Towards Efficient Reinforcement Fine-Tuning

Zhi Wang (王志)

https://heyuanmingong.github.io

Email: zhiwang@nju.edu.cn

Nanjing University, China

2025-05-12

Contents

Reinforcement Learning

- Supervised Learning
  - (input, label)
- Unsupervised Learning
  - (input)
- **Reinforcement Learning**
  - sequential decision-making

- Computer Vision
  - Input: image pixels
- Natural Language Processing
  - Input: sentences
- **Reinforcement Learning**
  - Input: states

The Era of RL

- Video games: Human-level control through DRL, Nature 2015 (视频游戏)
- AlphaGo, Nature 2016;   AlphaGo Zero, Nature 2017 (围棋)
- AlphaStar in StarCraft II, Nature 2019 (星际争霸II)
- DRL for legged robots, Science Robotics 2019 (机器人学习)
- Superhuman AI for multiplayer poker, Science 2019 (德州扑克，多人非完全信息博弈)
- Discovering faster matrix multiplication algorithms, Nature 2022 (矩阵相乘算法发现，基础数学)
- Magnetic control of tokamak plasmas, Nature 2022 (可控核聚变控制)
- Outracing champion Gran Turismo drivers, Nature 2022 (赛车模拟控制)
- Safety validation of autonomous vehicles, Nature 2023 (无人驾驶安全验证)
- Faster sorting algorithms discovering, Nature 2023 (排序算法发现，基础信息科学)
- Champion-level drone racing, Nature 2023 (无人机竞速)
- Mastering diverse control tasks through world models, Nature 2025
- ……

The Dilemma of RL

RL = Artificial General Intelligence (AGI)?

Yet?

The Dilemma of RL

Transformers

Attention is **all you need**

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

… to attend to **all** positions in the decoder up to and including that position. **We need** to prevent … **We** implement this inside of scaled dot-product **attention** by masking out (setting to −∞) …

☆ Save 99 Cite Cited by 176805 Related articles All 73 versions

Vision Transformers

[PDF] An **image** is **worth** 16x16 words: Transformers for **image** recognition at scale

A Dosovitskiy, L Beyer, A Kolesnikov… - arXiv preprint arXiv …, 2020 - arxiv.org

… directly to **images**, with the fewest possible modifications. To do so, we split an **image** into patches … only to small-resolution **images**, while we handle medium-resolution **images** as well. …

☆ Save 99 Cite Cited by 60296 Related articles All 21 versions

Decision Transformers

Decision transformer: Reinforcement learning via sequence modeling

L Chen, K Lu, A Rajeswaran, K Lee… - Advances in neural …, 2021 - proceedings.neurips.cc

… of the **Transformer** architecture, and associated advances in language modeling such as GPT-x and BERT. In particular, we present **Decision Transformer**, … , **Decision Transformer** simply …

☆ Save 99 Cite Cited by 1942 Related articles All 13 versions

Artificial General Intelligence (AGI)

ChatGPT (Generative Pre-Training)

Next-token prediction

Transformer

Architecture

Enter text:

The dog eats the apples.

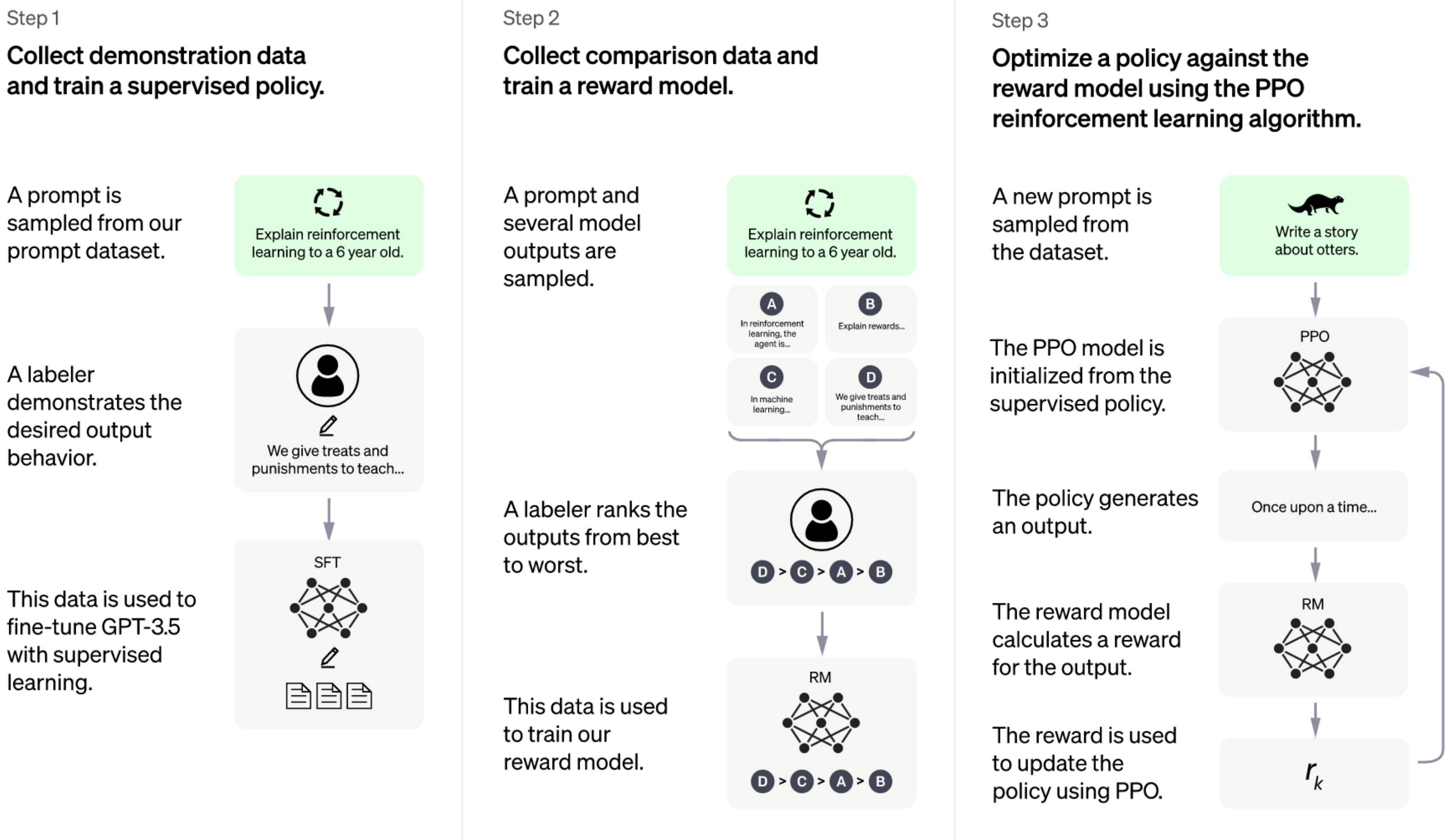
The	dog	eats	the	apples	.
464	3290	25365	262	22514	13

464 3290 25365 262 22514 13

Self-supervised learning

Algorithm

RL in ChatGPT



RL: Fine-tuning in Step 3, playing an auxiliary role

The Dilemma of RL

- Computer Vision
  - Input: **image pixels**
- Natural Language Processing
  - Input: **sentences**
- **Reinforcement Learning**
  - Input: **states, (states, actions)**

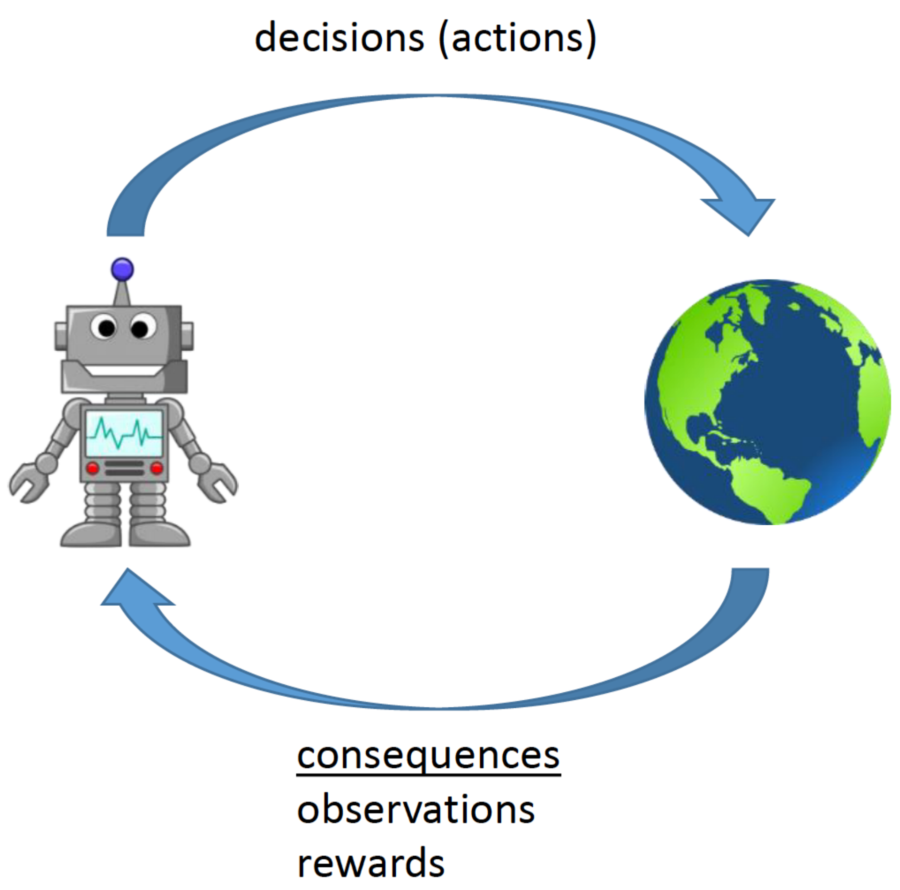
Semantics

not aligned

The Dilemma of RL

Data

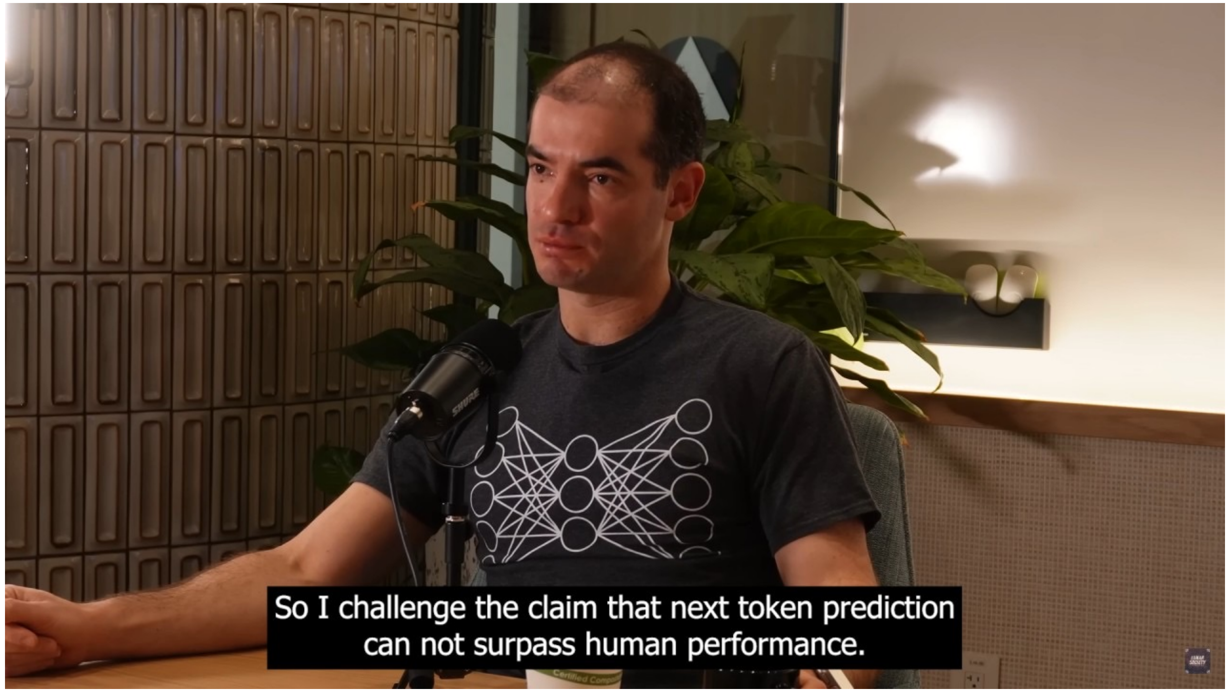
From online interactions



Confidence in RL

Supervised learning

Maybe imitating the intelligence within existing data?



Confidence in RL

Supervised learning

Maybe imitating the intelligence within existing data?

Reinforcement learning

Can surpass the intelligence within existing data definitely

Confidence in RL

LLM: From Pre-Training to Post-Training

Pre-training will end

-- by Ilya Sutskever

@NeurIPS 2025

> ### Pre-training as we know it will end
>
> Compute is growing:
>
> - Better hardware
> - Better algorithms
> - Larger clusters
>
> Data is not growing:
>
> - We have but one internet
> - **The fossil fuel of AI**

Confidence in RL

LLM: From Pre-Training to Post-Training

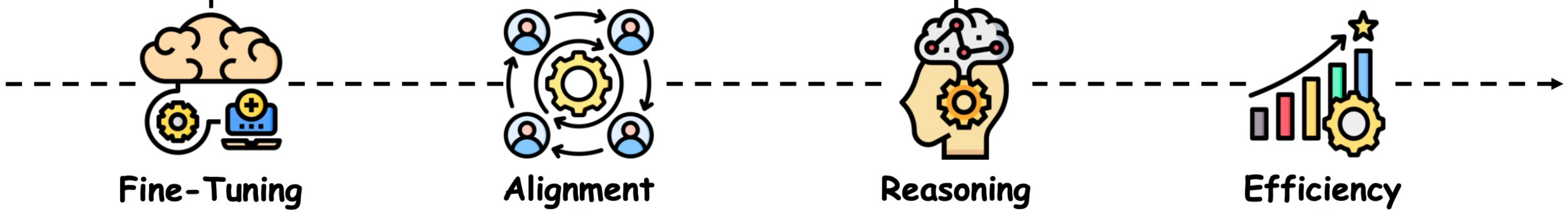
Reasoning, inference

What comes next?

- “Agents”??
- “Synthetic data”
- Inference time compute ~ O1

- Supervised Fine-Tuning
- Reinforcement Fine-Tuning

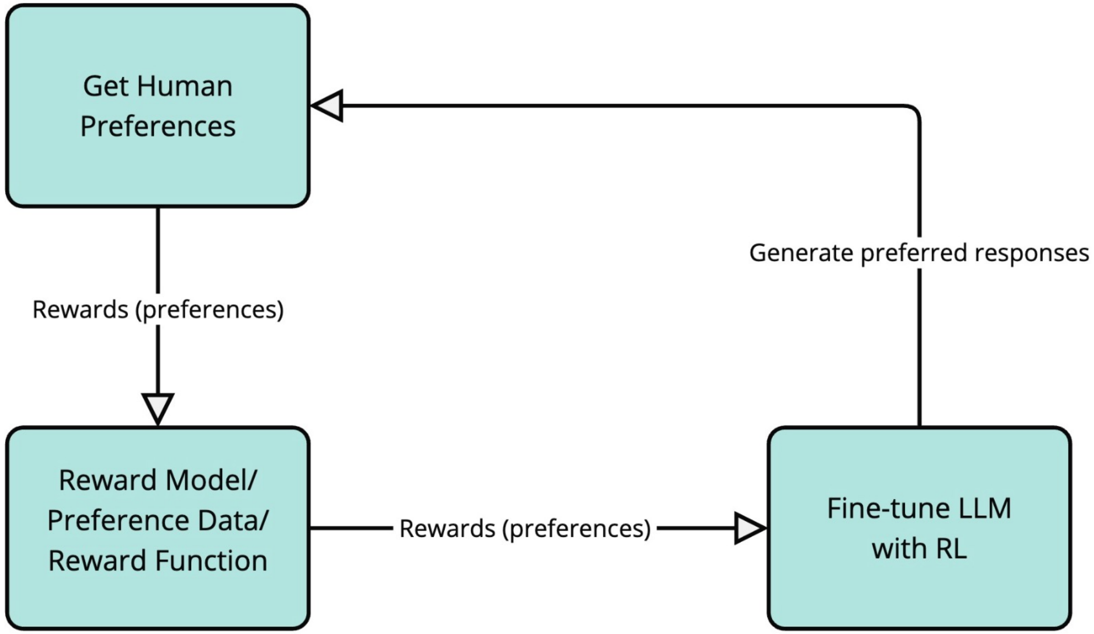
- Self-Refine for Reasoning
- Reinforcement Learning for Reasoning



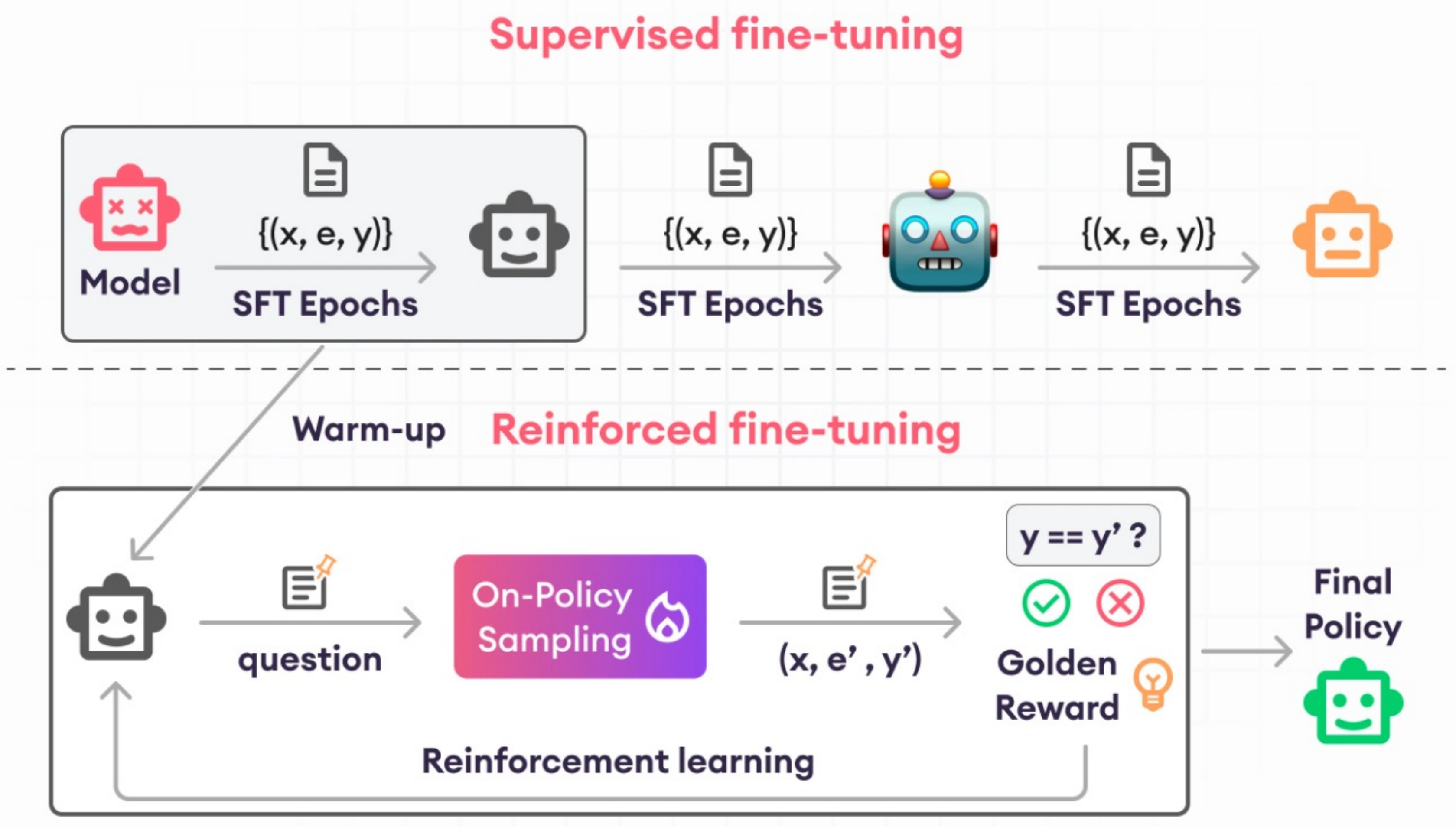
- Reinforcement Learning with Human Feedback
- Direct Preference Optimization
- Group Relative Policy Optimization

- Model Compression
- Parameter-Efficient Fine-Tuning
- Knowledge Distillation

Reinforcement Fine-Tuning



Reinforcement Fine-Tuning



Contents

LLM Agents

➢ AI Agents

- Capable of reasoning, decision-making, and communication

➢ LLM Agents

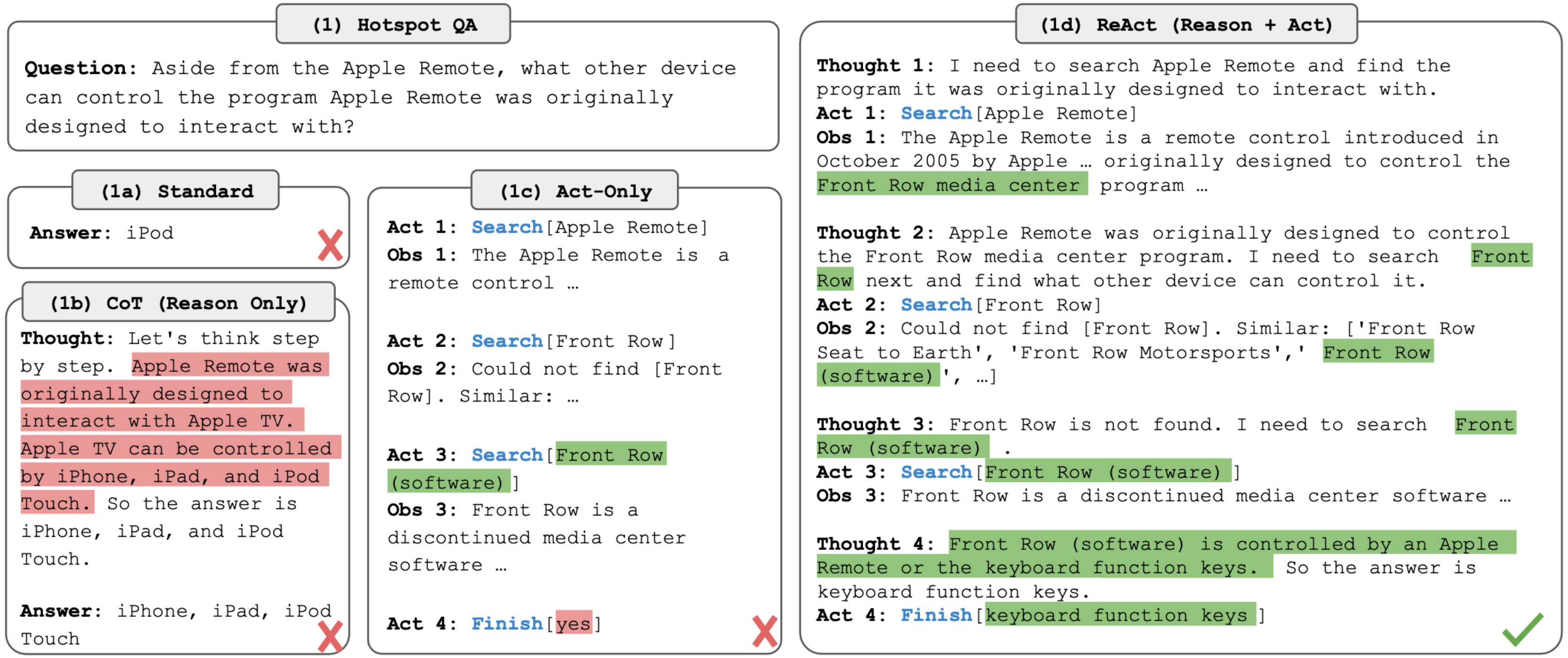
- Exploit LLMs as agents for tackling interactive decision-making tasks
- Prompt-based methods
- Supervised fine-tuning methods
- Reinforcement fine-tuning methods

- Wooldridge and Jennings, Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 1995.

Prompt-based Methods

➢ ReAct, Reflexion

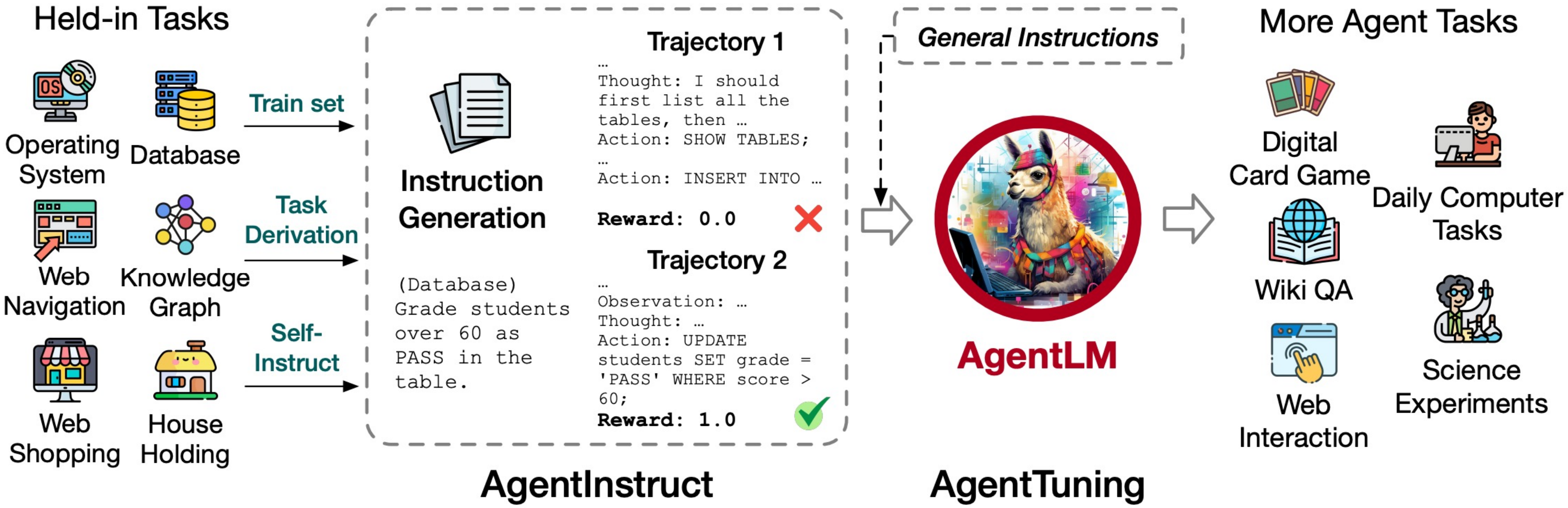
- recursively augment the prompt to a frozen LLM with verbal feedback
- prone to exceed the input length limit of in-context learning, especially for long-horizon tasks



- Yao et al., ReAct: Synergizing reasoning and acting in language models, ICLR 2023.
- Shinn et al., Reflexion: language agents with verbal reinforcement learning, NeurIPS 2023.

Supervised Fine-tuning

- AgentTuning, SwiftSage
  - unlock the potential of LLMs for downstream applications
  - performance is highly dependent on expensive expert demonstrations
  - can be limited due to deficient exploration of target environments

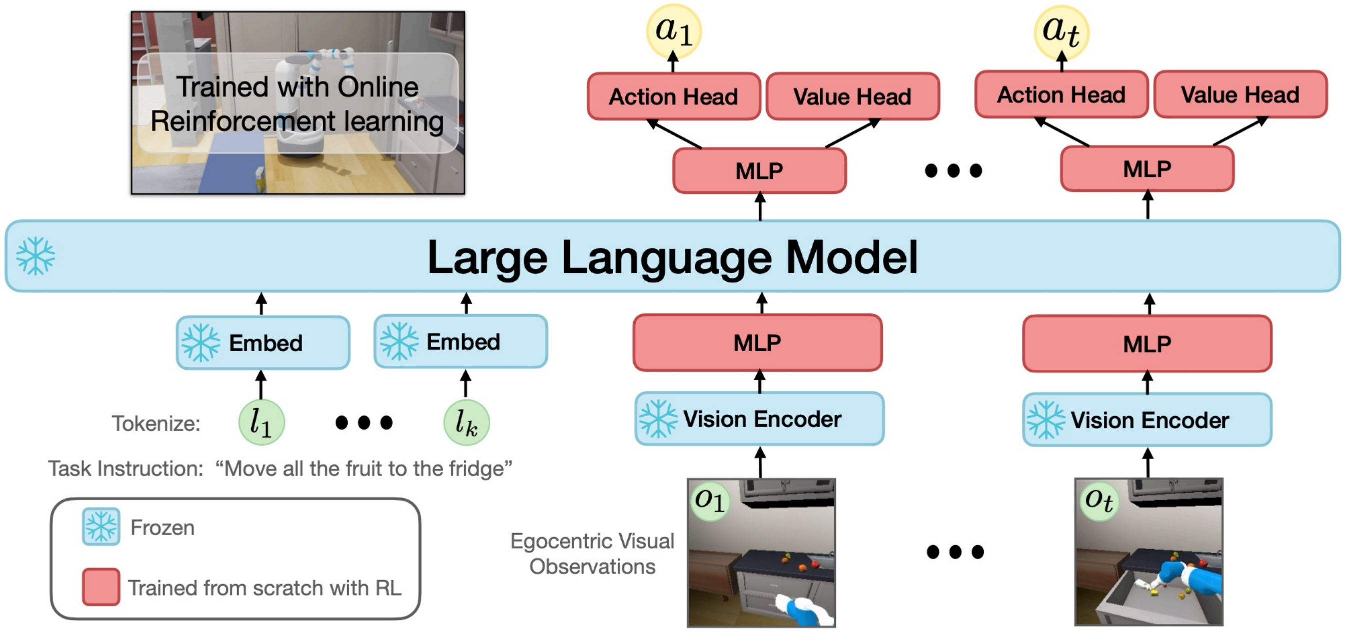


- Zeng et al., AgentTuning: Enabling Generalized Agent Abilities for LLMs, Findings of ACL 2024.
- Lin et al., SwiftSage: a generative agent with fast and slow thinking for complex interactive tasks, NeurIPS 2023.

Reinforcement Fine-tuning

➢ Intelligent agents must excel at imitating demonstrations and adapting behaviors through trial-and-error

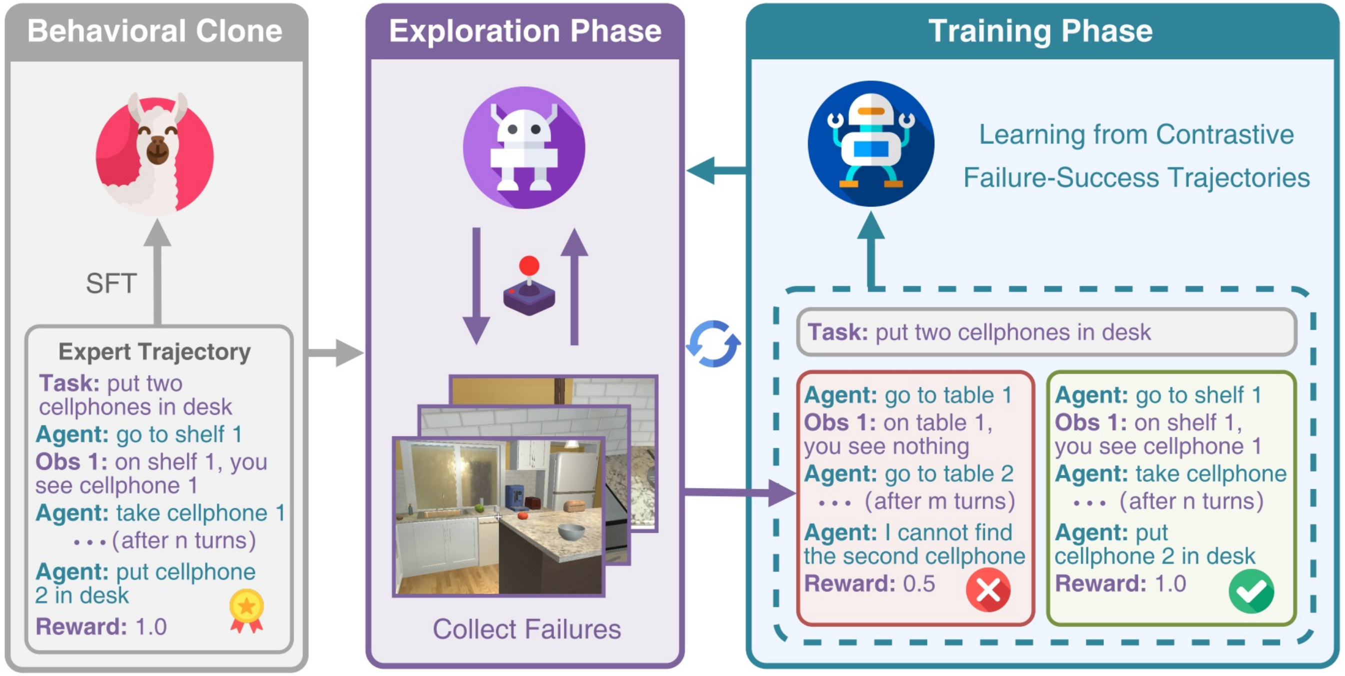
- Steer LLMs toward user-specified tasks, using offline Q-learning, PPO, DPO, etc.
- OpenAI o3, DeepSeek-R1



- Szot et al., Large language models as generalizable policies for embodied tasks, ICLR 2024.

Reinforcement Fine-tuning

- Intelligent agents must excel at imitating demonstrations and adapting behaviors through trial-and-error
  - Steer LLMs toward user-specified tasks, using offline Q-learning, PPO, DPO, etc.
  - OpenAI o3, DeepSeek-R1



- Song et al., Trial and error: Exploration-based trajectory optimization for LLM agents, ACL 2024.

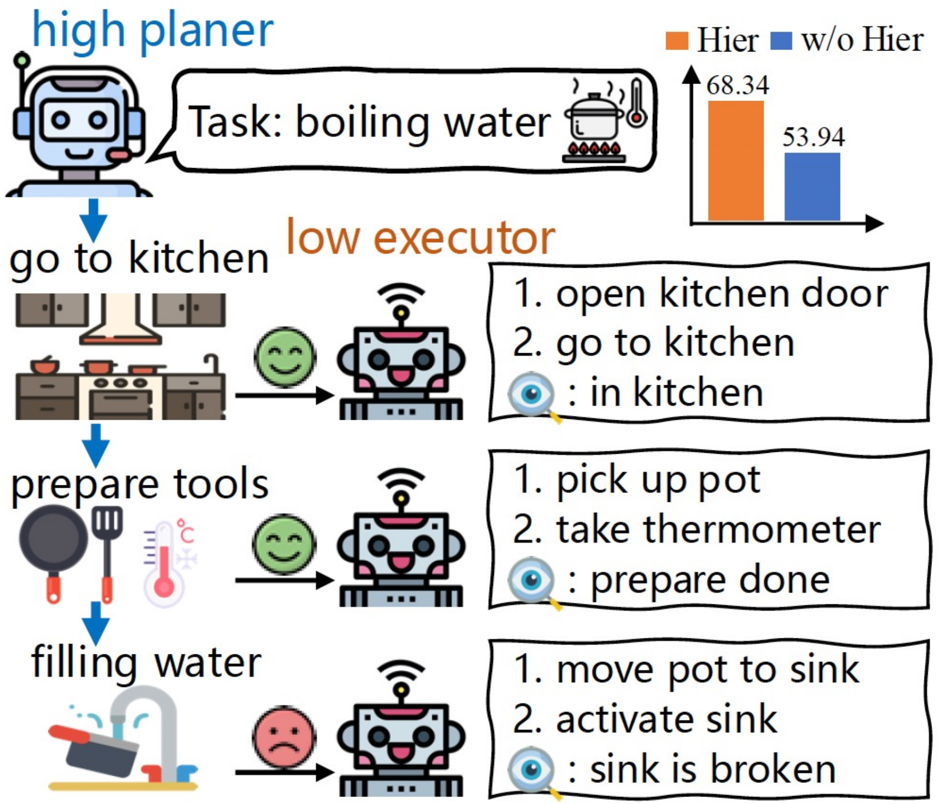
Reinforcement Fine-tuning

➢ Challenges

- RL intrinsically requires tedious and vast environment interactions, leading to brittle performance and poor sample efficiency
- Build LLM agents with open-ended textual commands: tackle huge action spaces, execute long-horizon planning, and learn from sparse-reward feedback
- Demand a broad spectrum of vital capabilities: long-term credit assignment, understanding the real physical world, and sophisticated exploration with structured reasoning

Our Solution: Hierarchical RL

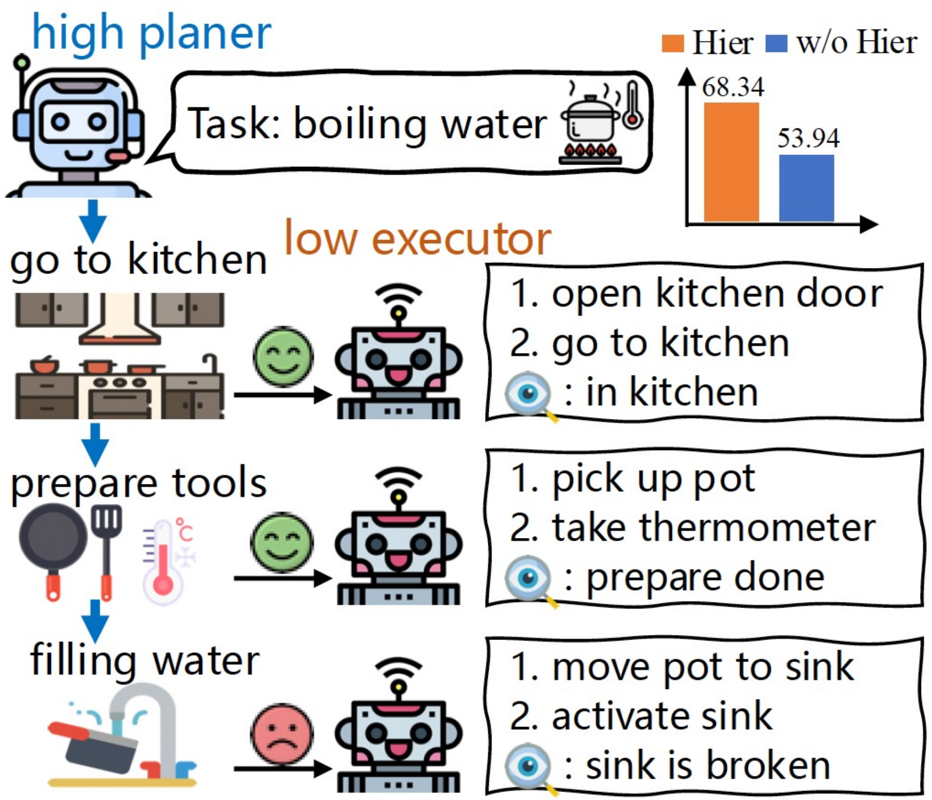
- The divide-and-conquer principle
  - how corporations divide into specialized departments
  - how biological systems organize cells to form tissues and organs
  - showcase remarkable efficiency for solving intricate tasks in a more human-like manner



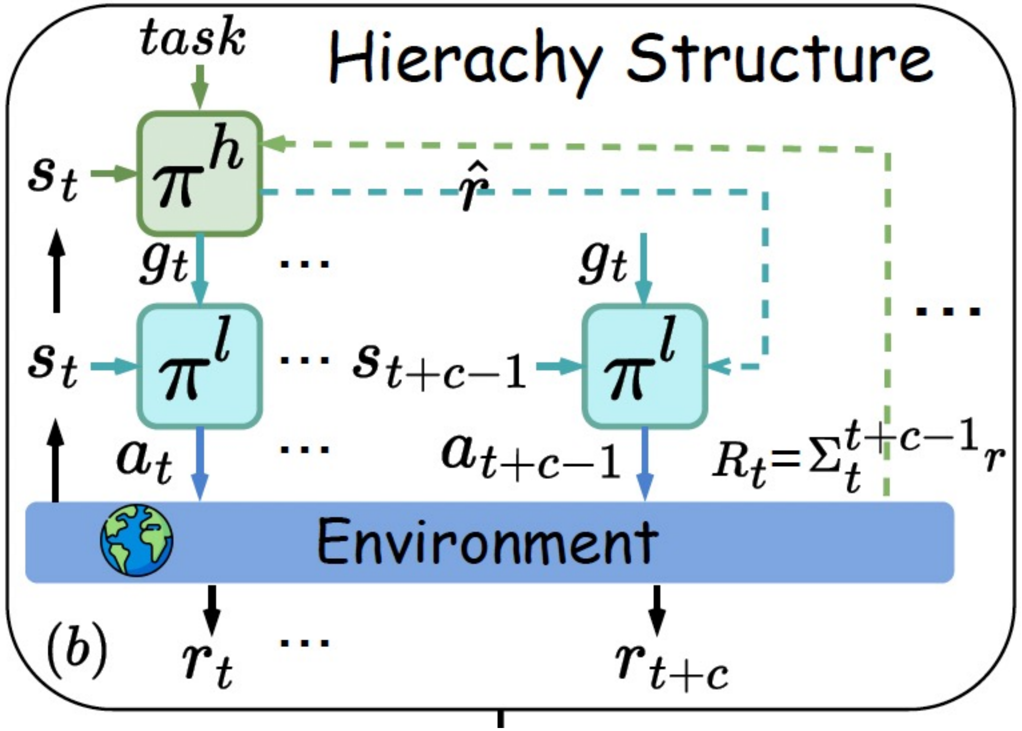
Our method: GLIDER

Grounding Language Models as EffIcient Decision-Making Agents via Offline HiErarchical RL

- Decomposes complicated problems into a series of coherent chain-of-thought reasoning sub-tasks
- Flexible temporal abstraction, enhance exploration
- Divide and conquer, a human-like manner



Hierarchical Structure



- High-level dataset

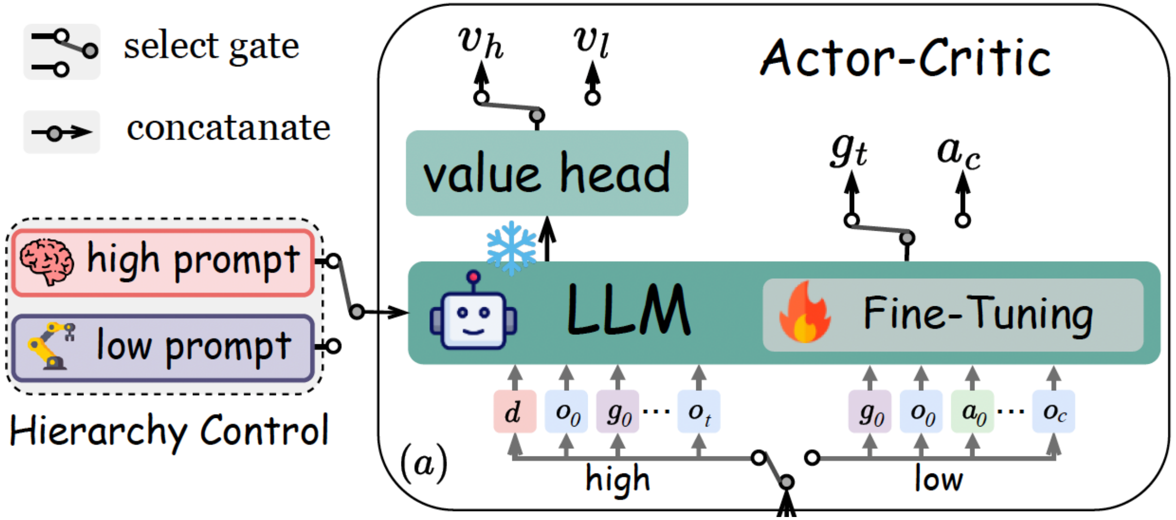
$$\mathcal{D}^h = \Sigma_N \left[\, d;\ (o_0, g_0, \Sigma r_{0:c-1}, o_c)\,, \ldots, \right.$$
$$\left. (o_t, g_t, \Sigma r_{t:t+c-1}, o_{t+c})\,, \ldots \,\right]$$

- Low-level dataset

$$\mathcal{D}^l = \Sigma_N \Sigma_t \left[\, g_t;\ (o_t, a_t, \hat{r}_t, o_{t+1})\,, \ldots, \right.$$
$$\left. (o_{t+c-1}, a_{t+c-1}, \hat{r}_{t+c-1}, o_{t+c})\,\right]$$

Parameter-efficient hierarchical model

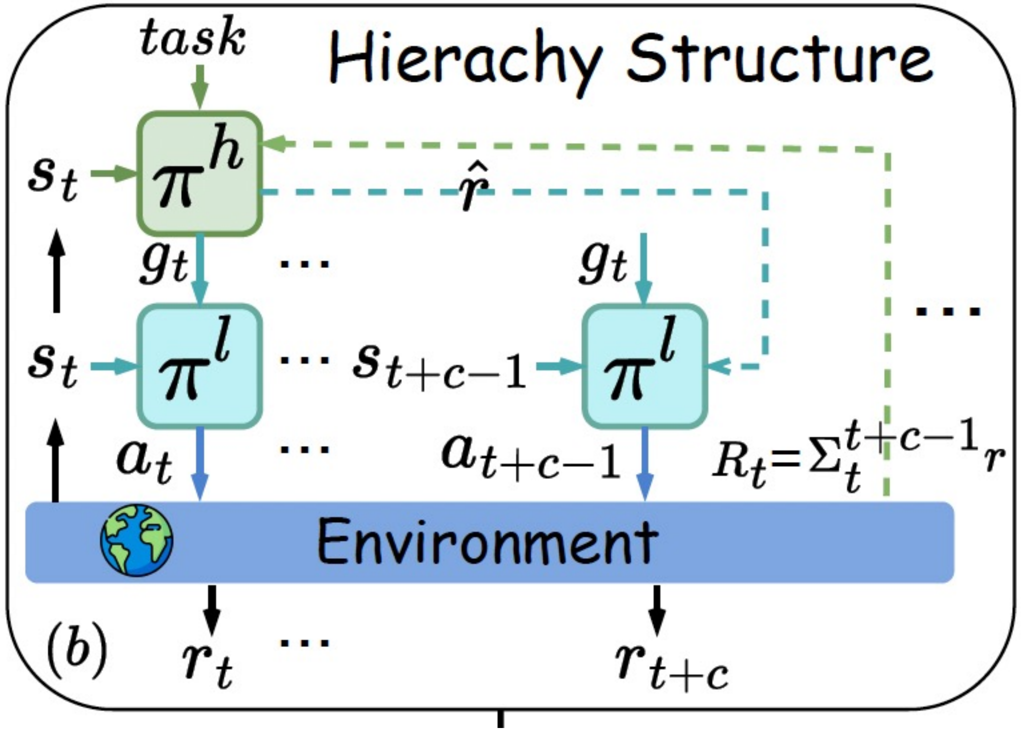
- Actor-critic: share the same frozen LLM backbone
  - Actor – add LoRA layers
  - Critic - add MLP layers



- High- and low-level policies share the same actor-critic models
  - differ in a hierarchy prompt that specifies the level of current inputs
  - harness the powerful capability of LLMs to perform in-context learning

Generally Applicable Hierarchy

- ➢ High-level planner → sub-task goals → low level policy
  - High-level planner is guided by environment-provided rewards
  - Low-level policy is instructed by the sub-task completion signal
- ➢ Completion derived from environment observations
  - Eliminate the necessity for any manual or task-specific design
  - Make it broadly applicable



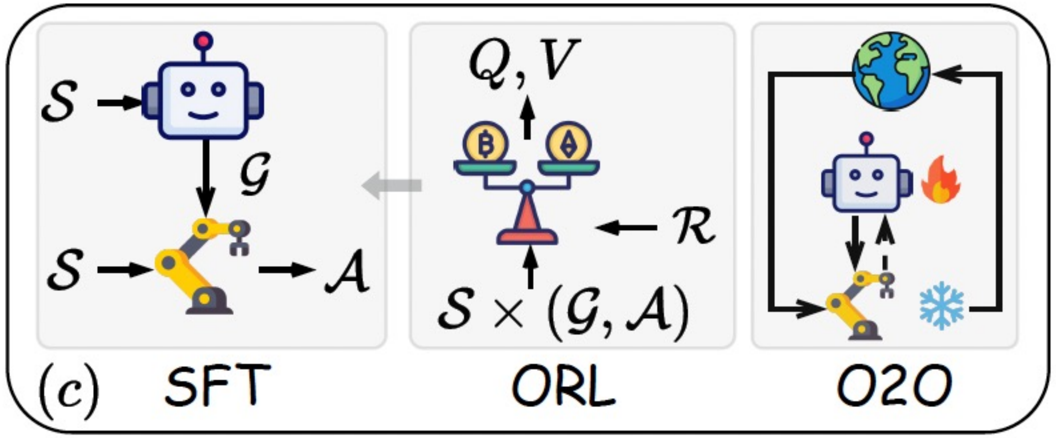
Training Pipeline

➢ Base agent construction using SFT

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(d,o;g)\sim\mathcal{D}^h}\left[\log \pi_\theta^h(g|d,o)\right] + \lambda \cdot n_h - \mathbb{E}_{(g,o;a)\sim\mathcal{D}^l}\left[\log \pi_\theta^l(a|g,o)\right] + \lambda \cdot n_l,$$

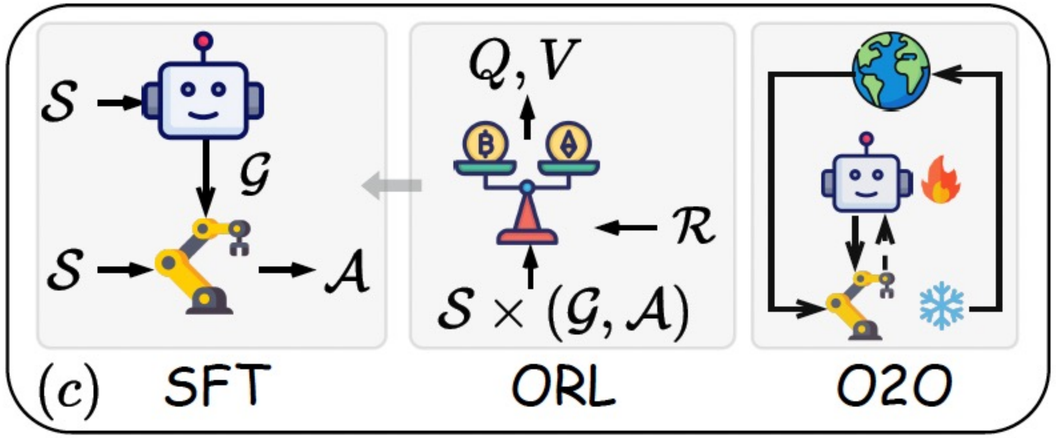
✓ a length regularization term

- encourage the LLM policy to generate concise task plans and atomic actions
- for effective interaction with the environment



Training Pipeline

➢ Offline Hierarchical RL



✓ Token-level actor

$$\begin{aligned}\mathcal{L}_\pi(\theta) &= -\mathbb{E}_{(s,u)\sim D_r}\left[\exp\left(\frac{1}{\lambda}A(s,u)\right)\cdot\log\pi_\theta(u\mid s)\right]\\ &= -\mathbb{E}_{(s,u)\sim D_r}\left[\exp\left(\frac{1}{\lambda}\left(Q_\phi(s,u)-V_\psi(s)\right)\right)\right.\\ &\qquad\left.\cdot\sum_{i=1}^{n}\log\pi_\theta(w_i\mid s,w_{1:i-1})\right].\end{aligned} \tag{7}$$

✓ Sentence-level critic

$$\mathcal{L}_Q(\phi)=\mathbb{E}_{(s,u,r,s')\sim D_r}\left[\left(Q_\phi(s,u)-r-\gamma V_{\bar\psi}(s')\right)^2\right]$$

$$\mathcal{L}_V(\psi)=\mathbb{E}_{s\sim D_r}\left[\mathbb{E}_{u\sim\pi_\theta(\cdot|s)}\left[L_2^\tau\left(V_\psi(s)-Q_{\bar\phi}(s,u)\right)\right]\right]$$

Training Pipeline

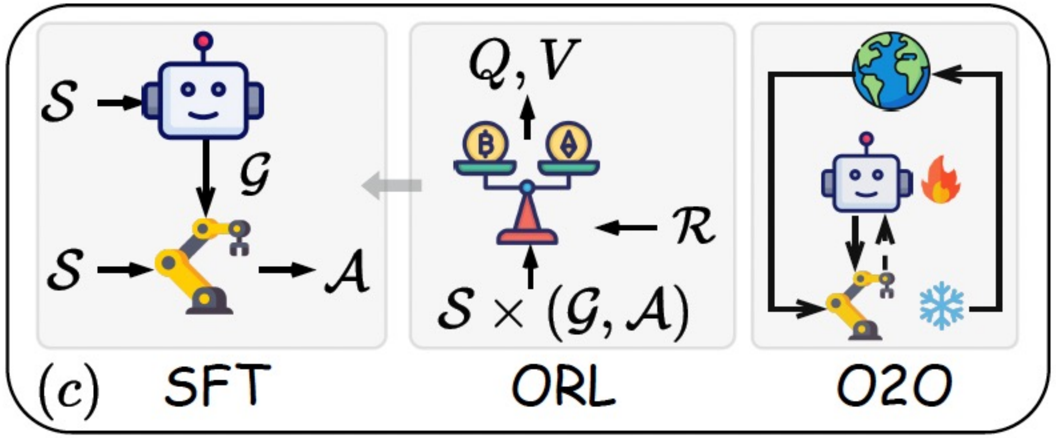
➢ Offline-to-Online Adaptation

✓ Fix low-level skills

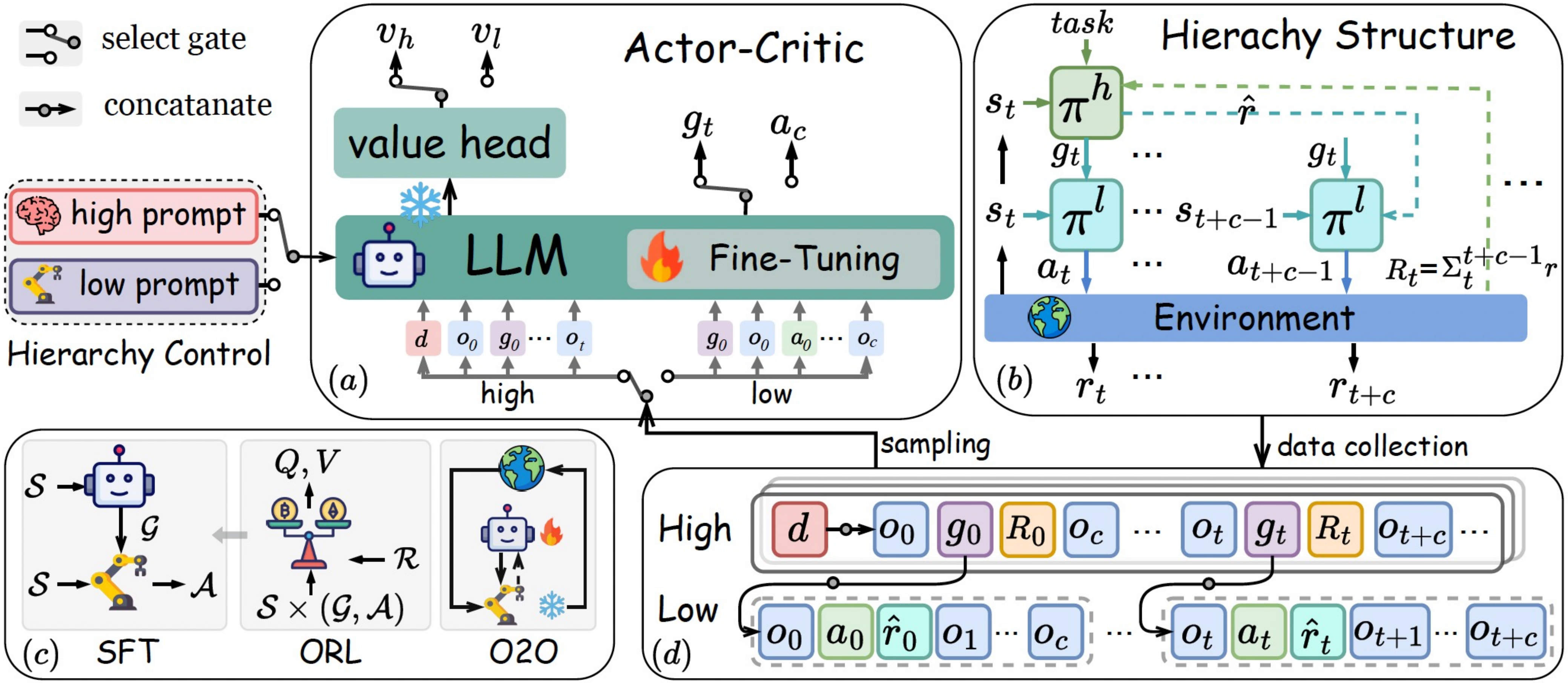
- pre-trained using intrinsic reward functions, not task-specific ones
- high generalization capacity across tasks
- good robustness to distribution shift

✓ Finetune high-level policy

- quickly adapt to new tasks with improved exploration efficiency



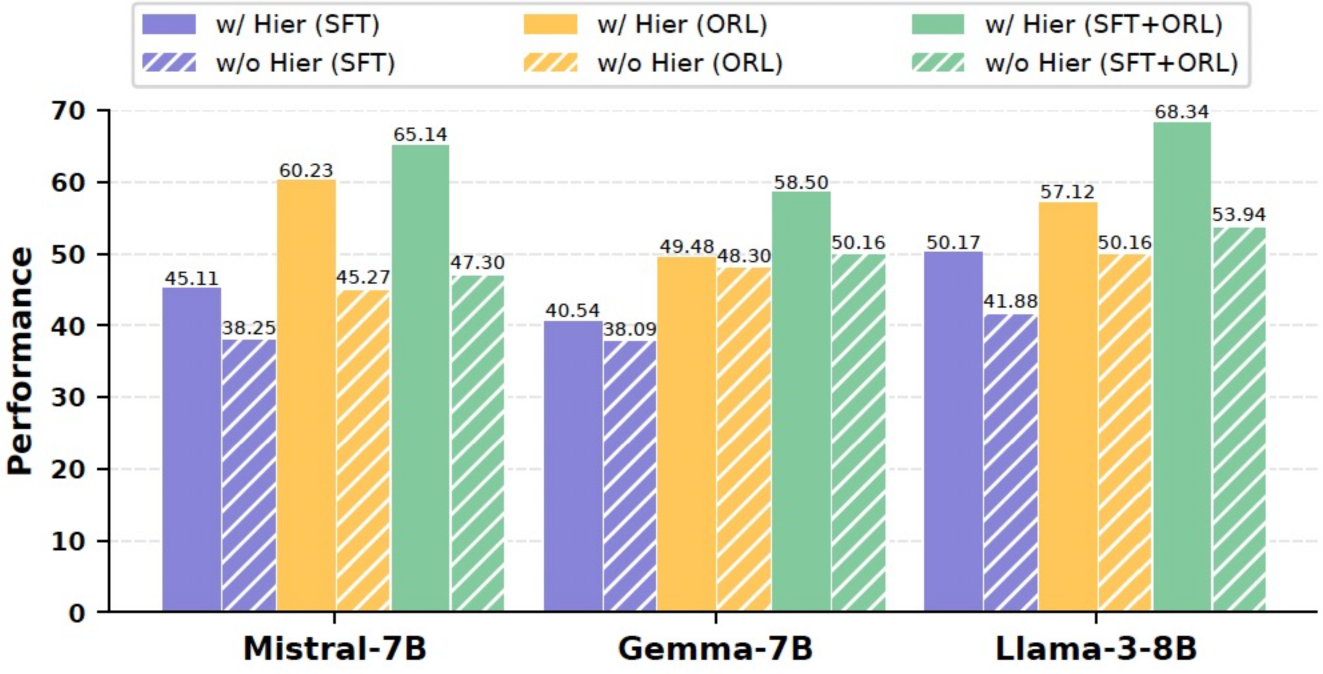
GLIDER: Overall Architecture



Primary Results

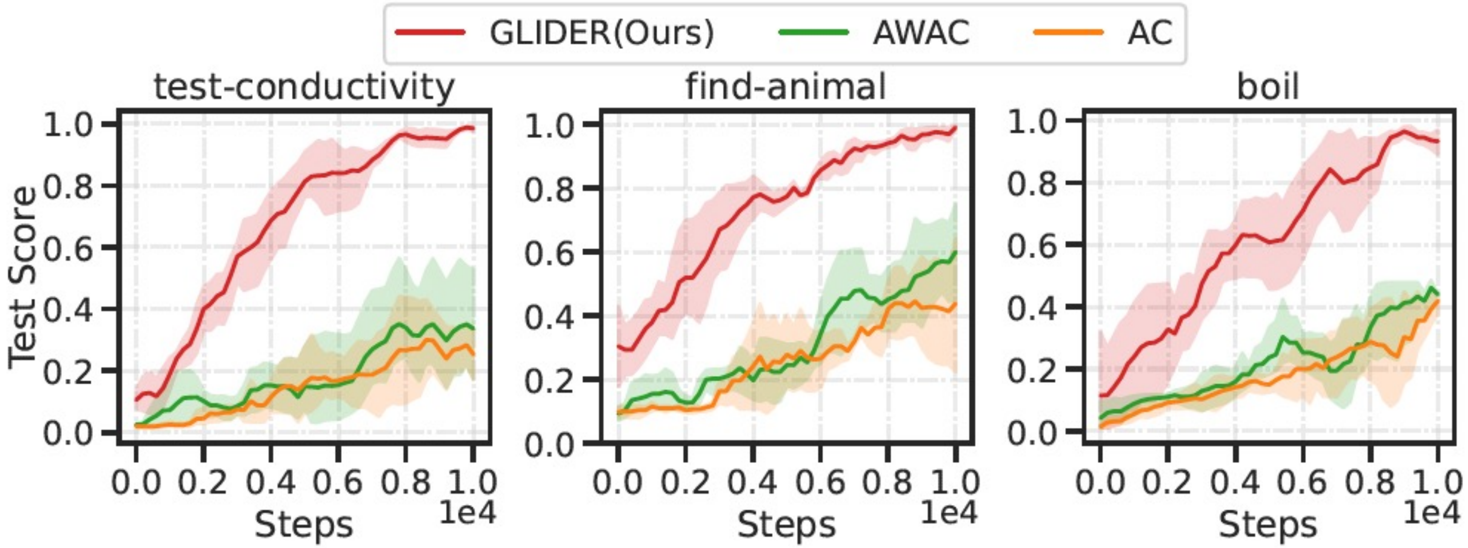
Backbone	Method	ScienceWorld		AlfWorld	
		Seen	Unseen	Seen	Unseen
Mistral-7B	ReAct	20.72	17.65	7.86	5.22
	Reflexion	21.07	18.11	11.56	6.00
	SwitchSage	48.40	45.25	30.29	26.52
	NAT	57.12	50.79	64.43	68.96
	ETO	58.17	51.85	66.84	71.43
	GLIDER	**67.31** (↑ 15.71%)	**65.14** (↑ 25.63%)	**70.02** (↑ 4.76%)	**74.83** (↑ 4.76%)
Gemma-7B	ReAct	3.58	3.51	6.43	2.24
	Reflexion	4.94	3.93	7.14	2.99
	SwitchSage	33.43	30.90	8.23	5.72
	NAT	47.63	44.98	67.86	65.88
	ETO	50.44	47.84	66.43	68.66
	GLIDER	**63.67** (↑ 26.23%)	**58.50** (↑ 22.28%)	**72.12** (↑ 6.28%)	**70.88** (↑ 3.23%)
Llama-3-8B	ReAct	24.76	22.66	2.86	3.73
	Reflexion	27.23	25.41	4.29	4.48
	SwitchSage	42.22	40.58	20.39	10.78
	NAT	55.24	48.76	60.71	59.70
	ETO	57.90	52.33	64.29	64.18
	GLIDER	**77.43** (↑ 33.73%)	**68.34** (↑ 30.59%)	**71.56** (↑ 11.31%)	**75.38** (↑ 17.45%)

Ablations



- ✓ the hierarchical structure plays a crucial part in all training stages
- ✓ training offline RL agents from scratch performs better than training SFT agents

Offline-to-Online Adaptation



- ✓ a higher initial test score, superior zero-shot generalization capacity
- ✓ faster adaptation, better final performance

In Summary

✓ An innovative hierarchical model architecture

- superior parameter efficiency and broad applicability
- efficiently grounding LLM agents to tackle complex, long-horizon tasks

✓ Future directions

- extend beyond strict agent tasks： many LLM tasks can also be reformulated as the sequential decision-making paradigm through process reward model (PRM)
- Extend to broader domains, e.g., mathematical reasoning, code generation

Contents

Large Reasoning Models

✓ OpenAI-o1, DeepSeek-R1, Kimi-1.5

- Extensive CoT responses, sophisticated behaviors (self-reflection, self-correction)
- Through RL with purely rule-based rewards

✓ Zero-RL

- Reinforcement fine-tuning to the base model, without SFT
- Elicit reasoning potentials using models' own rollouts

Zero-RL

✓ On-policy

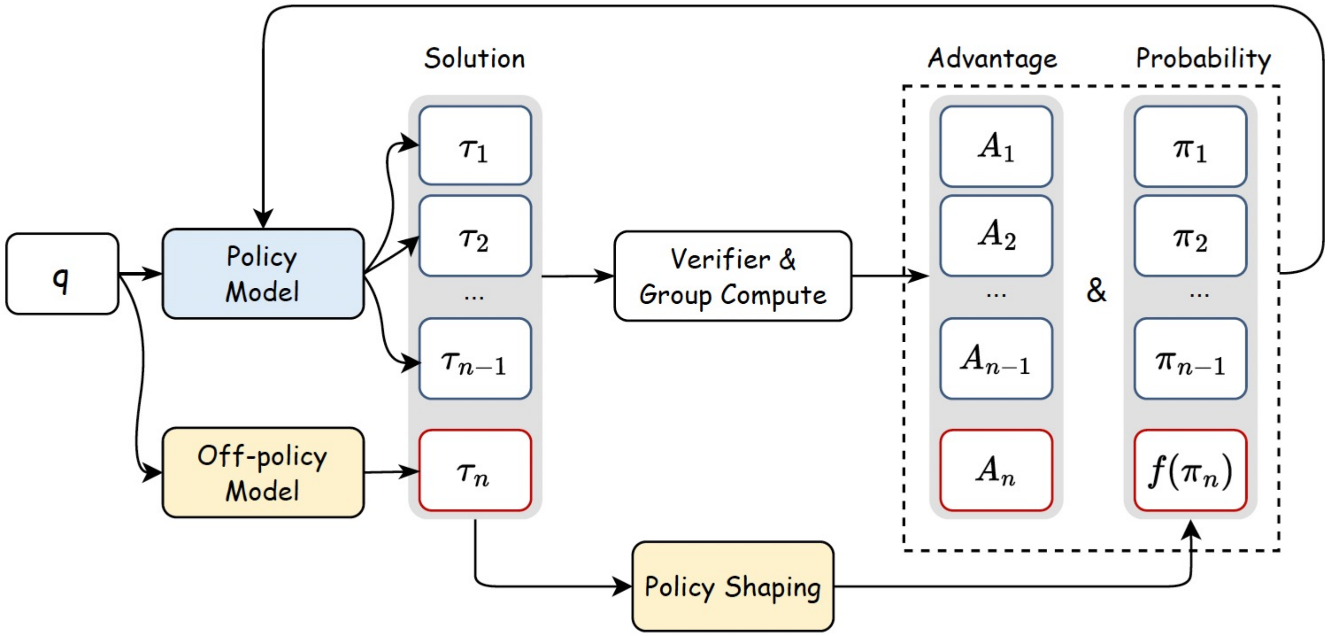
- A huge language space, hard exploration
- amplifie existing behaviors rather than introducing genuinely novel cognitive capacities

How can we empower LLMs to acquire reasoning behaviors surpassing their initial cognitive boundaries

Off-policy guidance

✓ On-policy rollouts + off-policy knowledge

- vs. pure imitation: the generalization limits, which locks models into superficial and rigid reasoning models that impede further learning



Our Method: LUFFY

Learning to Reason Under OFF-policY Guidance

- balance imitation and exploration by combining off-policy demonstrations with on-policy rollouts

$$\mathcal{J}_{\text{Mixed}}(\theta) = \frac{1}{Z}(\underbrace{\sum_{j=1}^{N_{\text{off}}}\sum_{t=1}^{|\tau_j|}\min[\hat{r}_{j,t}(\theta,\phi)\hat{A}_j, \text{clip}(\hat{r}_{j,t}(\theta,\phi); 1-\epsilon, 1+\epsilon)\hat{A}_j]}_{\text{off-policy objective}}$$

$$+ \underbrace{\sum_{i=1}^{N_{\text{on}}}\sum_{t=1}^{|\tau_i|}\min[r_{i,t}(\theta)\hat{A}_i, \text{clip}(r_{i,t}(\theta); 1-\epsilon, 1+\epsilon)\hat{A}_i]}_{\text{on-policy objective}}),$$

$$\text{where } \hat{r}_{j,t}(\theta,\phi) = \frac{\pi_\theta(\tau_{j,t}|q, \tau_{j,<t})}{\pi_\phi(\tau_{j,t}|q, \tau_{j,<t})} \text{ and } r_{i,t}(\theta) = \frac{\pi_\theta(\tau_{i,t}|q, \tau_{i,<t})}{\pi_{\theta_{\text{old}}}(\tau_{i,t}|q, \tau_{i,<t})}.$$

Mixed Policy GRPO

- Project Page: https://github.com/ElliottYan/LUFFY

Our Method: LUFFY

✓ Mixed Policy GRPO

- Importance sampling
- Convergence rate $O(1/\sqrt{K})$

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{\tau_j \sim \pi_\theta(\tau)} \left[\nabla_\theta \log \pi_\theta(\tau_j) \hat{A}_j \right]$$
$$= \mathbb{E}_{\tau_j \sim \pi_\phi(\tau)} \left[\frac{\pi_\theta(\tau_j)}{\pi_\phi(\tau_j)} \nabla_\theta \log \pi_\theta(\tau_j) \hat{A}_j \right].$$

Theorem 1. *Suppose the objective function of the policy gradient algorithm $J \in \mathcal{J}_n$, where $\mathcal{J}_n$ is the class of finite-sum Lipschitz smooth functions, has σ-bounded gradients, and the importance weight $w = \pi_{\boldsymbol{\theta}}/\pi_{\boldsymbol{\phi}}$ is clipped to be bounded by $[\underline{w}, \overline{w}]$. Let $\alpha_k = \alpha = c/\sqrt{K}$ where $c = \sqrt{\frac{2(J(\boldsymbol{\theta}^*)-J(\boldsymbol{\theta}^0))}{L\sigma^2 \underline{w}\overline{w}}}$, and $\boldsymbol{\theta}^*$ is an optimal solution. Then, the iterates of our algorithm in Eq. (3) satisfy:*

$$\min_{0 \le k \le K-1} \mathbb{E}[||\nabla J(\boldsymbol{\theta}^k)||^2] \le \sqrt{\frac{2(J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}^0))L\overline{w}}{K\underline{w}}}\sigma.$$

- Project Page: https://github.com/ElliottYan/LUFFY

Our Method: LUFFY

✓ **Policy shaping via regularized importance sampling**

- re-weights the gradient of off-policy distributions
- enhance learning from low-probability tokens

$$\begin{aligned}\nabla_\theta \mathcal{J}_{\text{SHAPING-OFF}}(\theta) &= \mathbb{E}_{\tau\sim\pi_\phi}\left[\nabla_\theta f(\frac{\pi_\theta}{\pi_\phi})\cdot \hat{A}_j\right] \\ &= \mathbb{E}_{\tau\sim\pi_\phi}\left[f'(\frac{\pi_\theta}{\pi_\phi})\frac{1}{\pi_\phi}\nabla_\theta \pi_\theta \cdot \hat{A}_j\right] \\ &= \mathbb{E}_{\tau\sim\pi_\phi}\left[f'(\pi_\theta)\underbrace{\frac{\pi_\theta}{\pi_\phi}\nabla_\theta \log \pi_\theta \cdot \hat{A}_j}_{\text{importance sampling}}.\right]\end{aligned}$$

$$f(x) = \frac{x}{x+\lambda}$$

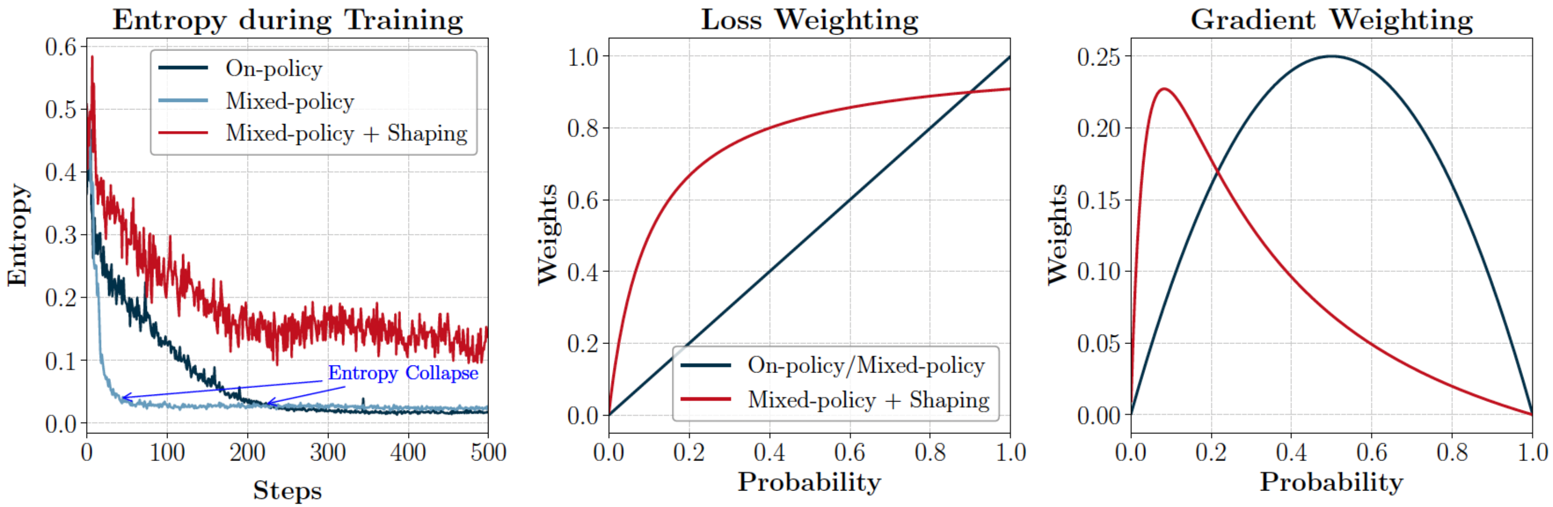
$$\Rightarrow \mathbb{E}_{\tau\sim\pi_\phi}\left[\frac{\gamma}{(\pi_\theta+\gamma)^2}\pi_\theta\left(1-\pi_\theta\right)\cdot \hat{A}_j\right]$$

- Project Page: https://github.com/ElliottYan/LUFFY

Our Method: LUFFY

✓ Policy shaping via regularized importance sampling

- re-weights the gradient of off-policy distributions
- enhance learning from low-probability tokens



- Project Page: https://github.com/ElliottYan/LUFFY

LUFFY: Primary Results

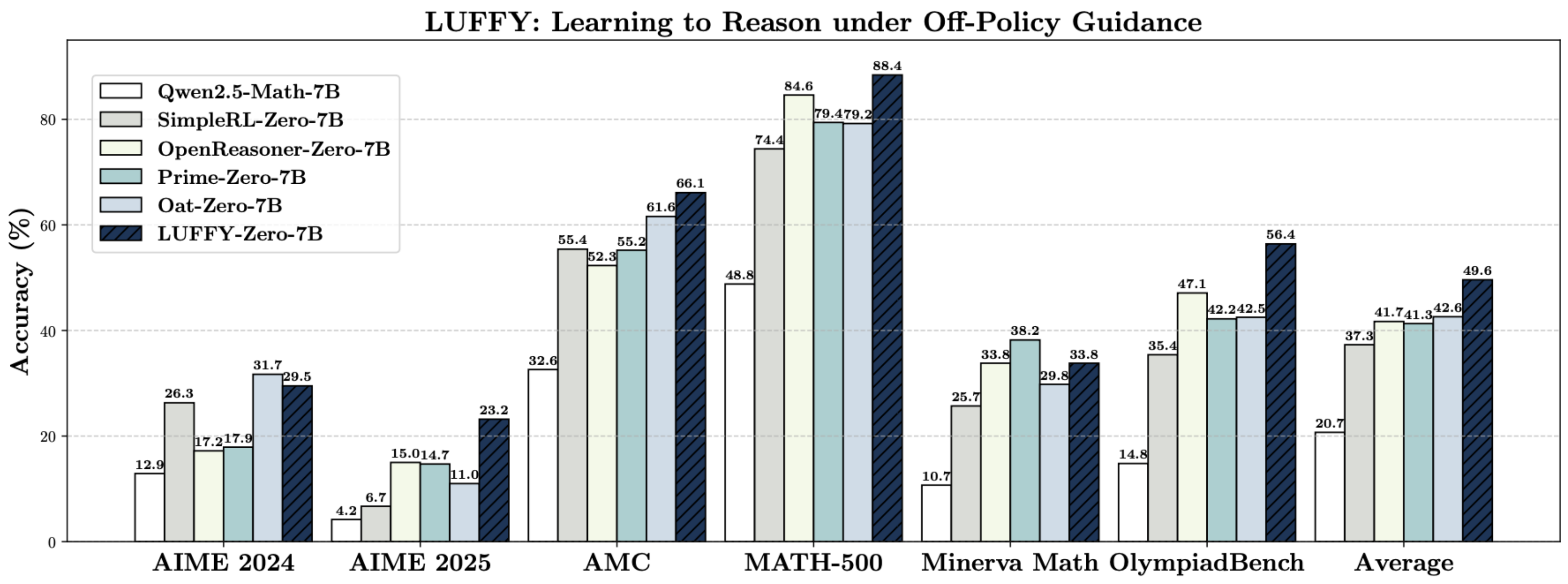


Figure 2: Overall performance across six competition-level benchmarks (AIME 2024, AIME 2025, AMC, MATH-500, Minerva Math, and OlympiadBench). **LUFFY** achieves an average score of **49.6**, delivering a substantial performance gain of over **+7.0** points on average compared to existing zero reinforcement learning methods.

- Project Page: https://github.com/ElliottYan/LUFFY

Contents

Conclusions

- Reinforcement Learning
  - Bottlenecks: semantics of input space, online interactions
  - Reinforcement fine-tuning, post-training
- Efficient LLM agents via Offline Hierarchical RL
  - Divide-and-conquer, a human-like manner
  - Parameter-efficient and generally applicable hierarchy, offline-to-online adaptation
- Learning to reason under off-policy guidance
  - On-policy rollouts + off-policy guidance, a pure RL framework

Thank You.

Zhi Wang (王志)

https://heyuanmingong.github.io

Email: zhiwang@nju.edu.cn

Nanjing University, China

2025-05-12