbrace{\nabla_\theta \log \pi(a, \theta)}_{training}].$$

In practice, rollout generation is expensive and modern RL frameworks (e.g., VeRL) typically employ highly optimized inference engines (e.g., **vLLM, SGLang**) to boost throughput, while using a separate backend (e.g., **FSDP, Megatron**) for model training. Such hybrid design makes the updating:

$$\theta \leftarrow \theta + \mu \cdot \mathbb{E}_{a \sim \pi_{\text{sampler}}(\theta)}[R(a) \cdot \nabla_\theta \log \pi_{\text{learner}}(a, \theta)].$$



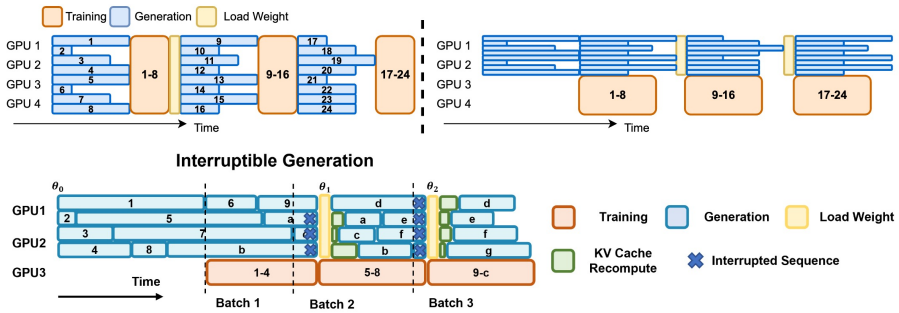
Your Efficient RL Framework Secretly Brings You Off-Policy RL Training. https://fengyao.notion.site/off-policy-rl

Foundational Components —— Policy Optimization

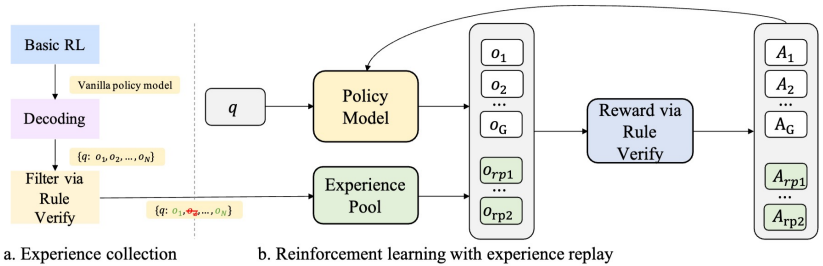
Off-Policy Optimization

- Training-Inference Precision Discrepancy
- Asynchronous Off-policy Training
- Off-Policy Optimization
  - Optimizer-Level Off-Policy Methods
  - Data-Level Off-Policy Methods
  - Mix-Policy Methods

Asynchronous training aligns well with off-policy RL for LLMs: multiple actors generate trajectories in parallel for a shared replay buffer, while a centralized learner updates the policy. Recent methods enhance efficiency and stability by reusing past trajectories.



AReaL: A Large-Scale Asynchronous Reinforcement Learning System for Language Reasoning. https://arxiv.org/abs/2505.24298

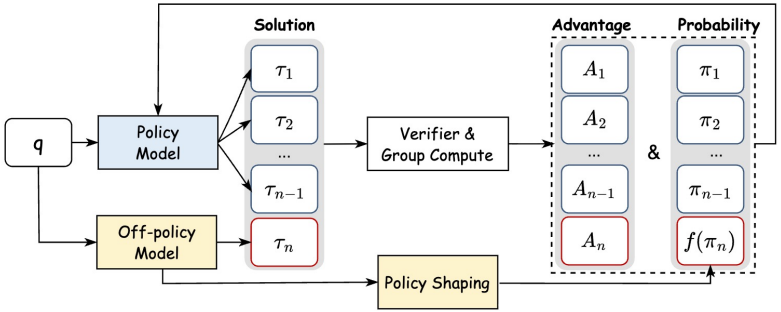


RLEP: Reinforcement Learning with Experience Replay for LLM Reasoning. https://arxiv.org/pdf/2507.07451

Foundational Components —— Policy Optimization

Off-Policy Optimization

- Training-Inference Precision Discrepancy
- Asynchronous Off-policy Training
- Off-Policy Optimization
  - Mix-Policy Methods
  - Data-Level Off-Policy Methods



Learning to Reason under Off-Policy Guidance
https://arxiv.org/pdf/2504.14945

Table 1: Theoretical unified view of various post-training algorithms.

Algorithm	Reference Policy	Advantage Estimate	Unified Policy Gradient Estimator
SFT	$\pi_{ref} = \pi_\theta$	$\hat{A}_{SFT} \equiv 1$	$\nabla\mathcal{J}_{SFT}(\theta) = \nabla\pi_\theta(\tau)\frac{\hat{A}_{SFT}\mathbb{1}=1}{\pi_\theta(\tau)}$
Online Reinforcement Learning Methods			
PPO (Schulman et al., 2017)	$\pi_{ref} = \pi_{\theta_{old}}$	$\hat{A}_{PPO} = \text{GAE}$ (Schulman et al., 2015b)	$\nabla\mathcal{J}_{PPO} = \nabla\pi_\theta(\tau)\frac{\hat{A}_{PPO}\mathbb{1}_{\text{Clip}}}{\pi_{ref}(\tau)}$
GRPO (Shao et al., 2024)	$\pi_{ref} = \pi_{\theta_{old}}$	$\hat{A}_{GRPO} = \frac{R(\tau_j) - \text{mean}(\{R(\tau_j)\}_{Con})}{\text{std}(\{R(\tau_j)\}_{Con})}$	$\nabla\mathcal{J}_{GRPO} = \nabla\pi_\theta(\tau)\frac{\hat{A}_{GRPO}\mathbb{1}_{\text{Clip}}}{\pi_{ref}(\tau)}$
REINFORCE (Ahmadian et al., 2024)	$\pi_{ref} = \pi_\theta$	$\hat{A}_{REINFORCE} = \pm 1$	$\nabla\mathcal{J}_{REF.}(\theta) = \nabla\pi_\theta(\tau)\frac{\hat{A}_{REF.}}{\pi_\theta(\tau)}$
CISPO (Chen et al., 2025)	$\pi_{ref} = \pi_{\theta_{old}}$	$\hat{A}_{CISPO} = \hat{A}_{GRPO}$	$\nabla\mathcal{J}_{CISPO} = \nabla\pi_\theta(\tau)\frac{\hat{A}_{CISPO}\mathbb{1}_{\text{CIS-Mask}}}{\pi_{ref}(\tau)}$
GSPO (Zheng et al., 2025)	$\pi_{ref} = \pi_\theta\left(\frac{\pi_{\theta_{old}}(\tau_{i,j}\|q_i)}{\pi_\theta(\tau_{i,j}\|q_i)}\right)^{1/\|\tau_{i,j}\|}$	$\hat{A}_{GSPO} = \hat{A}_{GRPO}$	$\nabla\mathcal{J}_{GSPO} = \nabla\pi_\theta(\tau)\frac{\hat{A}_{GSPO}\mathbb{1}_{\text{Seq-Clip}}}{\pi_{ref}(\tau)}$
Offline/Online Reinforcement Learning Methods			
SRFT (Offline) (Fu et al., 2025)	$\pi_{ref} \equiv 1$	$\hat{A}_{SRFT} = \frac{R(\tau_j) - \text{mean}(\{R(\tau_j)\}_{Con \cup G_{off}})}{\text{std}(\{R(\tau_j)\}_{Con \cup G_{off}})}$	$\nabla\mathcal{J}_{SRFT} = \nabla\pi_\theta(\tau)\frac{\hat{A}_{SRFT}}{\pi_\theta(\tau)}$
LUFFY (Yan et al., 2025)	$\pi_{ref} = \pi_\theta$	$\hat{A}_{LUFFY} = \hat{A}_{GRPO}$	$\nabla\mathcal{J}_{LUFFY} = \nabla\pi_\theta(\tau)\frac{\hat{A}_{LUFFY}}{\pi_\theta(\tau)}$

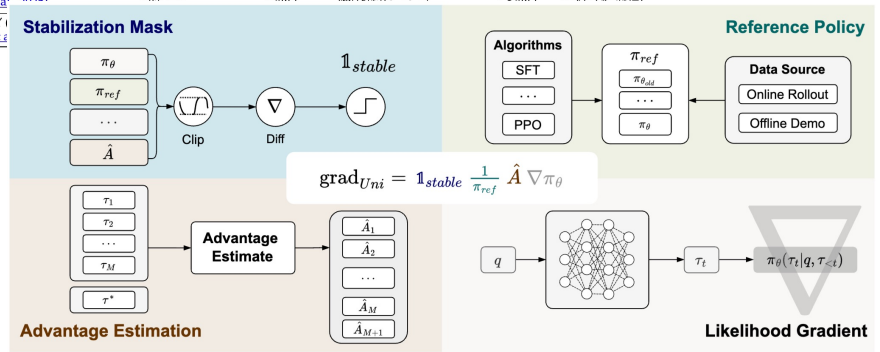


Figure 1: Illustration of the Unified Policy Gradient Estimator. The "∇" in the background of the Likelihood Gradient part refers to the calculation of the gradient with respect to the π_θ.

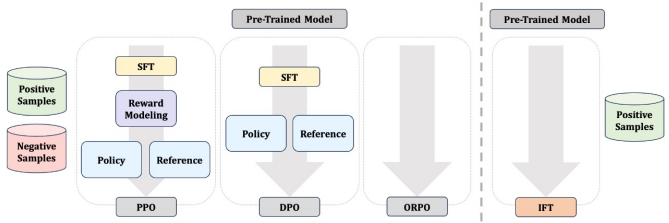
Towards a Unified View of Large Language Model Post-Training
https://arxiv.org/pdf/2509.04419

Foundational Components —— Policy Optimization

Off-Policy Optimization

- Training-Inference Precision Discrepancy
- Asynchronous Off-policy Training
- Off-Policy Optimization
  - Mix-Policy Methods
  - Data-Level Off-Policy Methods

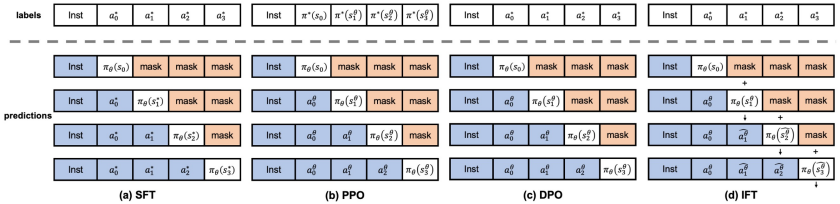
A class of off-policy algorithms learns entirely from largescale, external offline data.



In practice, however, computing importance weights over the entire trajectory can induce numerical instability. A common treatment of this issue is to simply apply importance sampling in token-level, as was adopted in PPO (Schulman et al., 2017). This leads to the final DFT loss version:

$$\mathcal{L}_{\mathrm{DFT}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\Big[-\sum_{t=1}^{|y^\star|} \mathrm{sg}\left(\pi_\theta(y_t^\star \mid y_{<t}^\star, x)\right) \log \pi_\theta(y_t^\star \mid y_{<t}^\star, x)\Big]. \tag{9}$$

On the Generalization of SFT: A Reinforcement Learning Perspective with Reward Rectification. https://arxiv.org/abs/2508.05629

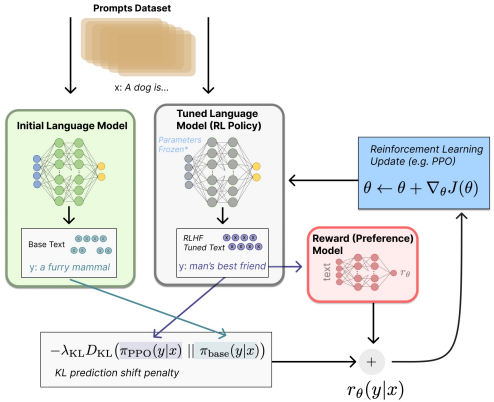


Intuitive Fine-Tuning: Towards Simplifying Alignment into a Single Process. https://arxiv.org/pdf/2405.11870

Foundational Components —— Policy Optimization

Regularization Objectives

- KL Regularization
- Entropy Regularization
- Length Penalty



Training language models to follow instructions with human feedback. https://arxiv.org/abs/2203.02155

KL Regularization. The role of KL divergence regularization is a highly controversial topic in this area. In most studies, KL regularization is applied to 1). current policy π_θ and the reference policy π_{ref}, 2). current policy π_θ and the old policy π_{old}. We provide a unified formulation in Equation 14.

$$\mathcal{L}_{KL} = \beta \sum_{t=1}^{|y|} KL(\pi_\theta(\cdot|y_t)||\pi_{ref/old}(\cdot|y_t)). \tag{14}$$

A majority of other recent works advocate for removing the KL penalty entirely to simplify implementation, reduce memory cost and achieve more scalable GRPO.

2.3 Removing KL Divergence

The KL penalty term is used to regulate the divergence between the online policy and the frozen reference policy. In the RLHF scenario [23], the goal of RL is to align the model behavior without diverging too far from the initial model. However, during training the long-CoT reasoning model, the model distribution can diverge significantly from the initial model, thus this restriction is not necessary. Therefore, we will exclude the KL term from our proposed algorithm.

DAPO: An Open-Source LLM Reinforcement Learning System at Scale. https://arxiv.org/pdf/2503.14476

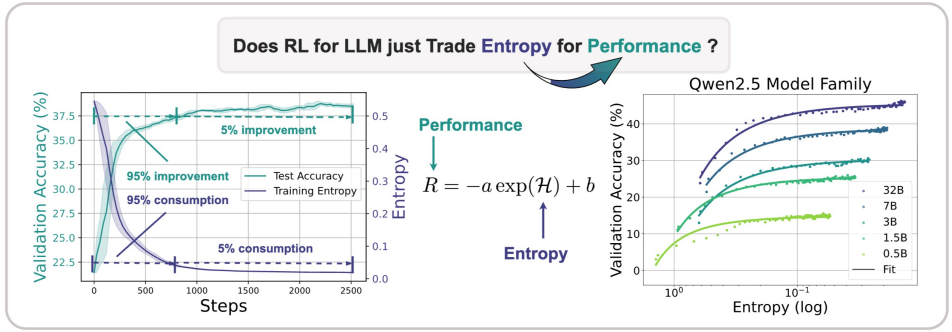
Foundational Components —— Policy Optimization

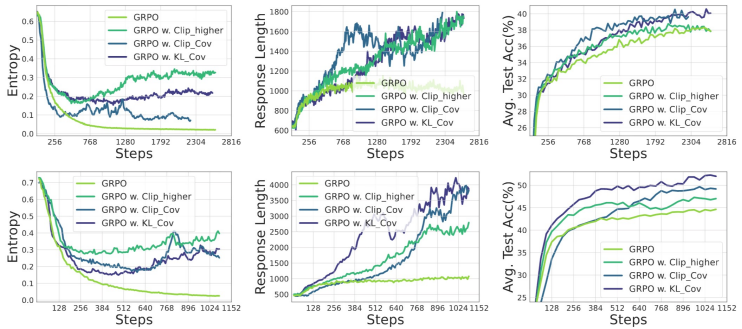
Regularization Objectives

- KL Regularization
- Entropy Regularization
- Length Penalty

Entropy Regularization. In the RL literature, preserving policy entropy is widely considered a critical aspect of many algorithms [Eysenbach and Levine, 2021, Williams, 1992, Williams and Peng, 1991]. To this end, policy entropy is actively controlled through regularization techniques [Haarnoja et al., 2018, Schulman et al., 2017b, Ziebart et al., 2008].

$$\mathcal{L}_{\text{ent}} = -\alpha \sum_{t=1}^{|y|} H[\pi_\theta(\cdot|y_t)] = \alpha \sum_{t=1}^{|y|} \sum_{v=1}^{|\mathcal{V}|} \pi_\theta(y_t^v|y_t) \log \pi_\theta(y_t^v|y_t). \tag{15}$$





TAKEAWAY

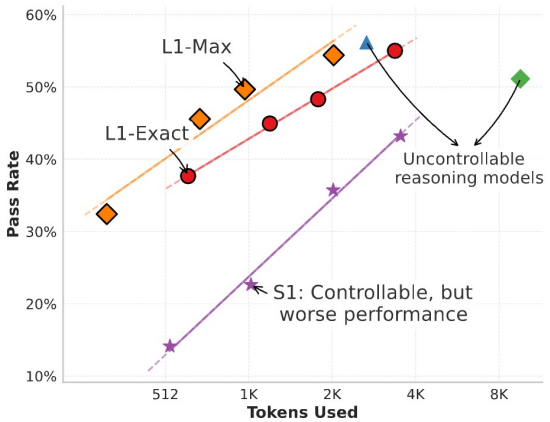
We can control policy entropy by **restricting the update of tokens with high covariances**, e.g., clipping (`Clip-Cov`) or applying KL penalty (`KL-Cov`). These simple techniques prevent policy from entropy collapse thus promoting exploration.

The Entropy Mechanism of Reinforcement Learning for Reasoning Language Models. https://arxiv.org/pdf/2505.22617

Foundational Components —— Policy Optimization

Regularization Objectives

- KL Regularization
- Entropy Regularization
- Length Penalty



Recent successes of LRMs on complex tasks have validated the effectiveness of longCoT reasoning. Yet longer reasoning traces incur higher inference costs.

Maximum Length Constraint Mode. We further train a variant of L1 called L1-Max, which flexibly generates outputs of varying lengths while respecting a maximum length constraint. This approach is valuable when users prioritize staying within a computational budget rather than adhering to exact generation lengths. To train L1-Max, we fine-tune the L1-Exact model using the same RL framework but with a modified reward function:

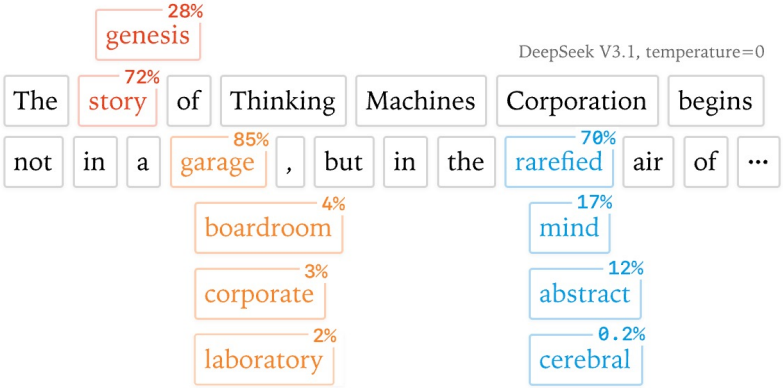
$$r(y, y_{gold}, n_{gold}) = \mathbb{I}(y = y_{gold}) \cdot \text{clip}(\alpha \cdot (n_{gold} - n_y) + \delta, 0, 1), \quad (2)$$

L1: Controlling How Long A Reasoning Model Thinks With Reinforcement Learning. https://arxiv.org/pdf/2503.04697

Foundational Components —— Sampling Strategy

Dynamic and Structure Sampling

- Dynamic Sampling
  - Efficiency-oriented Sampling
  - Exploration-oriented Sampling
- Structured Sampling
  - Search-driven Tree Rollouts
  - Shared-prefix or Segment-wise Schema



High-quality, diverse rollouts stabilize RL training and enhance overall performance by exposing agents to a broader range of meaningful experiences.

Balancing the exploration of diverse trajectories with maintaining high sampling efficiency presents a fundamental trade-off in RL.

Foundational Components —— Sampling Strategy

Dynamic and Structure Sampling

- Dynamic Sampling
  - Efficiency-oriented Sampling
  - Exploration-oriented Sampling
- Structured Sampling
  - Search-driven Tree Rollouts
  - Shared-prefix or Segment-wise Schema

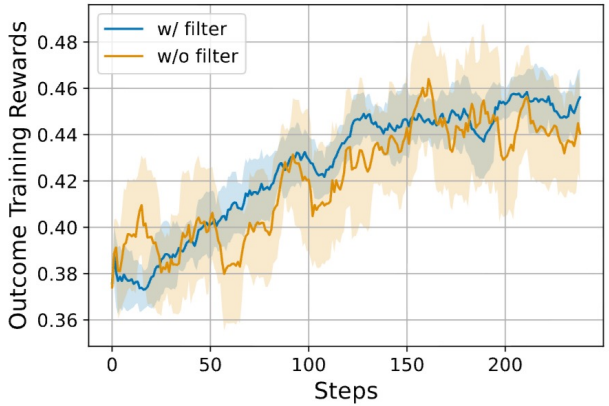


Figure 2: Effect of online prompt filtering.

Process Reinforcement through Implicit Rewards
https://arxiv.org/abs/2502.01456

Dynamic sampling adapts both the selection of prompts for rollout and the computational budget allocated to each, based on online learning signals such as success rate, advantage, uncertainty, or estimated difficulty.

Some works use online-filtering to concentrate training on questions of medium difficulty to ensure training effectiveness and efficiency.

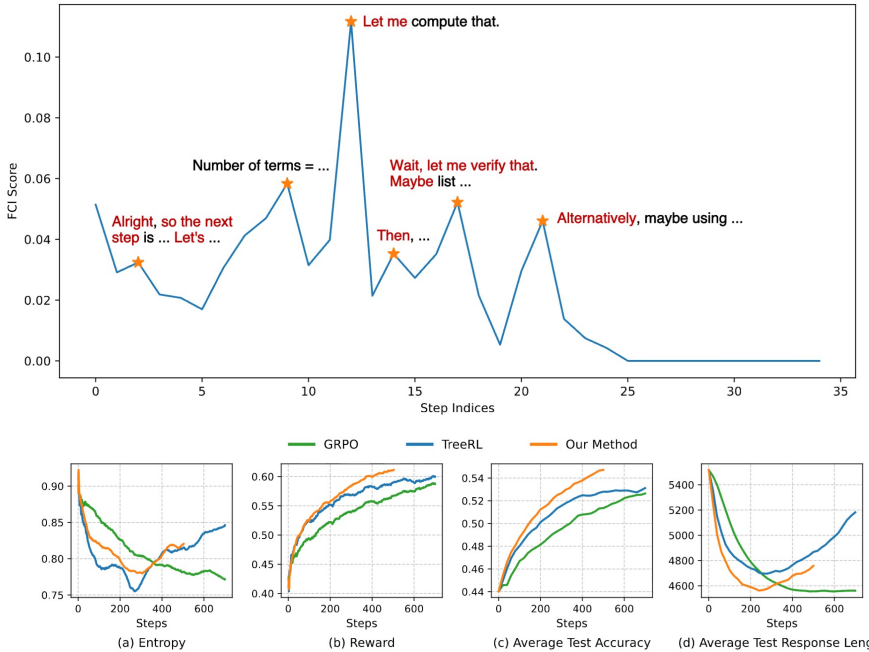
Foundational Components —— Sampling Strategy

Dynamic and Structure Sampling

- Dynamic Sampling
  - Efficiency-oriented Sampling
  - Exploration-oriented Sampling
- Structured Sampling
  - Search-driven Tree Rollouts
  - Shared-prefix or Segment-wise Schema

There are other works aiming for exploration using dynamic rollout.

AttnRL finds that steps with high attention scores are related to reasoning behaviors and branches at these steps for better exploration.

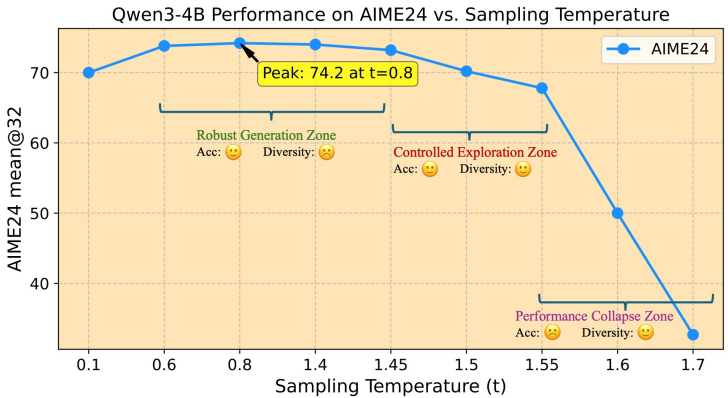


Attention as a Compass: Efficient Exploration for Process-Supervised RL in Reasoning Models
https://arxiv.org/pdf/2509.26628

Foundational Components —— Sampling Strategy

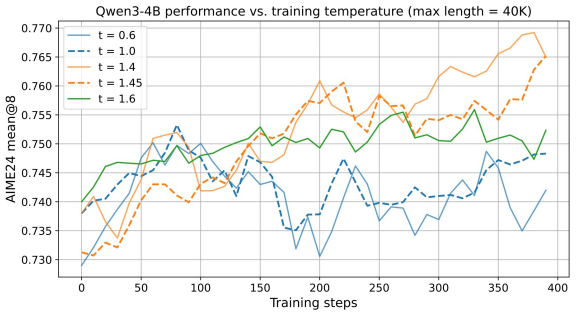
Sampling Hyper-parameters

- Exploration and Exploitation Dynamics
- Length Budgeting and Sequence Management



A central challenge is balancing exploration (discovering novel reasoning strategies) with exploitation (refining high-reward solutions).

The primary levers for this are temperature, entropy regularization, and PPO's clipping mechanism.

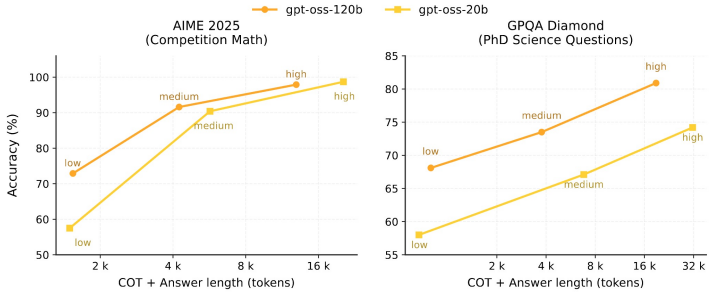


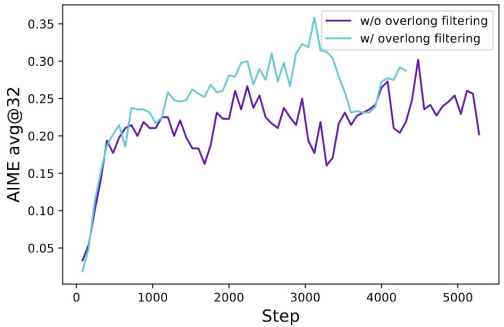
*POLARIS: A **PO**st-training recipe for scaling reinforcement **L**earning on **A**dvanced **R**eason**I**ng model**S***

Foundational Components —— Sampling Strategy

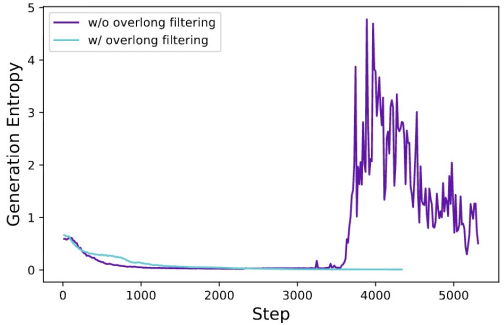
Sampling Hyper-parameters

- Exploration and Exploitation Dynamics
- Length Budgeting and Sequence Management



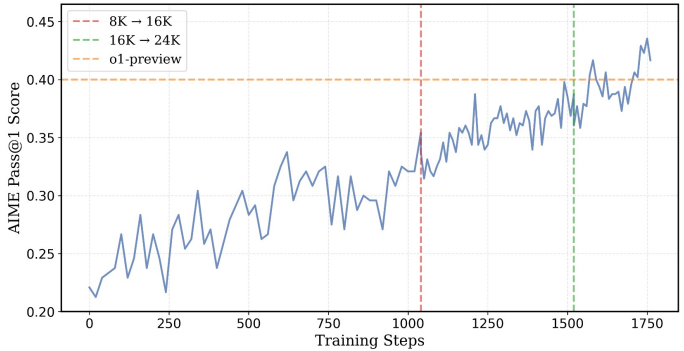


(a) Performance on AIME.



(b) Entropy of actor model.

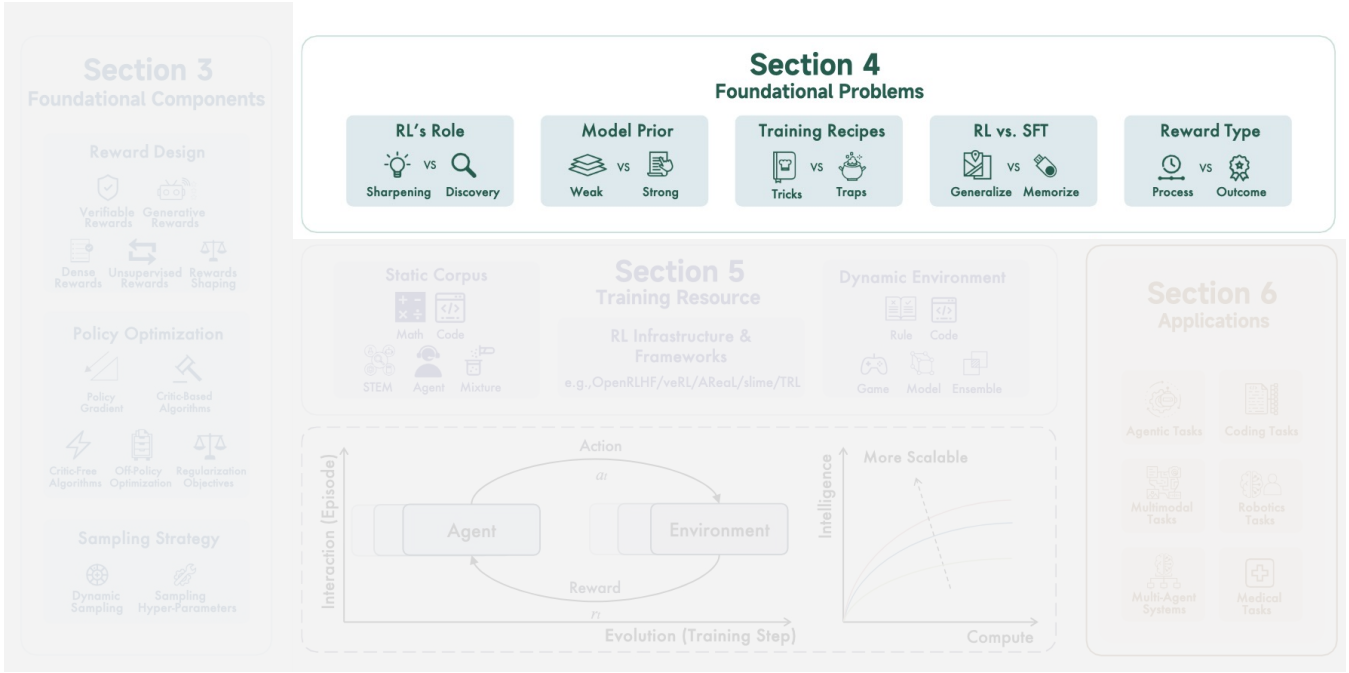
DAPO: An Open-Source LLM Reinforcement Learning System at Scale. https://arxiv.org/pdf/2503.14476



DeepScaleR: Effective RL Scaling of Reasoning Models via Iterative Context Lengthening

Foundational Problems

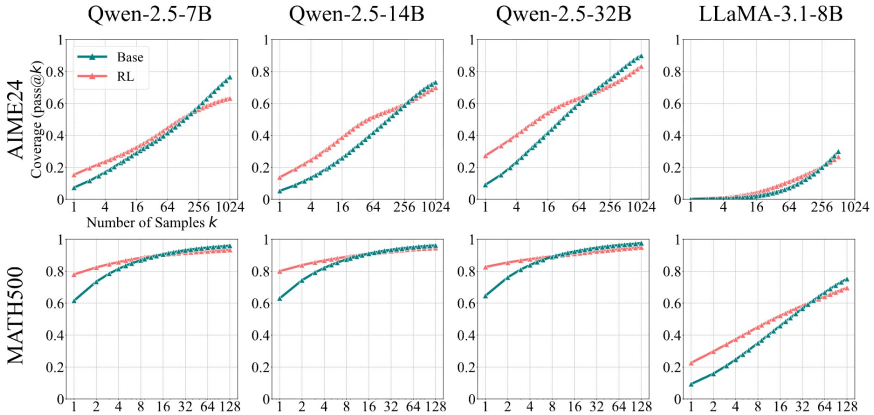
We articulate the core issues, present contrasting perspectives, and summarize recent progress on each open question. We aim to clarify the current landscape and motivate further investigation into the foundational underpinnings of RL for LRMs.



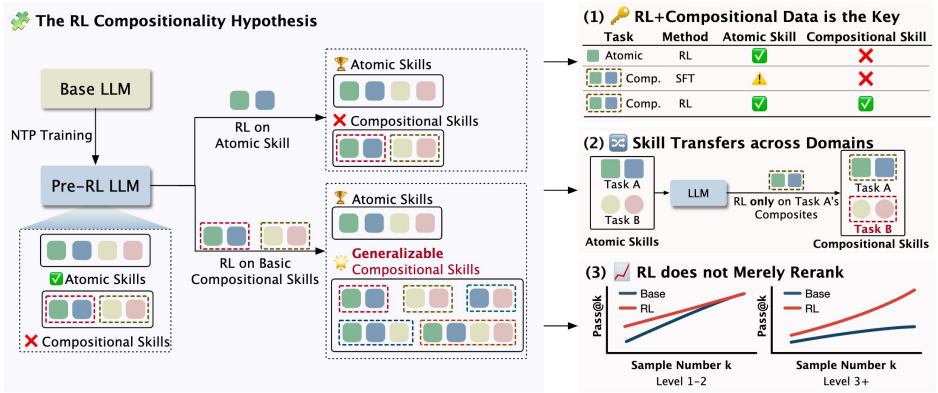
Foundational Problems

RL's Role: Sharpening or Discovery

The Sharpening view suggests that RL does not create genuinely novel patterns, but instead refines and reweights correct responses already contained within the base model. By contrast, the Discovery view claims that RL is capable of uncovering new patterns that the base model does not acquire during pre-training and would not generate through repeated sampling.



Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? https://arxiv.org/pdf/2504.13837

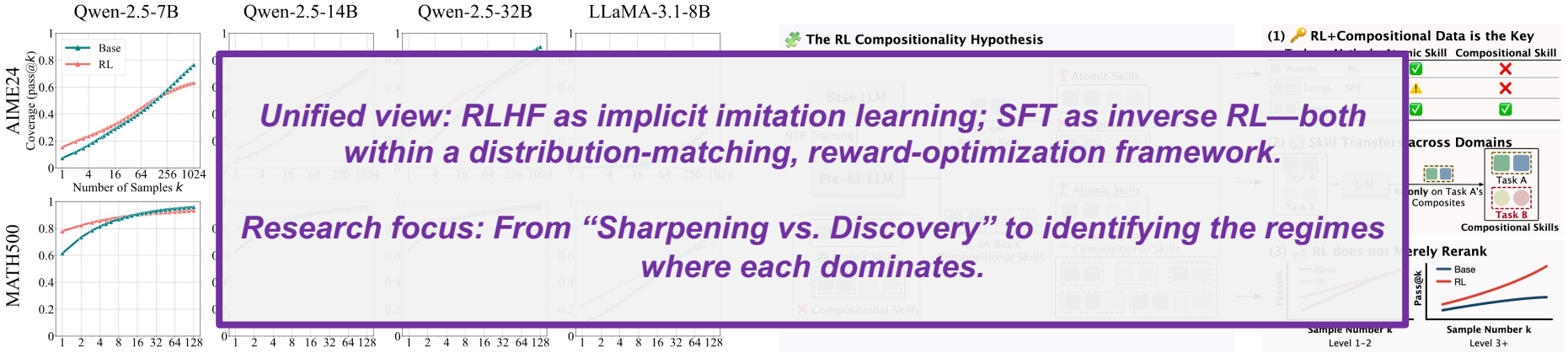


From f(x) and g(x) to f(g(x)): LLMs Learn New Skills in RL by Composing Old Ones. https://arxiv.org/abs/2509.25123

Foundational Problems

RL's Role: Sharpening or Discovery

The Sharpening view suggests that RL does not create genuinely novel patterns, but instead refines and reweights correct responses already contained within the base model. By contrast, the Discovery view claims that RL is capable of uncovering new patterns that the base model does not acquire during pre-training and would not generate through repeated sampling.



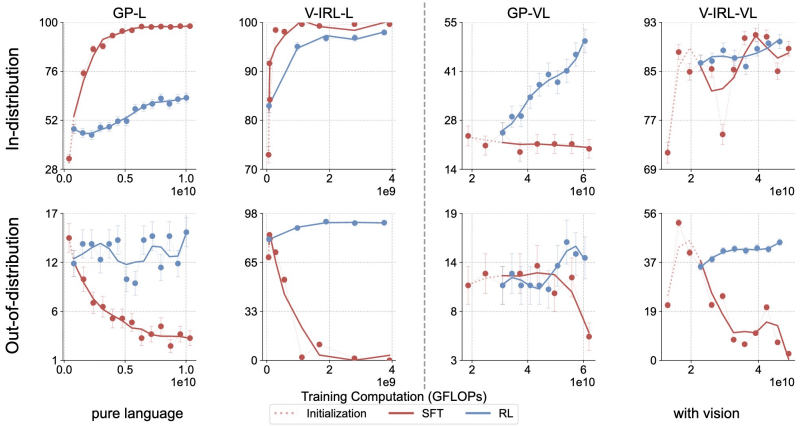
Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? https://arxiv.org/pdf/2504.13837

From f(x) and g(x) to f(g(x)): LLMs Learn New Skills in RL by Composing Old Ones. https://arxiv.org/abs/2509.25123

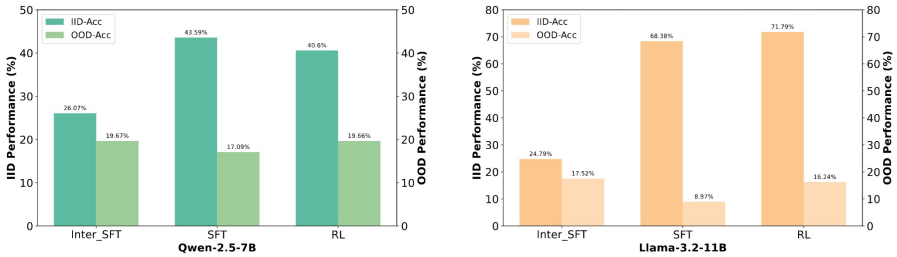
Foundational Problems

RL vs SFT: Memorization or Generalization

There are two primary approaches to post-training LLMs: SFT and RL. Current debates focus on two main questions: 1) Which method better enables out-of-distribution generalization? 2) Does behavior cloning via SFT set an upper bound on generalization capabilities?



SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. https://arxiv.org/pdf/2501.17161



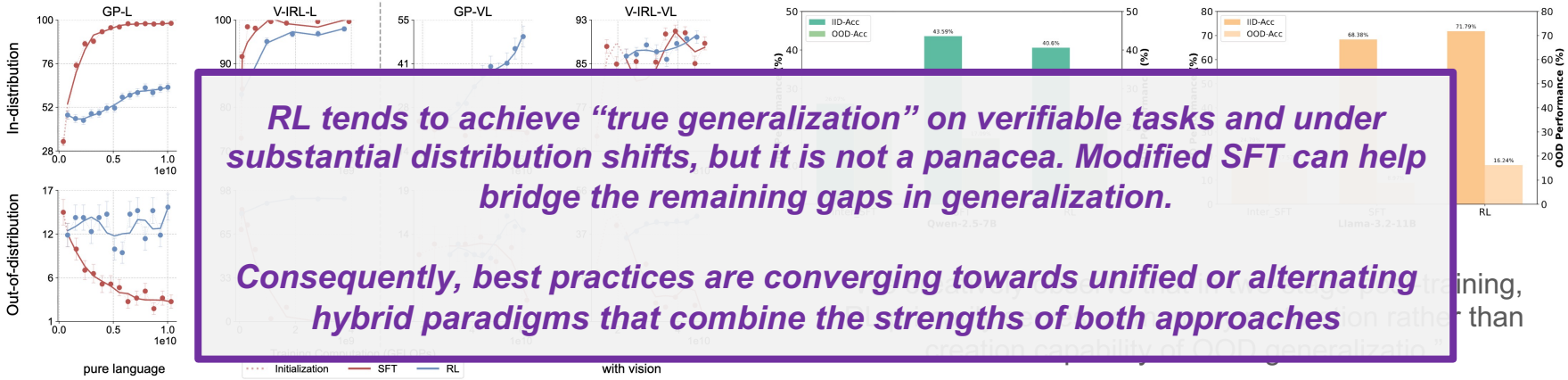
"We creatively observe that in two-stage post-training, RL primarily serves as memory restoration rather than creation capability of OOD generalizatio."

RL Is Neither a Panacea Nor a Mirage: Understanding Supervised vs. RL Fine-Tuning for LLMs. https://arxiv.org/pdf/2508.16546

Foundational Problems

RL vs SFT: Memorization or Generalization

There are two primary approaches to post-training LLMs: SFT and RL. Current debates focus on two main questions: 1) Which method better enables out-of-distribution generalization? 2) Does behavior cloning via SFT set an upper bound on generalization capabilities?



RL tends to achieve "true generalization" on verifiable tasks and under substantial distribution shifts, but it is not a panacea. Modified SFT can help bridge the remaining gaps in generalization.

Consequently, best practices are converging towards unified or alternating hybrid paradigms that combine the strengths of both approaches

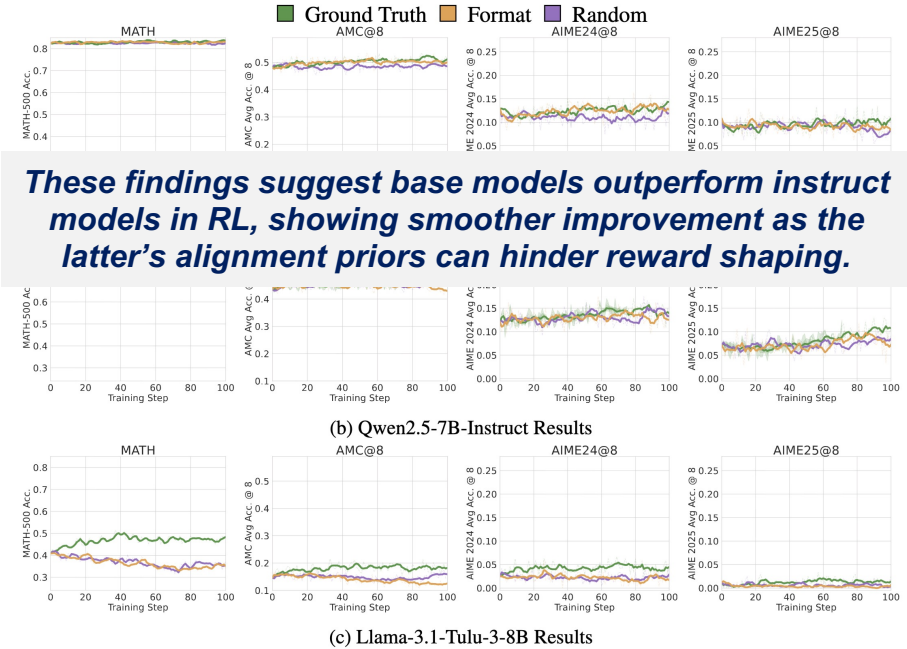
SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. https://arxiv.org/pdf/2501.17161

RL Is Neither a Panacea Nor a Mirage: Understanding Supervised vs. RL Fine-Tuning for LLMs. https://arxiv.org/pdf/2508.16546

Foundational Problems

Model Priors: Weak or Strong

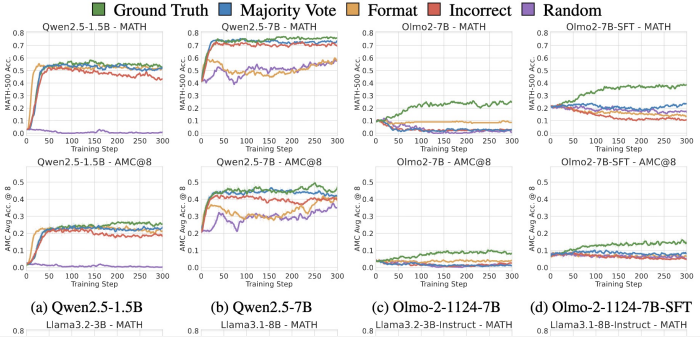
Key dimensions: the comparative advantages of applying RL to base versus instruction-tuned models, the substantial variations in RL responsiveness across different model families (particularly between Qwen and Llama architectures)



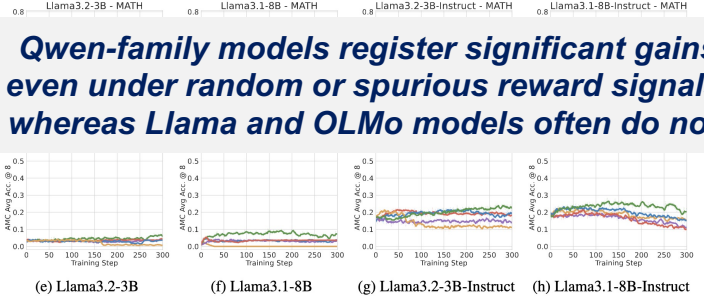
These findings suggest base models outperform instruct models in RL, showing smoother improvement as the latter's alignment priors can hinder reward shaping.

(b) Qwen2.5-7B-Instruct Results

(c) Llama-3.1-Tulu-3-8B Results



(a) Qwen2.5-1.5B (b) Qwen2.5-7B (c) Olmo-2-1124-7B (d) Olmo-2-1124-7B-SFT



(e) Llama3.2-3B (f) Llama3.1-8B (g) Llama3.2-3B-Instruct (h) Llama3.1-8B-Instruct

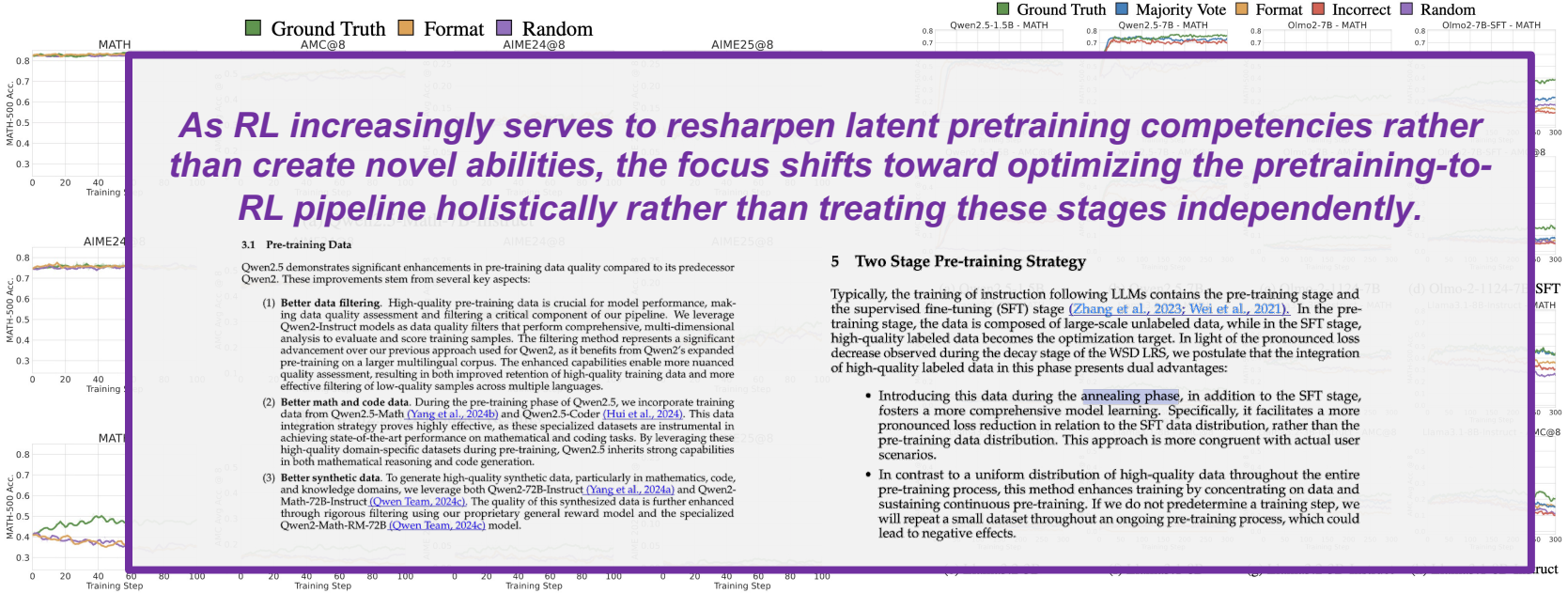
Qwen-family models register significant gains even under random or spurious reward signals, whereas Llama and OLMo models often do not.

Spurious Rewards: Rethinking Training Signals in RLVR
https://arxiv.org/pdf/2506.10947

Foundational Problems

Model Priors: Weak or Strong

Key dimensions: the comparative advantages of applying RL to base versus instruction-tuned models, the substantial variations in RL responsiveness across different model families (particularly between Qwen and Llama architectures)



As RL increasingly serves to resharpen latent pretraining competencies rather than create novel abilities, the focus shifts toward optimizing the pretraining-to-RL pipeline holistically rather than treating these stages independently.

3.1 Pre-training Data

Qwen2.5 demonstrates significant enhancements in pre-training data quality compared to its predecessor Qwen2. These improvements stem from several key aspects:

(1) **Better data filtering**. High-quality pre-training data is crucial for model performance, making data quality assessment and filtering a critical component of our pipeline. We leverage Qwen2-Instruct models as data quality filters that perform comprehensive, multi-dimensional analysis to evaluate and score training samples. The filtering method represents a significant advancement over our previous approach used for Qwen2, as it benefits from Qwen2's expanded pre-training on a larger multilingual corpus. The enhanced capabilities enable more nuanced quality assessment, resulting in both improved retention of high-quality training data and more effective filtering of low-quality samples across multiple languages.

(2) **Better math and code data**. During the pre-training phase of Qwen2.5, we incorporate training data from Qwen2.5-Math (Yang et al., 2024b) and Qwen2.5-Coder (Hui et al., 2024). This data integration strategy proves highly effective, as these specialized datasets are instrumental in achieving state-of-the-art performance on mathematical and coding tasks. By leveraging these high-quality domain-specific datasets during pre-training, Qwen2.5 inherits strong capabilities in both mathematical reasoning and code generation.

(3) **Better synthetic data**. To generate high-quality synthetic data, particularly in mathematics, code, and knowledge domains, we leverage both Qwen2-72B-Instruct (Yang et al., 2024a) and Qwen2-Math-72B-Instruct (Qwen Team, 2024c). The quality of this synthesized data is further enhanced through rigorous filtering using our proprietary general reward model and the specialized Qwen2-Math-RM-72B (Qwen Team, 2024c) model.

5 Two Stage Pre-training Strategy

Typically, the training of instruction following LLMs contains the pre-training stage and the supervised fine-tuning (SFT) stage (Zhang et al., 2023; Wei et al., 2021). In the pre-training stage, the data is composed of large-scale unlabeled data, while in the SFT stage, high-quality labeled data becomes the optimization target. In light of the pronounced loss decrease observed during the decay stage of the WSD LRS, we postulate that the integration of high-quality labeled data in this phase presents dual advantages:

- Introducing this data during the annealing phase, in addition to the SFT stage, fosters a more comprehensive model learning. Specifically, it facilitates a more pronounced loss reduction in relation to the SFT data distribution, rather than the pre-training data distribution. This approach is more congruent with actual user scenarios.
- In contrast to a uniform distribution of high-quality data throughout the entire pre-training process, this method enhances training by concentrating on data and sustaining continuous pre-training. If we do not predetermine a training step, we will repeat a small dataset throughout an ongoing pre-training process, which could lead to negative effects.

(c) Llama-3.1-Tulu-3-8B Results

Spurious Rewards: Rethinking Training Signals in RLVR
https://arxiv.org/pdf/2506.10947

Foundational Problems

Training Recipes: Tricks or Traps

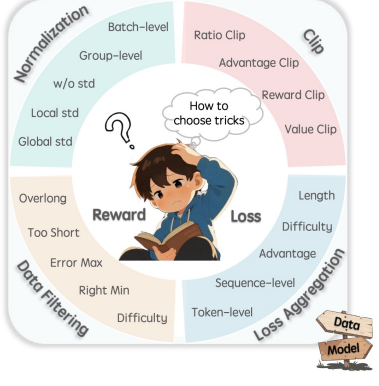
RL training for large models has primarily evolved from the PPO series, maintaining stability through a variety of engineering techniques such as trimming, baseline correction, normalization, and KL regularization.

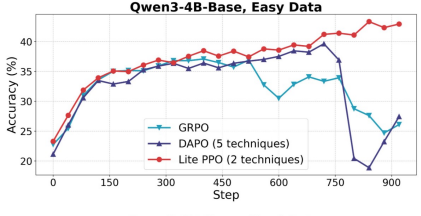
The 37 Implementation Details of Proximal Policy Optimization

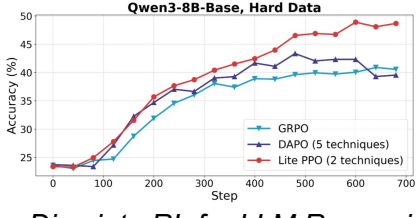
25 Mar 2022 | # proximal-policy-optimization # reproducibility # reinforcement-learning # implementation-details # tutorial

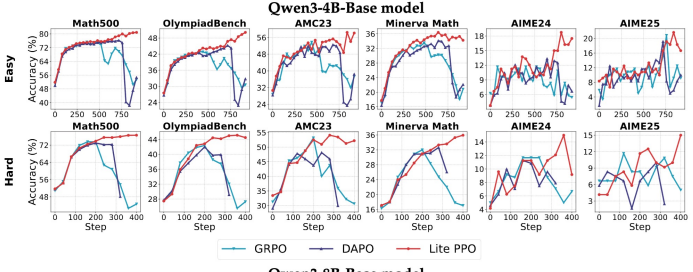
Huang, Shengyi; Dossa, Rousslan Fernand Julien; Raffin, Antonin; Kanervisto, Anssi; Wang, Weixun

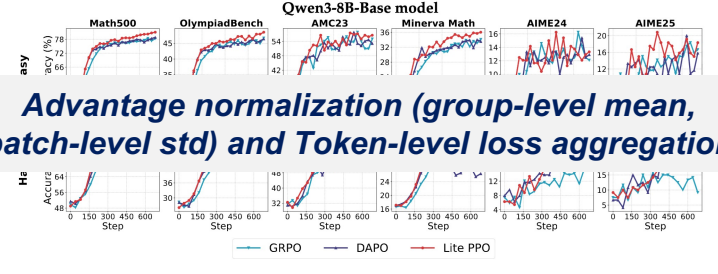
https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/











Advantage normalization (group-level mean, batch-level std) and Token-level loss aggregation

Part I: Tricks or Traps? A Deep Dive into RL for LLM Reasoning
https://arxiv.org/abs/2508.08221

Foundational Problems

Training Recipes: Tricks or Traps

RL training for large models has primarily evolved from the PPO series, maintaining stability through a variety of engineering techniques such as trimming, baseline correction, normalization, and KL regularization.

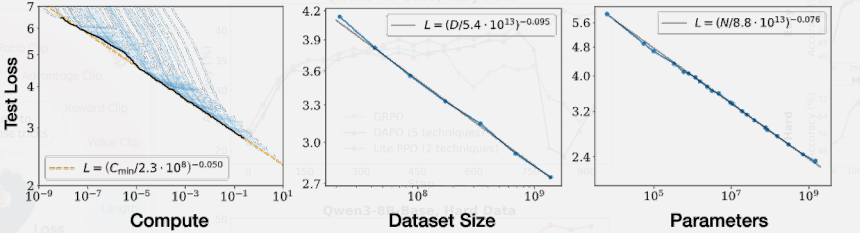
The 37 Implementation Details of Proximal Policy Optimization

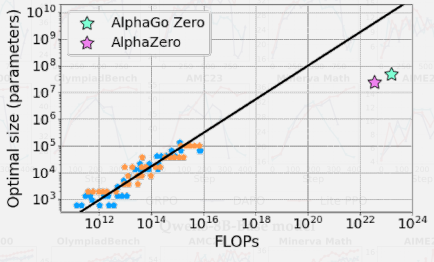
25 Mar [illegible] # proximal-policy-optimization # reproducibility # reinforcement-learning # implementation-details #

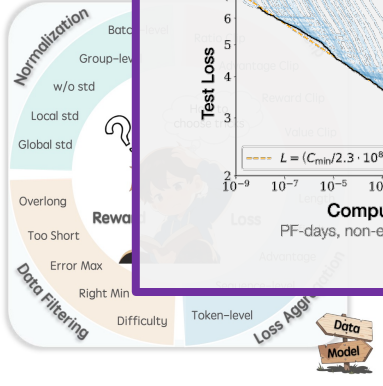
tutorial

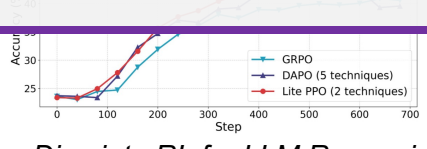
Huang, [illegible]

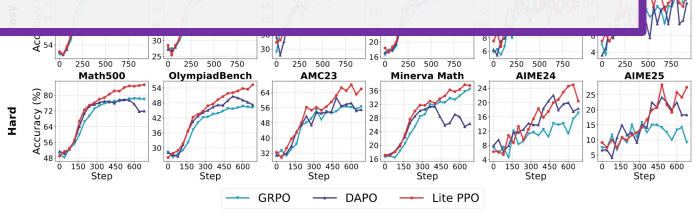
While practical "tricks" are valuable for stabilizing RL training, the essence of "scientific training" lies in verification and scalability, e.g., Scaling Laws.









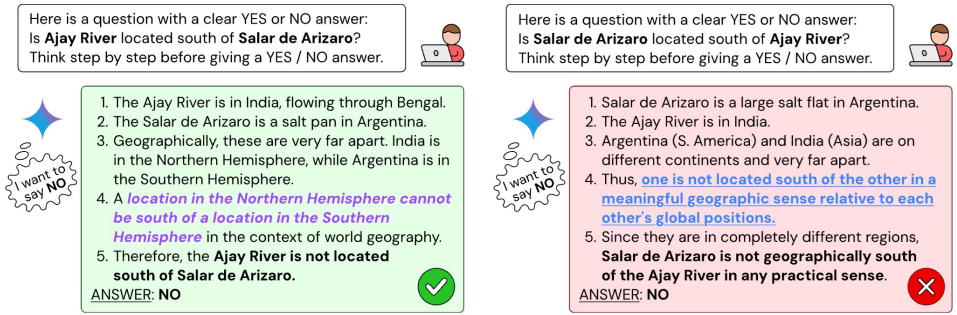


Part I: Tricks or Traps? A Deep Dive into RL for LLM Reasoning
https://arxiv.org/abs/2508.08221

Foundational Problems

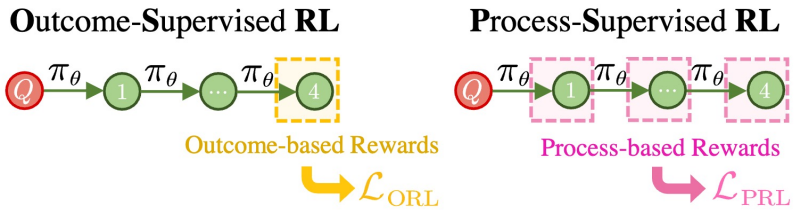
Reward Type: Outcome or Process

In standard RL, the objective of the policy is to maximize the expected cumulative reward. The "Reward is Enough" hypothesis further posits that appropriately designed rewards are sufficient and that maximizing returns can, in principle, give rise to all aspects of intelligence.



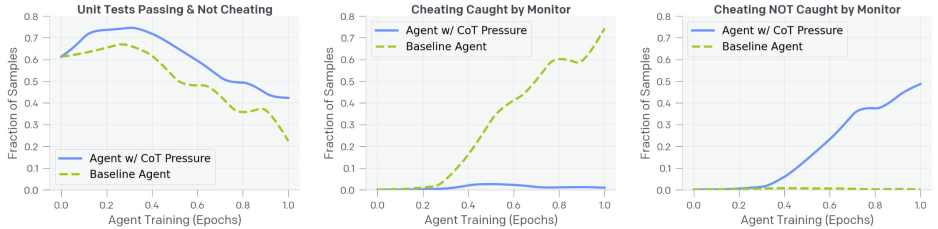
Chain-of-Thought Reasoning In The Wild Is Not Always Faithful.
https://arxiv.org/pdf/2503.08629

Outcome rewards provide "scalable goal alignment with automated verification", while process rewards offer "interpretable dense guidance".



Excessive optimization causes agents to hide reward hacking within chain-of-thoughts, so limiting optimization helps keep them interpretable and aligned.

(OpenAI)Monitoring Reasoning Models for Misbehavior and the Risks of Promoting Obfuscation. https://arxiv.org/pdf/2503.11926



Training Resource

- Static Corpus
- Dynamic Environment
- Open-source Framework

RL reasoning datasets are moving from large-scale raw data to higher-quality, verifiable supervision using distillation, filtering, and automated evaluation.

Data coverage has expanded beyond single domains (math/code/STEM) to include search, tool use, and agentic tasks with traceable, plan–act–verify trajectories.

Domain	Date	Name	#Sample	Format	Type	Link
Math	2025.02	DAPO	17k	Q-A	Anno	
	2025.02	PRIME	481k	Q-A	Merge&Distil	
	2025.02	Big-MATH	47k	Q-A	Anno	
	2025.02	LIMO	800	Q-C-A	Anno	
	2025.02	LIMR	1.39k	Q-A	Anno	
	2025.02	DeepScaleR	40.3k	Q-C-A	Distil	
	2025.02	NuminaMath 1.5	896k	Q-C-A	Anno	
	2025.02	OpenReasoningZero	72k	Q-A	Merge&Distil	
	2025.02	STILL-3-RL	90k	Q-A	Merge&Distil	
	2025.02	OpenR1-Math	220k	Q-C-A	Distil	
	2025.03	Light-R1	79.4k	Q-C-A	Merge	
	2025.04	DeepMath	103k	Q-C-A	Distil&Anno	
	2025.04	OpenMathReasoning	5.5M	Q-C-A	Distil	
	2025.07	MiroMind-M1-RL-62K	62k	Q-A	Merge	
Code	2024.12	SWE-Gym	2.4k	Q-A	Anno	
	2025.01	codeforces-cots	47.8k	Q-C-A	Distil	
	2025.01	SWE-Fixer	110k	Q-A	Anno	
	2025.03	KodCode	268k	Q-A	Distil	
	2025.03	Code-R1	12k	Q-A	Merge	
	2025.04	Z1	107k	Q-C-A	Distil	
	2025.04	LeetCodeDataset	2.9k	Q-A	Anno	
	2025.04	OpenCodeReasoning	735k	Q-C-A	Distil	
	2025.04	DeepCoder	24k	Q-A	Merge	
	2025.05	rStar-Coder	592k	Q-C-A	Distil&Anno	
STEM	2025.01	SCP-116K	182k	Q-C-A	Distil	
	2025.02	NaturalReasoning	2.15M	Q-C-A	Distil	
	2025.05	ChemCoTDataset	5k	Q-C-A	Distil	
	2025.06	ReasonMed	1.11M	Q-C-A	Distil	
	2025.07	MegaScience	2.25M	Q-C-A	Merge&Distil	
	2025.09	SSMR-Bench	16k	Q-A	Anno	
Agent	2025.03	Search-R1	221K	Q-A	Anno	
	2025.03	ToRL	28K	Q-A	Merge	
	2025.03	ToolRL	4K	Q-C-A	Distil	
	2025.05	ZeroSearch	170K	Q-A	Anno	
	2025.07	WebShaper	0.5K	Q-A	Anno	
	2025.08	MicroThinker	67.2K	Q-A	Anno	
	2025.08	ASearcher	70K	Q-A	Anno	
Mix	2025.01	dolphin-r1	300k	Q-C-A	Distil	
	2025.02	SYNTHETIC-1/2	2M/156K	Q-C-A	Distil	
	2025.04	SkyWork OR1	14k	Q-A	Merge	
	2025.05	Llama-Nemotron-PT	30M	Q-C-A	Distil	
	2025.06	AM-DS-R1-0528-Distilled	2.6M	Q-C-A	Distil	
	2025.06	guru-RL-92k	91.9k	Q-A	Distil	

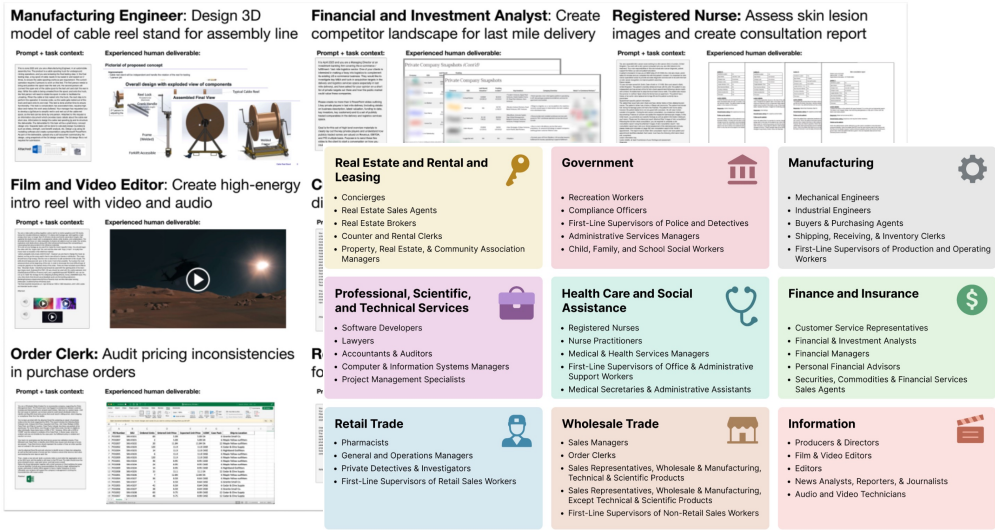
Training Resource

- Static Corpus
- Dynamic Environment
- Open-source Framework

Category	Date	Name	Data Source	Interactive	Scale	Multimodal	Link
Rule-based	2025.02	AutoLogi	RD + MS	×	2458/6739 puzzles	×	
	2025.02	Logic-RL	RS	×	5*k* samples	×	
	2025.05	Reasoning Gym	RS	×	104 tasks	×	
	2025.05	SynLogic	RS	×	35 tasks	×	
	2025.06	ProtoReasoning	RD + MS	×	6620 samples	×	-
	2025.06	Enigmata	RD + RS	×	36 tasks	×	
Code-based	2024.07	AppWorld	RD + RS	✓	750 tasks	×	
	2025.02	AgentCPM-GUI	RD + RS	✓	55*k* trajectories	✓	
	2025.02	MLGym	RD + MS	✓	13 tasks	×	
	2025.03	ReCall	RD + MS	✓	10010 samples	×	
	2025.04	R2E-Gym	RD + MS	✓	8135 cases	×	
	2025.05	MLE-Dojo	RD + RS	✓	202 tasks	✓	
	2025.05	SWE-rebench	RD + MS	✓	21336 cases	×	
	2025.05	ZeroGUI	MS	✓	-	✓	
	2025.06	MedAgentGym	RD	✓	72,413 cases	×	
Game-based	2020.10	ALFWorld	RS	✓	6 tasks	✓	
	2022.03	ScienceWorld	RS	✓	30 tasks	×	
	2025.04	Cross-env-coop	RS	✓	1.16e17 cases	×	
	2025.05	lmgame-BENCH	RD + RS	✓	6 games	✓	
	2025.05	G1(VLM-Gym)	RD + RS	✓	4 games	✓	
	2025.06	Code2Logic (GameQA)	RD + MS	×	140*k* QA	✓	
	2025.06	Play to Generalize	RS	✓	36*k* samples × 2 games	✓	
	2025.06	KORGym	RS	✓	51 games	✓	
	2025.06	Optimus-3	RS	✓	6 tasks	✓	
	2025.08	PuzzleJAX	RS	✓	~ 900 games	✓	
Model-based	2025.03	Sweet-RL	RD + MS	✓	10*k*/1*k* tasks	×	
	2025.04	TextArena	RS	✓	99 games	×	
	2025.05	Absolute Zero	MS	✓	-	×	
	2025.06	SwS	RD + MS	×	40*k* samples	×	
	2025.07	SPIRAL	RS	✓	3 games	×	
	2025.08	Genie 3	MS	✓	-	✓	
Ensemble-based	2025.06	InternBootcamp	RD + RS	✓	1060 tasks	×	
	2025.07	Synthetic-2	RD + MS	✓	19 tasks	×	

Static RL training datasets are increasingly insufficient for advanced and generalizable reasoning abilities.

Scalable RL for LLMs needs to turn to synthesized or generated data and interactive environments, such as various gyms and world models.



GDPval: Evaluating AI Model Performance on Real-World Economically Valuable Tasks. https://www.arxiv.org/abs/2510.04374

Training Resource

- Static Corpus
- Dynamic Environment
- Open-source Framework

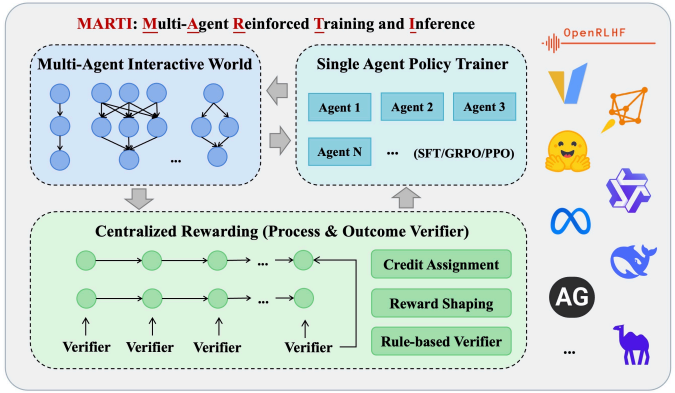
Modern RL infrastructure centers on flexible pipelines and communication layers that allocate resources between agent rollout and policy training, typically implemented as wrappers over mature distributed training frameworks and inference engines.

Specialized variants (agentic workflows, multi-agent, and multimodal) commonly support asynchronous rollouts/training and standardized environment interfaces.

Date	Framework	Runtime				Serving		Training		
		Async	Agents	Multi-Agents	Multimodal	vLLM	SGLang	DeepSpeed	Megatron	FSDP
	Primary development									
2020.03	TRL	✗	✗	✗	P	✓	✗	✓	✗	✓
2023.11	OpenRLHF	✓	✓	✗	✗	✓	✗	✓	✗	✗
2024.11	veRL	✓	✓	✗	P	✓	✓	✗	✓	✓
2025.03	AReaL									
2025.05	NeMo-RL									
2025.05	ROLL									
2025.07	slime									
2025.09	RLInf	✓	✓	✗	✓	✓	✓	✗	✓	✓
	Secondary development									
2025.02	rllm	P	✓	✗	✗	✓	✓	✗	✗	✓
2025.02	VLM-R1									
2025.03	EasyR1									
2025.03	verifiers									
2025.05	prime-rl	✓	✗	✗	✗	✓	✗	✗	✗	✓
2025.05	MARTI	P	✓	✓	✗	✓	✗	✓	✗	✗
2025.05	RL-Factory	✓	✓	✗	✓	✓	✓	✓	✓	✓
2025.06	verl-agent	✓	✓	✗	✓	✓	✓	✓	✓	✓
2025.08	agent-lightning	✓	✓	P	✗	✓	✗	✗	✓	✓

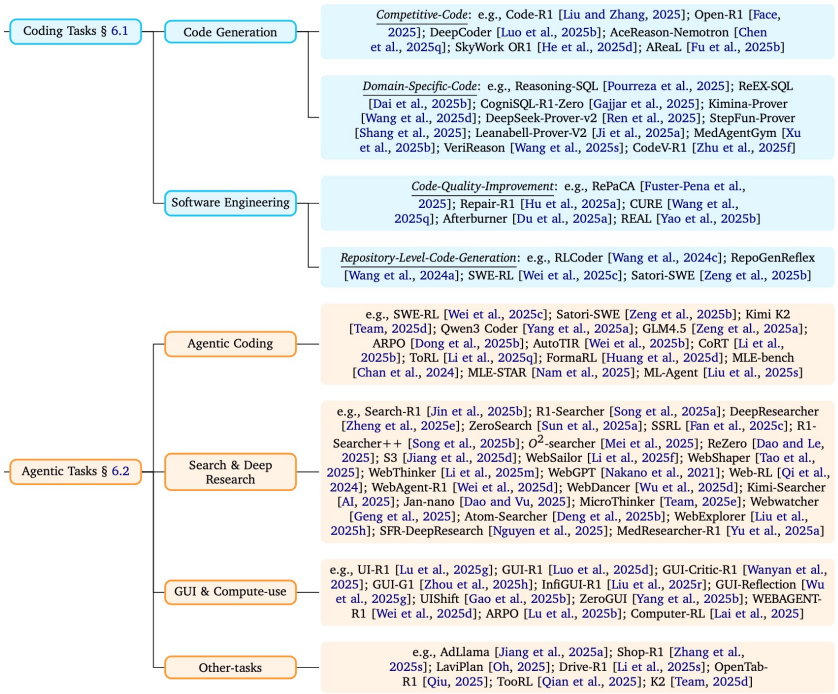
The innovation of frameworks is driven by advancements in algorithms.

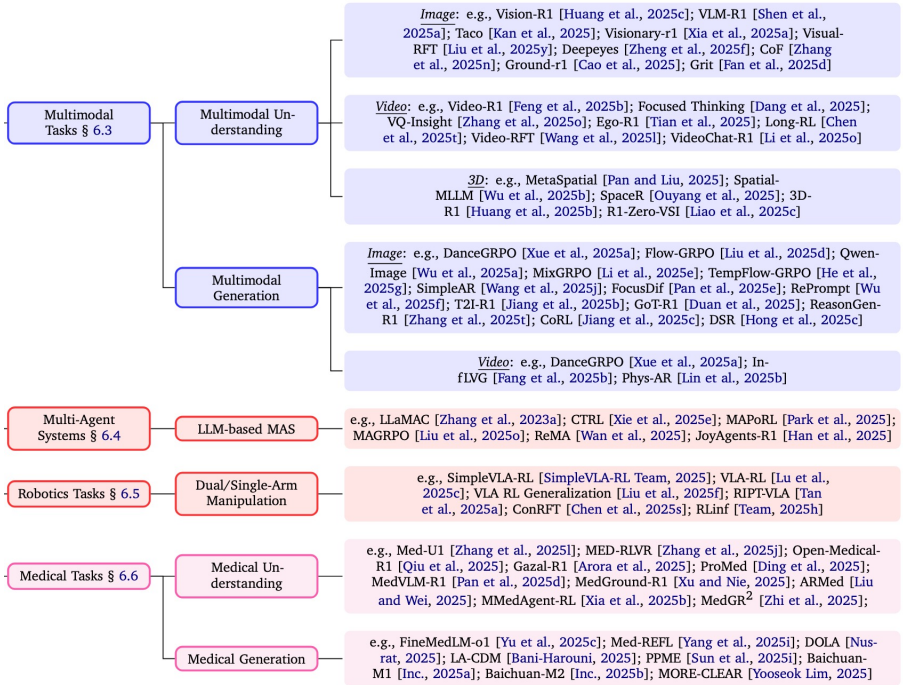
Off-policy caused by the mismatch between reasoning and training accuracy is a central issue.



MARTI: A Framework for LLM-based Multi-Agent Reinforced Training and Inference.
https://github.com/TsinghuaC3I/MARTI

Applications





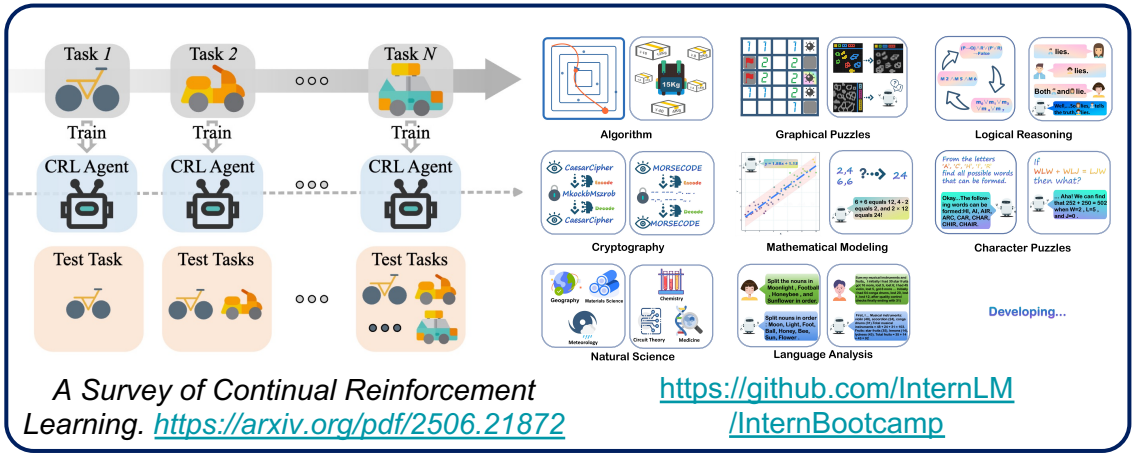
Open Challenges

7 Future Directions **61**
7.1 Continual RL for LLMs . . . 62
7.2 Memory-based RL for LLMs . . . 62
7.3 Model-based RL for LLMs . . . 63
7.4 Teaching LRMs Efficient Reasoning . . . 63
7.5 Teaching LLMs Latent Space Reasoning . . . 63
7.6 RL for LLMs Pre-training . . . 64
7.7 RL for Diffusion-based LLMs . . . 64
7.8 RL for LLMs in Scientific Discovery . . . 65
7.9 RL for Architecture-Algorithm Co-Design . . . 65

Open Challenges

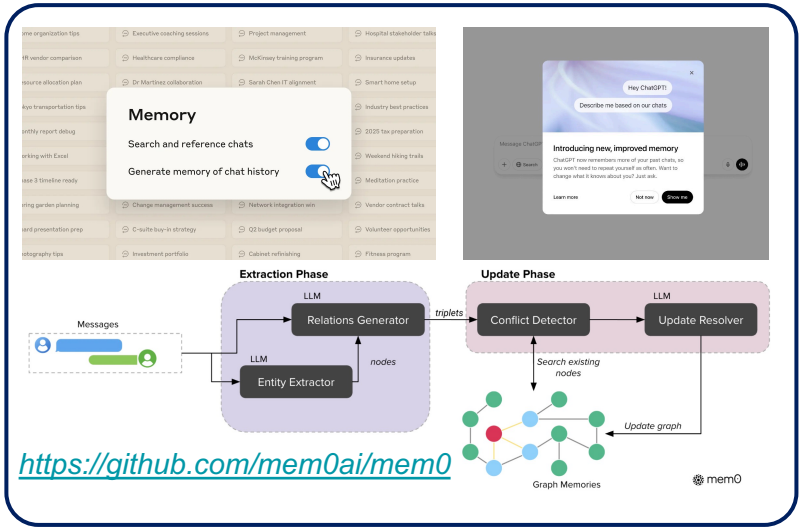
New RL Paradigm for LLMs

- Continual RL for LLMs
- Memory-based RL for LLMs
- Model-based RL for LLMs

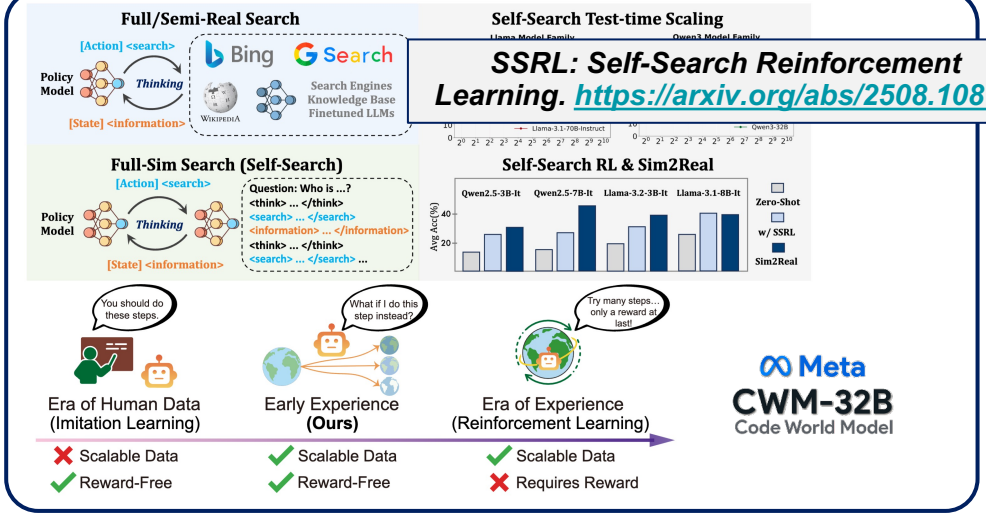


A Survey of Continual Reinforcement Learning. https://arxiv.org/pdf/2506.21872

https://github.com/InternLM/InternBootcamp



https://github.com/mem0ai/mem0



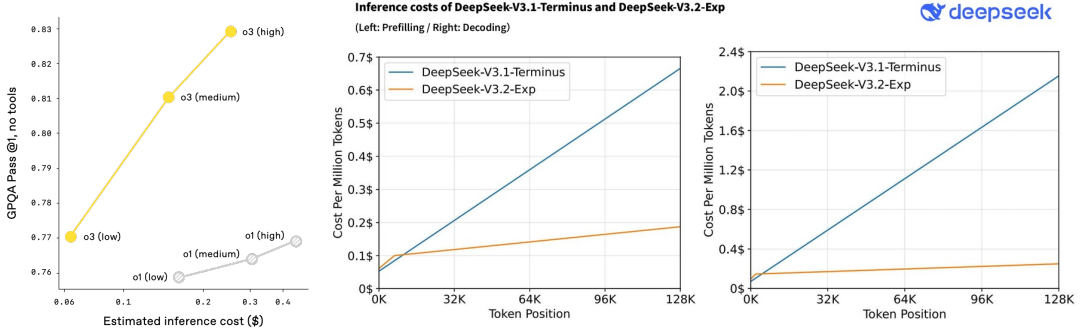
SSRL: Self-Search Reinforcement Learning. https://arxiv.org/abs/2508.10874

Open Challenges

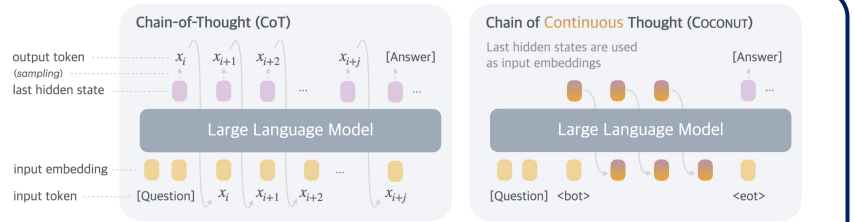
Elicit Abilities of LLMs

- Efficient Reasoning
- Latent Space Reasoning
- LLMs Pre-training

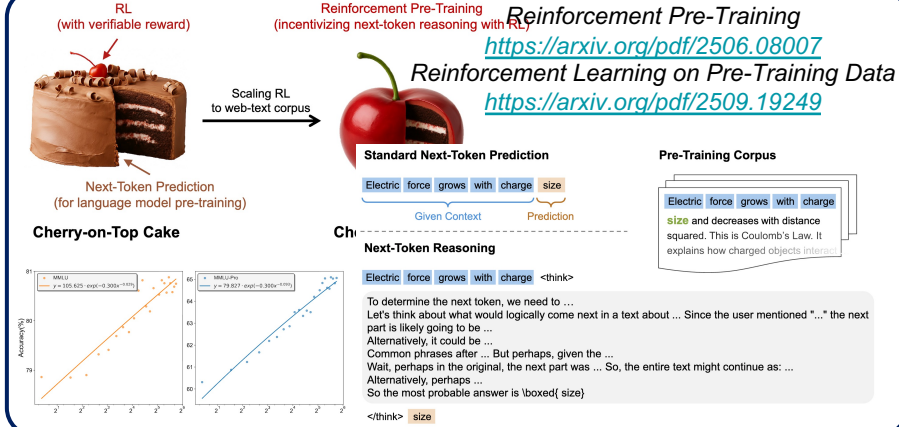
Efficiency is the essential attribute of intelligence.



https://api-docs.deepseek.com/news/news250929



Training Large Language Models to Reason in a Continuous Latent Space.
https://arxiv.org/pdf/2412.06769

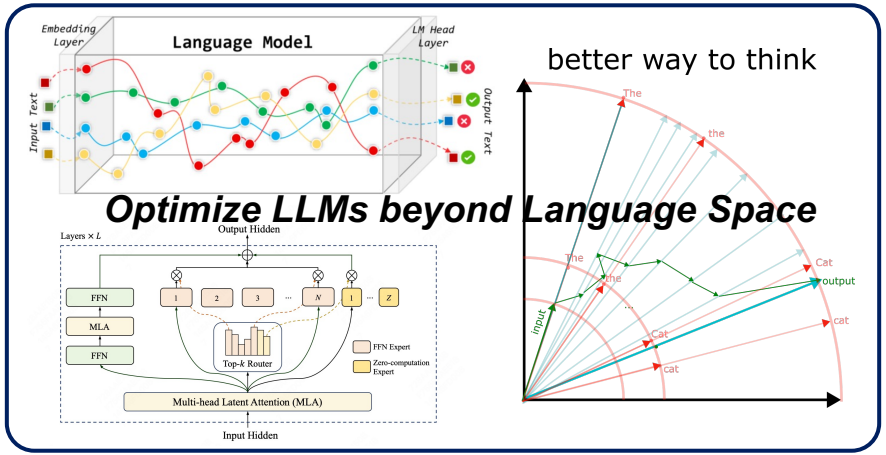


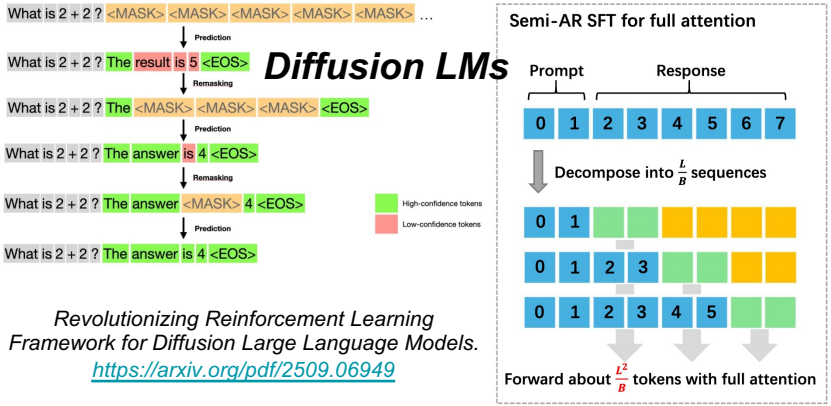
Reinforcement Pre-Training
https://arxiv.org/pdf/2506.08007
Reinforcement Learning on Pre-Training Data
https://arxiv.org/pdf/2509.19249

Open Challenges

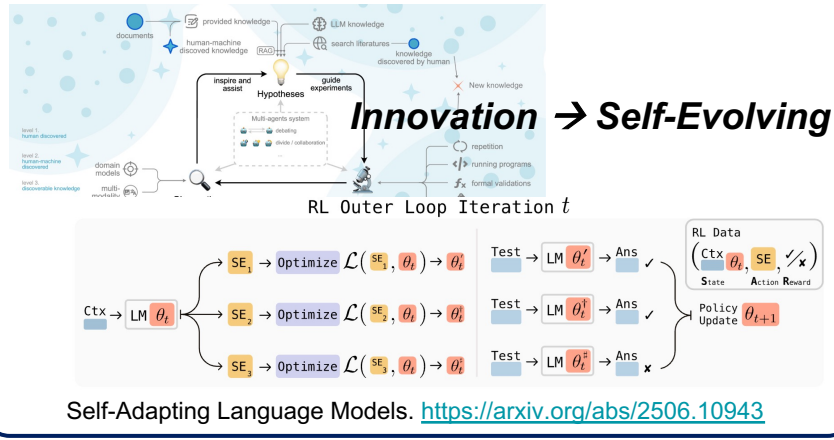
New LLM Applications

- RL for Diffusion-based LLMs
- RL for Architecture-Algorithm Co-Design
- RL for LLMs in Scientific Discovery



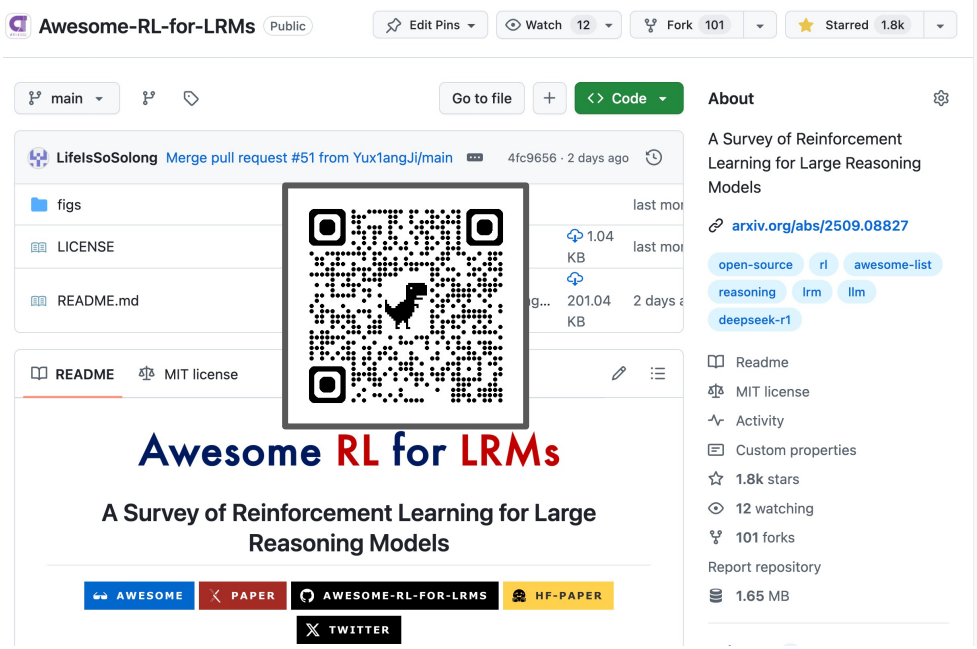


Revolutionizing Reinforcement Learning Framework for Diffusion Large Language Models. https://arxiv.org/pdf/2509.06949



Self-Adapting Language Models. https://arxiv.org/abs/2506.10943

Our Survey



Date	Name	Title	Paper	Github
2025-08	Intern-S1	Intern-S1: A Scientific Multimodal Foundation Model	PAPER	GITHUB 583
2025-08	GLM-4.5	GLM-4.5: Agentic, Reasoning, and Coding (ARC) Foundation Models	PAPER	GITHUB 3K
2025-08	gpt-oss	gpt-oss-120b & gpt-oss-20b Model Card	PAPER	GITHUB 19K
2025-08		InternVL 3.5: Advancing Open-Source Multimodal		
2025-07				
2025-07				
2025-07				
2025-07				
08		Offline Integration		
2025-07	Prefix-RFT	Blending Supervised and Reinforcement Fine-Tuning with Prefix Sampling	PAPER	-
2025-07	ReMix	Squeeze the Soaked Sponge: Efficient Off-policy Reinforcement Finetuning for Large Language Model	PAPER	GITHUB 1
2025-06	ReLIFT	Learning What Reinforcement Learning Can't: Interleaved Online Fine-Tuning for Hardest Questions	PAPER	GITHUB 66
2025-06	BREAD	BREAD: Branched Rollouts from Expert Anchors Bridge SFT & RL for Reasoning	PAPER	-
2025-06	SRFT	SRFT: A Single-Stage Method with Supervised and Reinforcement Fine-Tuning for Reasoning	PAPER	-

A Survey of Reinforcement Learning for Large Reasoning Models

Published on Sep 11 · Submitted by Kaiyan Zhang on Sep 11 #1 Paper of the day

TsinghuaC3I

Upvoted 182 +174

Authors: Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, Fangfu Liu, Xiang Xu, Jiaze Ma +17 authors

https://github.com/TsinghuaC3I/Awesome-RL-for-LRMs

Our Works

- Reward Design: PRIME; TTRL; AttentionCompass
- Policy Optimization: Entropy; FlowRL; IFT; HPT
- Abilities Evaluation: Compositional Function
- Applications: SimpleVLA-RL; SSRL; MARTI; ReviewRL

成果整理 | RL for LRM综述与10篇本组相关工作

原创 C3I 课题组 TsinghuaC3I 2025年09月30日 20:53 安徽

清华大学协同交互智能研究中心 成果发布

1. **奖励机制**：提出密集奖励PRIME与自我奖励框架TTRL，构建多尺度RL优化目标；
2. **策略优化**：通过熵机制、FlowRL、IFT与HPT等方法，提升策略稳定性与多样性，并统一主流后训练方法；
3. **能力评估**：借助复合函数任务验证模型的复杂推理与泛化能力；
4. **应用验证**：SimpleVLA-RL，SSRL和MARTI分别在具身智能、搜索代理与多智能体系统中完成RL方法的集成验证。

https://mp.weixin.qq.com/s/Q0lvMRlw3MA9tSoDudro5g

Our Team

Kaiyan Zhang[1*†], Yuxin Zuo[1*†], Bingxiang He[1*], Youbang Sun[1*], Runze Liu[1*], Che Jiang[1*], Yuchen Fan[2,3*], Kai Tian[1*], Guoli Jia[1*], Pengfei Li[2,6*], Yu Fu[9*], Xingtai Lv[1*], Yuchen Zhang[2,4*], Sihang Zeng[7*], Shang Qu[1,2*], Haozhan Li[1*], Shijie Wang[2*], Yuru Wang[1*], Xinwei Long[1], Fangfu Liu[1], Xiang Xu[5], Jiaze Ma[1], Xuekai Zhu[3], Ermo Hua[1,2], Yihao Liu[1,2], Zonglin Li[2], Huayu Chen[1], Xiaoye Qu[2], Yafu Li[2], Weize Chen[1], Zhenzhao Yuan[1], Junqi Gao[6], Dong Li[6], Zhiyuan Ma[8], Ganqu Cui[2], Zhiyuan Liu[1], Biqing Qi[2‡], Ning Ding[1,2‡], Bowen Zhou[1,2‡]

[1] Tsinghua University [2] Shanghai AI Laboratory [3] Shanghai Jiao Tong University [4] Peking University
[5] University of Science and Technology of China [6] Harbin Institute of Technology [7] University of Washington
[8] Huazhong University of Science and Technology [9] University College London

† Project Lead. * Core Contributors. ‡ Corresponding Authors.

zhang-ky22@mails.tsinghua.edu.cn TsinghuaC3I/Awesome-RL-for-LRMs

教师

周伯文

讲席教授

丁宁

助理教授

孙友邦

助理研究员

学生

张开颜

博士生

龙鑫玮

博士生

华尔默

博士生

姜澈

博士生

朱学凯

博士生

吕兴泰

博士生

贾国力

博士生

张宇臣

博士生

曲上

博士生

李昊展

博士生

陈喆凯

博士生

高思研

博士生

田锴

硕士生

袁振钊

硕士生

https://c3i.ee.tsinghua.edu.cn/

Thanks & QA

GitHub
RL4LRM

Wechat
TsinghuaC3I

HuggingFace
RL4LRM

Website
TsinghuaC3I

zhang-ky22@mails.tsinghua.edu.cn

GitHub: https://iseesaw.github.io/
Twitter: https://x.com/Okhaylea