Agentic RL 如何让语言模型成为自主智能体

Guibin Zhang @ NUS SoC

2025.9.18

0.1 The Concept of "Agentic RL"

- Agentic RL的提法对传统RL社区或许听起来有一些奇怪：
  - RL本来就是for Agent的
  - 或许RL for Agentic LLM，RL for LLM Agents会更合适
- 不过，在当前的语境下，Agentic RL似乎往往被普遍认知为"帮助LLM更具备Agent特质的RL算法"

Agentic Reinforcement Learning (Agentic RL) refers to a paradigm in which LLMs, rather than being treated as *static conditional generators* optimized for single-turn output alignment or benchmark performance, are conceptualized as *learnable policies* embedded within sequential decision-making loops, where RL endows them with autonomous agentic capabilities, such as planning, reasoning, tool use, memory maintenance, and self-reflection, enabling the emergence of long-horizon cognitive and interactive behaviors in *partially observable, dynamic environments*.

0.2 Scope of Our Survey

Main Focus:

✓ RL如何在动态环境中增强基于LLM的Agent（或具有Agent特质的LLM）的能力

Out-of-Scope（尽管偶尔会提到）：

✗ 用于人类价值对齐的强化学习（例如，用于拒绝有害查询的RLHF）；
✗ 非基于LLM的传统RL算法（如传统多智能体强化学习MARL）；
✗ 用于提升LLM在静态基准测试（比如GSM8K，MATH-500）中表现的RL算法。

Primary focus:

✔ how RL empowers LLM-based agents (or, LLMs with agentic characteristics) in *dynamic environments*

Out of scope (though occasionally mentioned):

✗ RL for human value alignment (*e.g.*, RL for harmful query refusal);
✗ traditional RL algorithms that are not LLM-based (*e.g.*, MARL [28]);
✗ RL for boosting pure LLM performance on static benchmarks.

0.3 Content of Our Survey

- From LLM RL to Agentic RL: formulation
- RL for Different Agentic Capability
- RL for Different Vertical Domains
- Environments and Frameworks
- Challenges and Future Directions

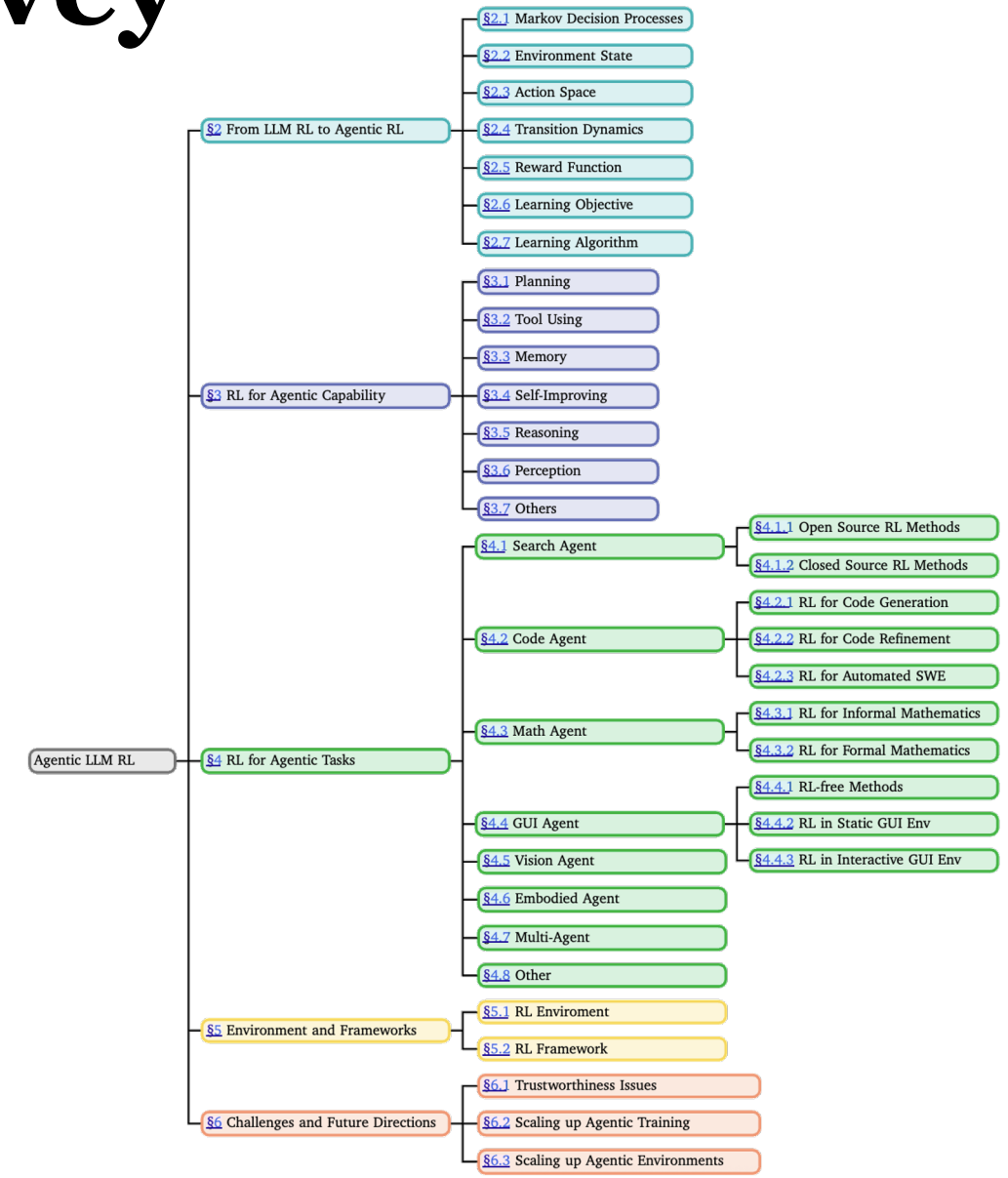


Figure 1: The primary organizational structure of the survey.

1.0 From LLM RL to Agentic RL

- Reward Design
- Transition Dynamic
- Action Space
- Objective
- RL Algorithms
- Environments

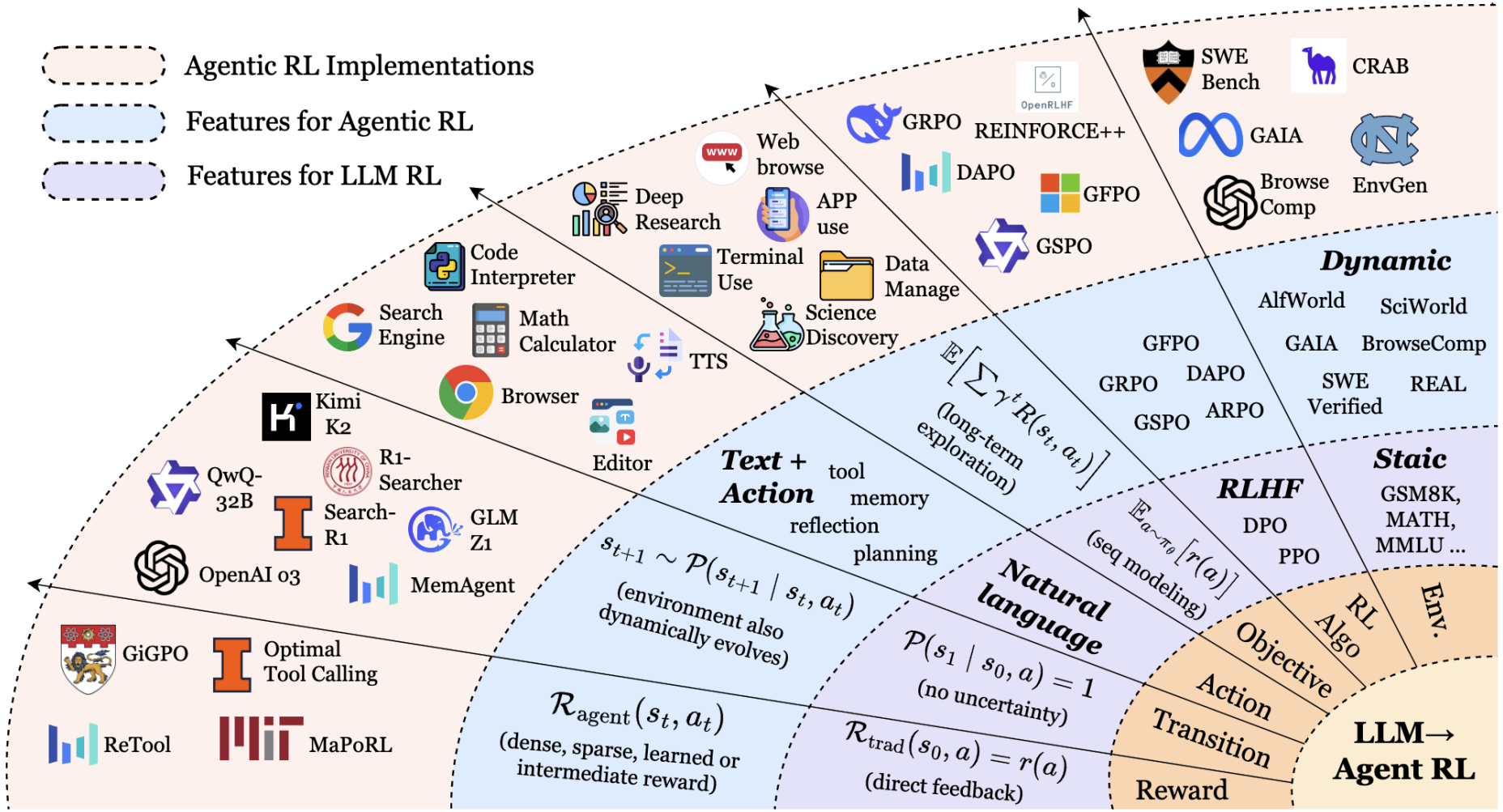


Figure 2: Paradigm shift from LLM-RL to agentic RL.

1.1 Markov Decision Processes

PBRFT. The RL training process of PBRFT is formalized as a degenerate MDP defined by the tuple:

$$\langle \mathcal{S}_{\text{trad}}, \mathcal{A}_{\text{trad}}, \mathcal{P}_{\text{trad}}, \mathcal{R}_{\text{trad}}, T = 1 \rangle. \tag{1}$$

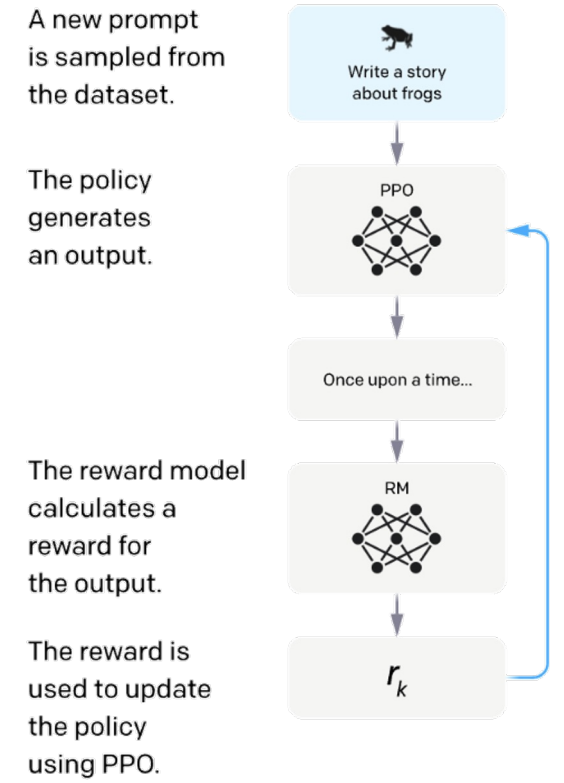
Agentic RL. The RL training process of agentic RL is modeled as a POMDP:

$$\langle \mathcal{S}_{\text{agent}}, \mathcal{A}_{\text{agent}}, \mathcal{P}_{\text{agent}}, \mathcal{R}_{\text{agent}}, \gamma, \mathcal{O} \rangle, \tag{2}$$

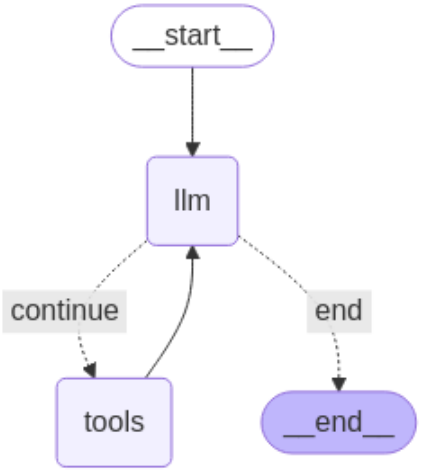
where the agent receives observations $o_t = O(s_t)$ based on the state $s_t \in \mathcal{S}_{\text{agent}}$. The primary distinctions between PBRFT and agentic RL are delineated in Table 1. In summary, PBRFT optimizes sequences of output sentences within a fixed dataset under full observations, whereas agentic RL optimizes semantic-level behaviors in variable environments characterized by partial observations.

Table 1: Formal comparison between traditional PBRFT and Agentic RL.

Concept	Traditional PBRFT	Agentic RL
$\mathcal{S}$: State space	$\{s_0\}$ (single prompt); episode ends immediately.	$s_t \in \mathcal{S}_{\text{agent}}$; $o_t = O(s_t)$; horizon $T > 1$.
$\mathcal{A}$: Action space	Pure text sequence.	$\mathcal{A}_{\text{text}} \cup \mathcal{A}_{\text{action}}$.
$\mathcal{P}$: Transition	Deterministic to the terminal state.	Dynamic transition function $P(s_{t+1} \mid s_t, a_t)$.
$\mathcal{R}$: Reward	Single scalar $r(a)$.	Step-wise $R(s_t, a_t)$; combines sparse task and dense sub-rewards.
$J(\theta)$: Objective	$\mathbb{E}_{a \sim \pi_\theta}[r(a)]$.	$\mathbb{E}_{\tau \sim \pi_\theta}[\sum_t \gamma^t R(s_t, a_t)]$.



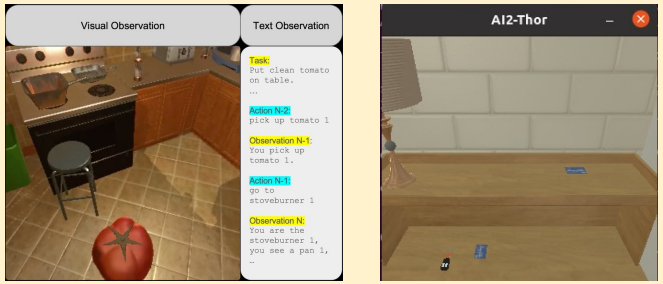
DPO for RLHF



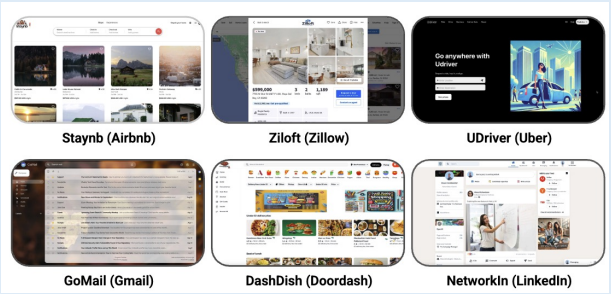
ReAct-style Long-horizon RL

1.2 Environment

Embodied (ALFWorld, SciWorld)



Web (WebShop, WebArena)



Real World (GAIA, xBench-DR)

2.2. Environment State

PBRFT. In the training process, each episode starts from a single prompt state s_0; the episode terminates immediately after the model emits one response. Formally, the underlying MDP degenerates to a *single-step* decision problem with horizon $T = 1$. The state space reduces to a single static prompt input:

$$\mathcal{S}_{\text{trad}} = \{\text{prompt}\}. \tag{3}$$

Agentic RL. The LLM agent acts over multiple time-steps in a POMDP. Let $s_t \in \mathcal{S}_{\text{agent}}$ denote the full world-state and the LLM agent gets observation O_t based on current state $o_t = \mathcal{O}(s_t)$. The LLM agent chooses an action a_t based on the current observation o_t, and the state evolves over time:

$$s_{t+1} \sim P(s_{t+1} \mid s_t, a_t), \tag{4}$$

as the agent accumulates intermediate signals such as retrieved tool results, user messages, or environment feedback. The interaction is thus inherently dynamic and temporally extended.

1.3 Action Space

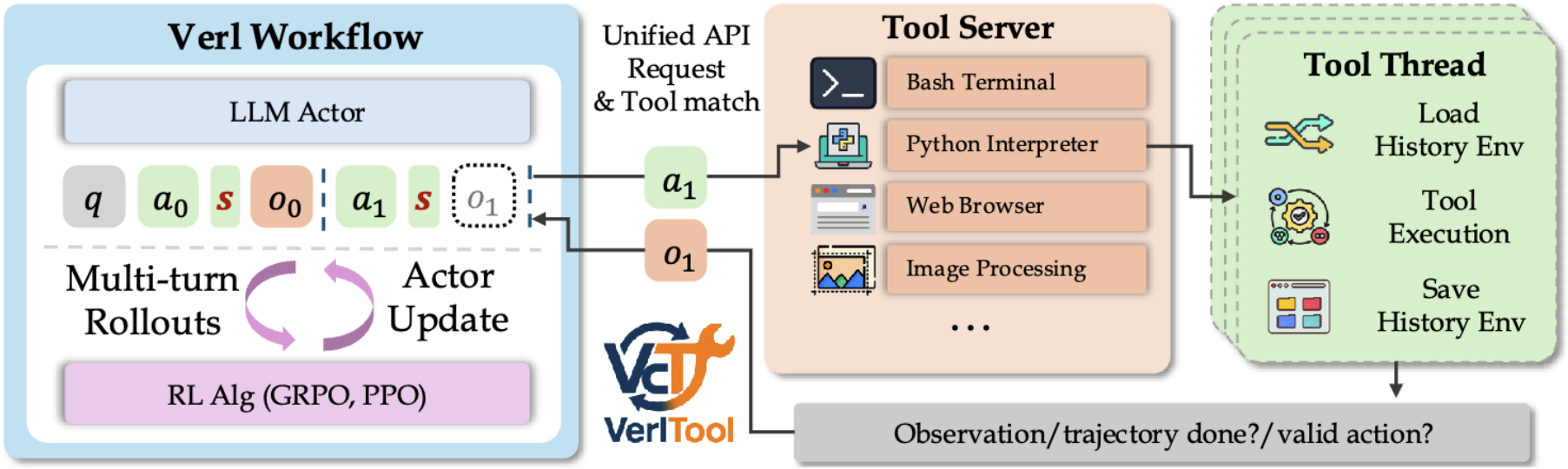


Figure 1: Overview of the VERLTOOL, a modularized and efficient framework for the Agentic Reinforcement Learning with Tool Use (ARLT) training paradigm, where the RL workflow and tool execution are fully disaggregated for both efficiency and extensibility.

Jiang D, Lu Y, Li Z, et al. VerlTool: Towards Holistic Agentic Reinforcement Learning with Tool Use[J]. arXiv preprint arXiv:2509.01055, 2025.

1.4 Transition Dynamics

PBRFT. In conventional PBRFT, the transition dynamics are deterministic: the next state is determined once an action is made, as follows:

$$\mathcal{P}(s_1 \mid s_0, a) = 1, \quad \text{where there is no uncertainty.} \tag{6}$$

Agentic RL. In agentic RL, the environment evolves under uncertainty according to

$$s_{t+1} \sim \mathcal{P}(s_{t+1} \mid s_t, a_t), \quad a_t \in \mathcal{A}_{\text{text}} \cup \mathcal{A}_{\text{action}}. \tag{7}$$

Text actions ($\mathcal{A}_{\text{text}}$) generate natural language outputs without altering the environment state. Structured actions ($\mathcal{A}_{\text{action}}$), delimited by `<action_start>` and `<action_end>`, can either query external tools or directly modify the environment. This sequential formulation contrasts with the one-shot mapping of PBRFT, enabling policies that iteratively combine communication, information acquisition, and environment manipulation.

1.5 Learning Objective

2.6. Learning Objective

PBRFT. The optimization objective of PBRFT is to maximize the response reward based on the policy π_θ:

$$J_{\text{trad}}(\theta) = \mathbb{E}_{a\sim\pi_\theta}\big[r(a)\big]. \tag{10}$$

No discount factor is required; optimization resembles maximum-expected-reward sequence modeling.

Agentic RL. The optimization objective of Agentic RL is to maximize the discounted reward:

$$J_{\text{agent}}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1}\gamma^t R_{\text{agent}}(s_t,a_t)\right], \qquad 0<\gamma<1, \tag{11}$$

optimized via policy-gradient or value-based methods with exploration and long-term credit assignment.

PBRFT focuses on single-turn text quality alignment without explicit planning, tool use, or environment feedback, while agentic RL involves multi-turn planning, adaptive tool invocation, stateful memory, and long-horizon credit assignment, enabling the LLM to function as an autonomous decision-making agent.

1.6 RL Algorithms

REINFORCE

$$\nabla_\theta J(\theta) = \mathbb{E}_{s_0}\left[\frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{R}(s_0, a^{(i)}) - b(s_0)\right)\nabla_\theta \log \pi_\theta(a^{(i)}|s_0)\right],$$

Proximal Policy Optimization

$$L_{PPO}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\min\left(\frac{\pi_\theta(a_t^{(i)}|s_t)}{\pi_{\theta_{old}}(a_t^{(i)}|s_t)}A(s_t, a_t^{(i)}),\ \text{clip}\left(\frac{\pi_\theta(a_t^{(i)}|s_t)}{\pi_{\theta_{old}}(a_t^{(i)}|s_t)}, 1-\epsilon, 1+\epsilon\right)A(s_t, a_t^{(i)})\right)$$

Direct Preference Optimization

$$L_{DPO}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x, y_w, y_l)\sim D}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}\right)\right]$$

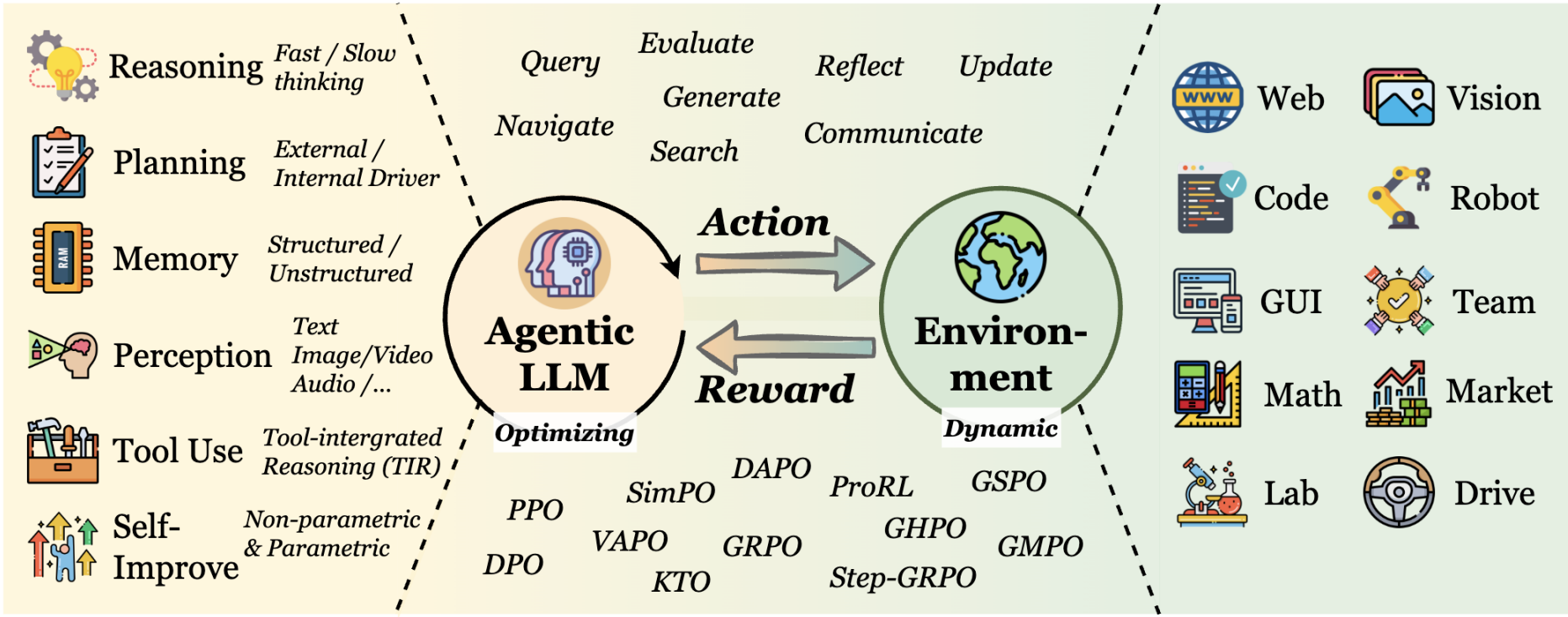
Group Relative Policy Optimization

$$L_{GRPO} = \frac{1}{G}\sum_{g=1}^{G}\min\left(\frac{\pi_\theta(a_t^{(g)}|s_t^{(g)})}{\pi_{\theta_{old}}(a_t^{(g)}|s_t^{(g)})}\hat{A}(s_t^{(g)}, a_t^{(g)}),\ \text{clip}\left(\frac{\pi_\theta(a_t^{(g)}|s_t^{(g)})}{\pi_{\theta_{old}}(a_t^{(g)}|s_t^{(g)})}, 1-\epsilon, 1+\epsilon\right)\hat{A}(s_t^{(g)}, a_t^{(g)})\right),$$

1.6 RL Algorithms

Method	Year	Objective Type	Clip	KL Penalty	Key Mechanism	Signal
			PPO family			
PPO [12]	2017	Policy gradient	Yes	No	Policy ratio clipping	Reward
VAPO [37]	2025	Policy gradient	Yes	Adaptive	Adaptive KL penalty + variance control	Reward + variance signal
PF-PPO [38]	2024	Policy gradient	Yes	Yes	Policy filtration	Noisy reward
VinePPO [39]	2024	Policy gradient	Yes	Yes	Unbiased value estimates	Reward
PSGPO [40]	2024	Policy gradient	Yes	Yes	Process supervision	Process Reward
			DPO family			
DPO [29]	2024	Preference optimization	No	Yes	Implicit reward related to the policy	Human preference
β-DPO [41]	2024	Preference optimization	No	Adaptive	Dynamic KL coefficient	Human preference
SimPO [42]	2024	Preference optimization	No	Scaled	Use the average log probability of a sequence as the implicit reward	Human preference
IPO [35]	2024	Implicit preference	No	No	Leverage generative LLMs as preference classifiers for reducing the dependence on external human feedback or reward models	Preference rank
KTO [36]	2024	Knowledge transfer optimization	No	Yes	Teacher stabilization	Teacher-student logit
ORPO [43]	2024	Online regularized preference optimization	No	Yes	Online stabilization	Online feedback reward
Step-DPO [44]	2024	Preference optimization	No	Yes	Step-wise supervision	Step-wise preference
LCPO [45]	2025	Preference optimization	No	Yes	Length preference with limited data and training	Reward
			GRPO family			
GRPO [31]	2025	Policy Gradient under group-based reward	Yes	Yes	Group-based relative reward to eliminate value estimates	Group-based reward
DAPO [46]	2025	Surrogate of GRPO's	Yes	Yes	Decoupled clip and dynamic sampling	Dynamic group-based reward
GSPO [47]	2025	Surrogate of GRPO's	Yes	Yes	Define the importance ratio based on sequence likelihood and performs sequence-level clipping, rewarding, and optimization	Smooth group-based reward
GMPO [48]	2025	Surrogate of GRPO's	Yes	Yes	Geometric mean of token-level rewards	Margin-based reward
ProRL [49]	2025	Same as GRPO's	Yes	Yes	Reference policy reset	Group-based reward
Posterior-GRPO [50]	2025	Same as GRPO's	Yes	Yes	Reward only successful processes	Process-based reward
Dr.GRPO [51]	2025	Unbiased GRPO's objective	Yes	Yes	Eliminate the bias in optimization of GRPO	Group-based reward
Step-GRPO [52]	2025	Same as GRPO's	Yes	Yes	Rule-based reasoning rewards	Step-wise reward
SRPO [53]	2025	Same as GRPO's	Yes	Yes	Two-staged history-resampling	Reward
GRESO [54]	2025	Same as GRPO's	Yes	Yes	Pre-rollout filtering	Reward
StarPO [55]	2025	Same as GRPO's	Yes	Yes	Reasoning-guided actions for multi-turn interactions	Group-based reward
GHPO [56]	2025	Policy gradient	Yes	Yes	Adaptive prompt refinement	Reward
Skywork R1V2 [57]	2025	GRPO's with hybrid reward signal	Yes	Yes	Selective sample buffer	Multimodal reward
ASPO [58]	2025	GRPO's with shaped advantage function	Yes	Yes	Apply a clipped bias directly to advantage function	Group-based reward
TreePo [59]	2025	Same as GRPO's	Yes	Yes	Self-guided policy rollout for reducing the compute burden	Group-based reward
EDGE-GRPO [60]	2025	Same as GRPO's	Yes	Yes	Entropy-driven advantage and duided error correction to mitigate advantage collapse	Group-based reward
DARS [61]	2025	Same as GRPO's	Yes	No	Reallocate compute from medium-difficulty to the hardest problems via multi-stage rollout sampling	Group-based reward
CHORD [62]	2025	Weighted sum of GRPO's and Supervised Fine-Tuning losses	Yes	Yes	Reframe Supervised Fine-Tuning as a dynamically weighted auxiliary objective within the on-policy RL process	Group-based reward
PAPO [63]	2025	Surrogate of GRPO's	Yes	Yes	Encourage learning to perceive while learning to reason through the Implicit Perception Loss	Group-based reward
Pass@k Training	2025	Same as GRPO's	Yes	Yes	Pass@k metric as the reward to continually train a model	Group-based reward

2.0 RL for Agentic Capacities



LLM Agent = LLM + Reasoning + Planning + Memory + Perception + Tool Use + Self-Improve

2.0 RL for Agentic Capacities

2.1 RL for Agent Planning

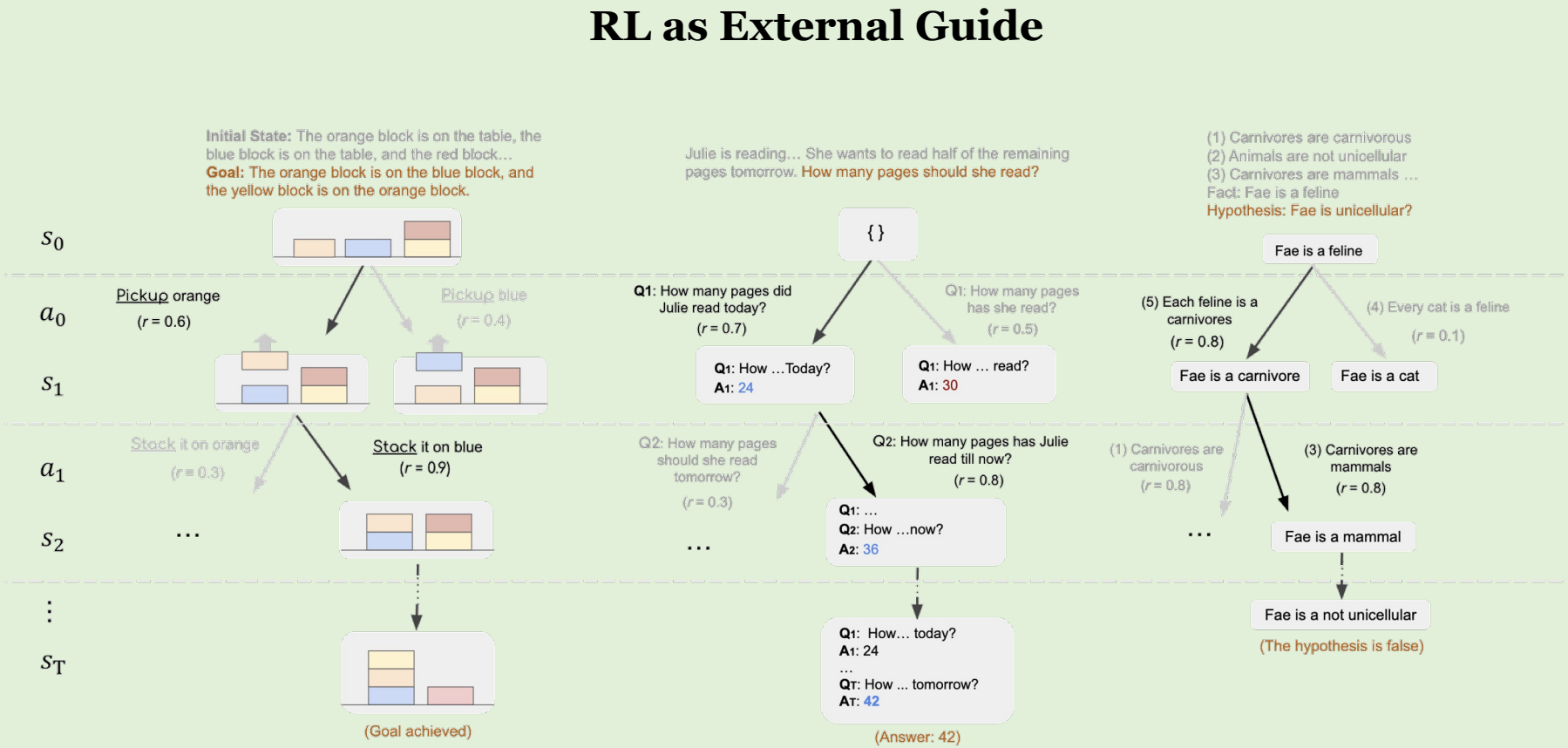
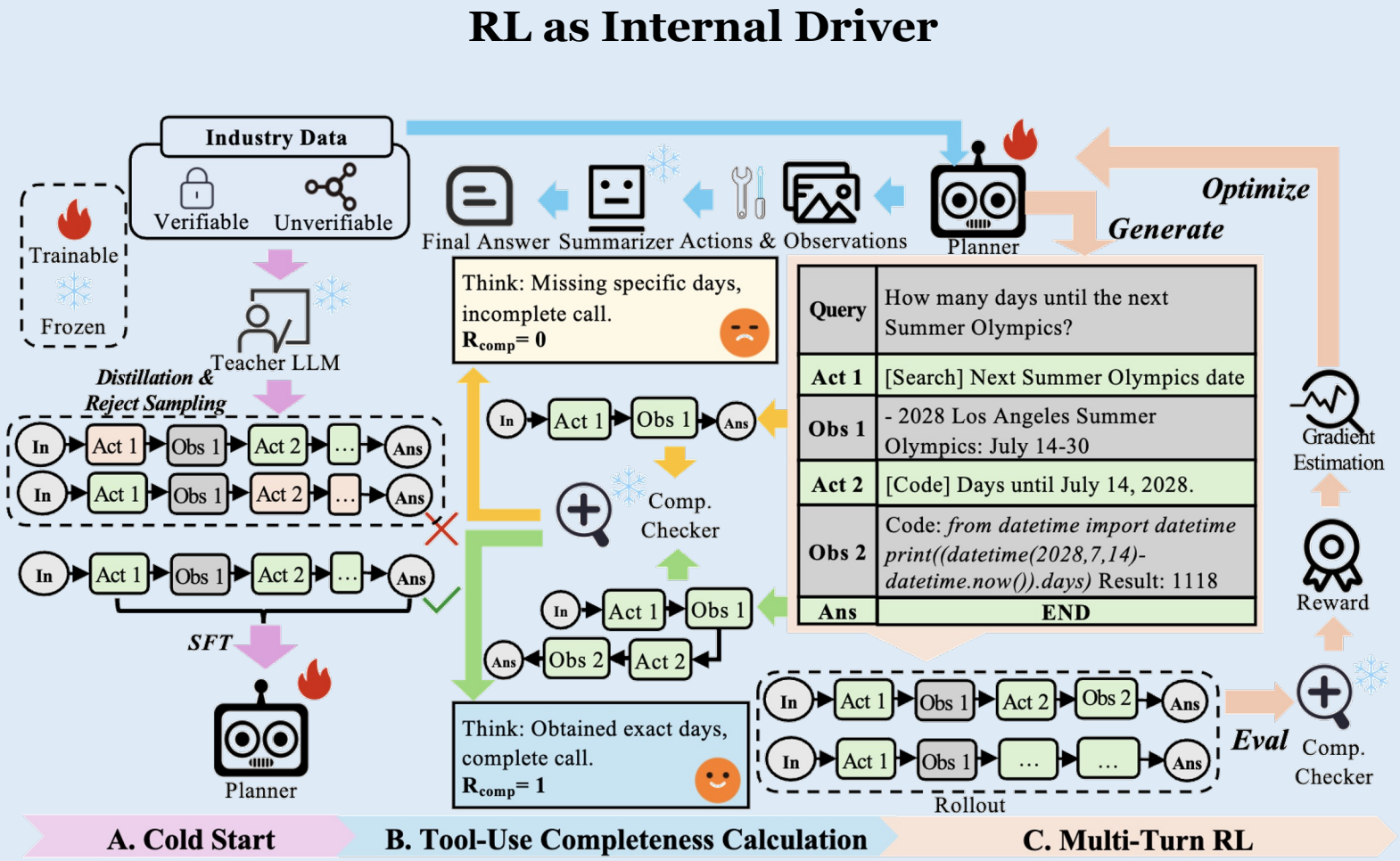


Figure 2: RAP for plan generation in Blocksworld (left), math reasoning in GSM8K (middle), and logical reasoning in PrOntoQA (right).

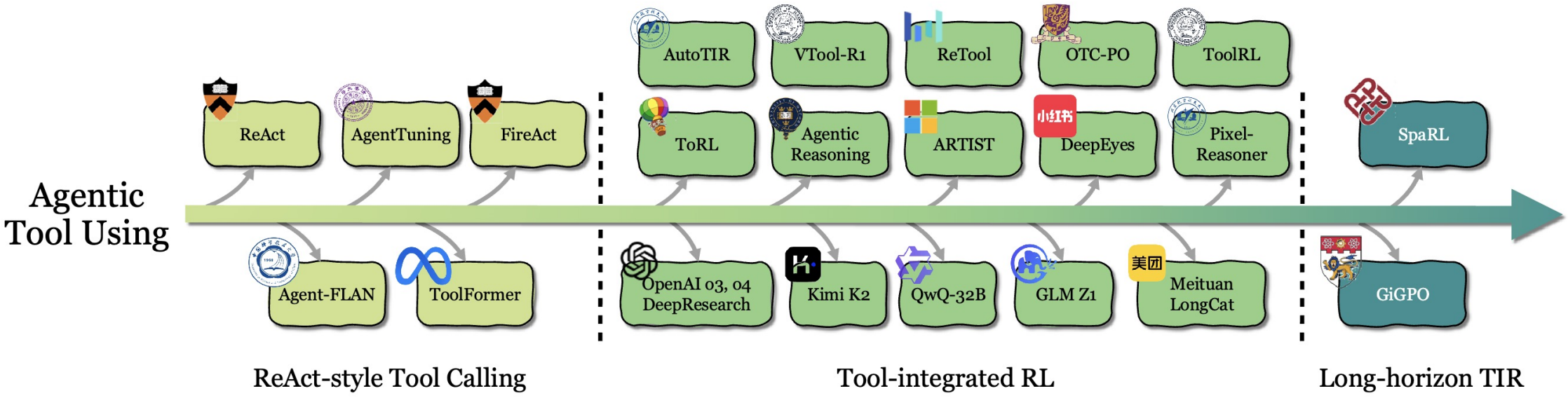
Hao S, Gu Y, Ma H, et al. Reasoning with language model is planning with world model[J]. arXiv preprint arXiv:2305.14992, 2023.

2.1 RL for Agent Planning



Li Z, Hu Y, Wang W. Encouraging Good Processes Without the Need for Good Answers: Reinforcement Learning for LLM Agent Planning[J]. arXiv preprint arXiv:2508.19598, 2025.

2.2 RL for Tool Use



2.2 RL for Tool Use

Phase 1: Prompt-based ReAct-style Tool Use 👉 SFT-empowered ReAct Agent

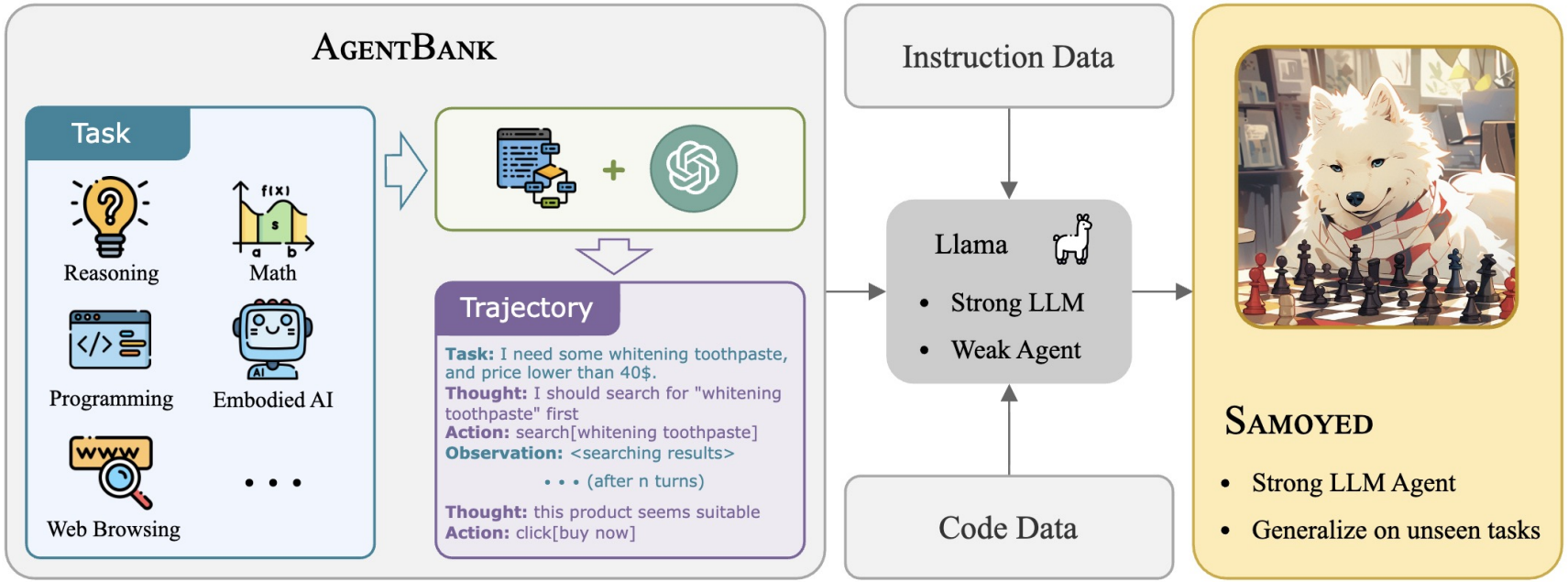


Figure 1: Overview of the construction process of AGENTBANK and the training procedure of SAMOYED

Prompt-based:
- ReAct

SFT-based:
- Agent Tuning
- Agent-FLAN
- AgentBank
- FireAct
- Toolformer

Song Y, Xiong W, Zhao X, et al. Agentbank: Towards generalized llm agents via fine-tuning on 50000+ interaction trajectories[J]. arXiv preprint arXiv:2410.07706, 2024.

2.2 RL for Tool Use

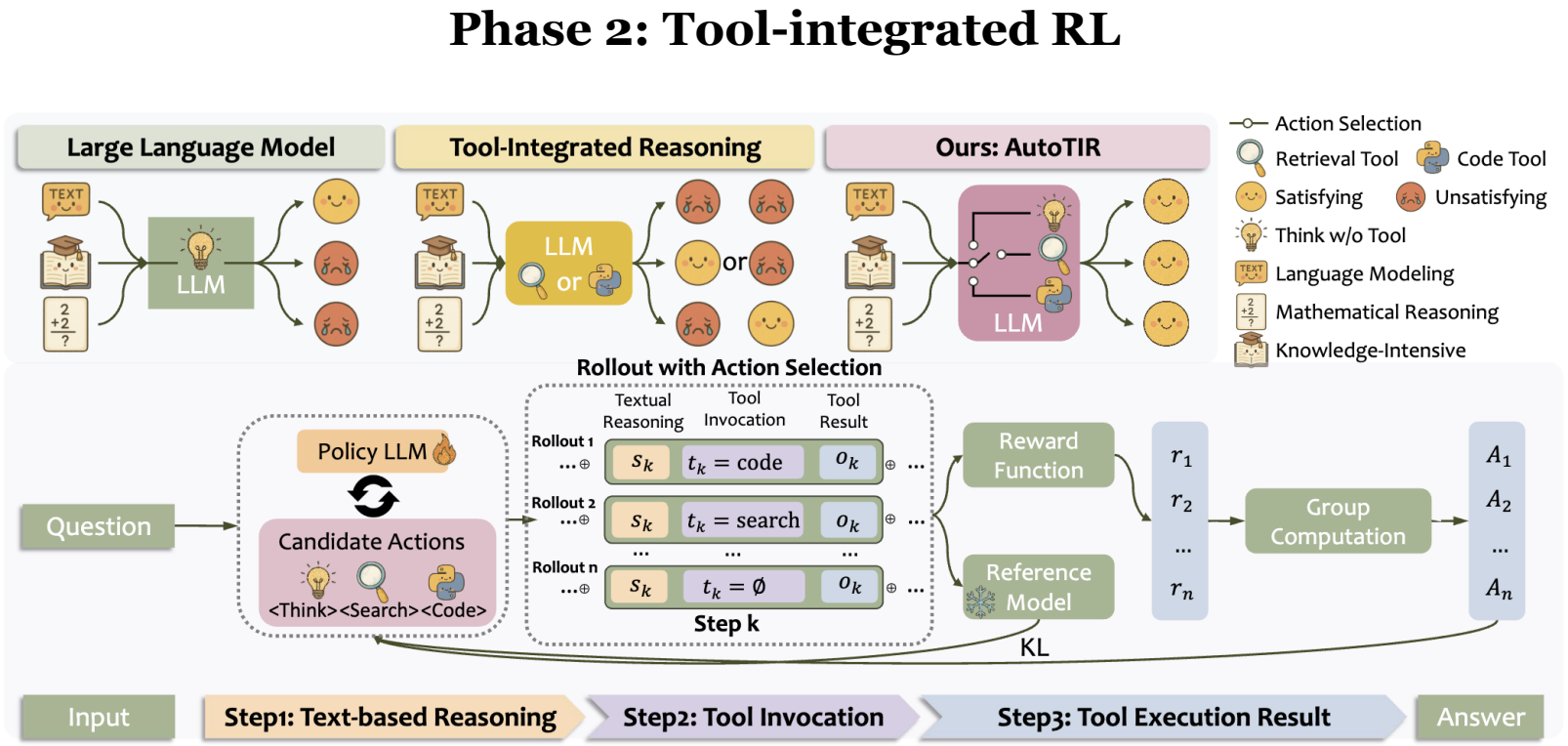
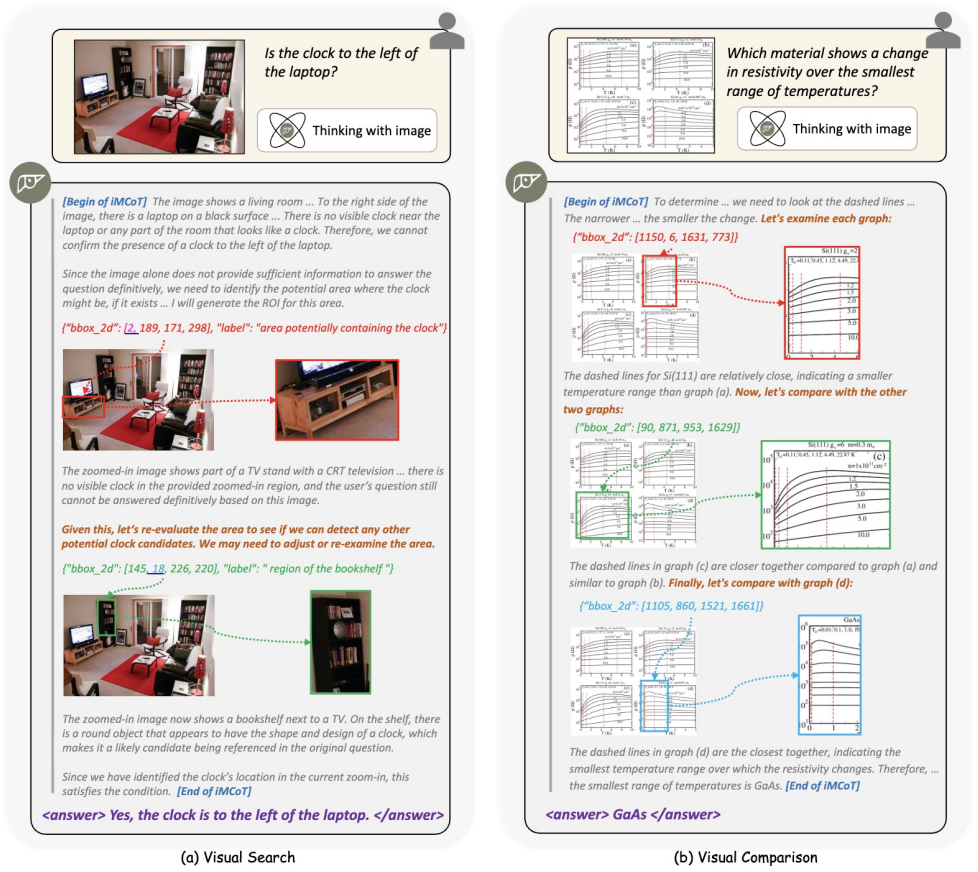


Figure 2: Overall framework of AutoTIR. **Top**: Comparison between AutoTIR and existing paradigms (fixed reasoning strategy vs. autonomous decision). **Bottom**: GRPO training pipeline that incorporates multiple reasoning actions.

Wei Y, Yu X, Weng Y, et al. AutoTIR: Autonomous Tools Integrated Reasoning via Reinforcement Learning[J]. arXiv preprint arXiv:2507.21836, 2025.

2.2 RL for Tool Use

Phase 2: Tool-integrated RL



Wang H, Zou H, Song H, et al. UI-TARS-2 Technical Report: Advancing GUI Agent with Multi-Turn Reinforcement Learning[J]. arXiv preprint arXiv:2509.02544, 2025.

2.2 RL for Tool Use

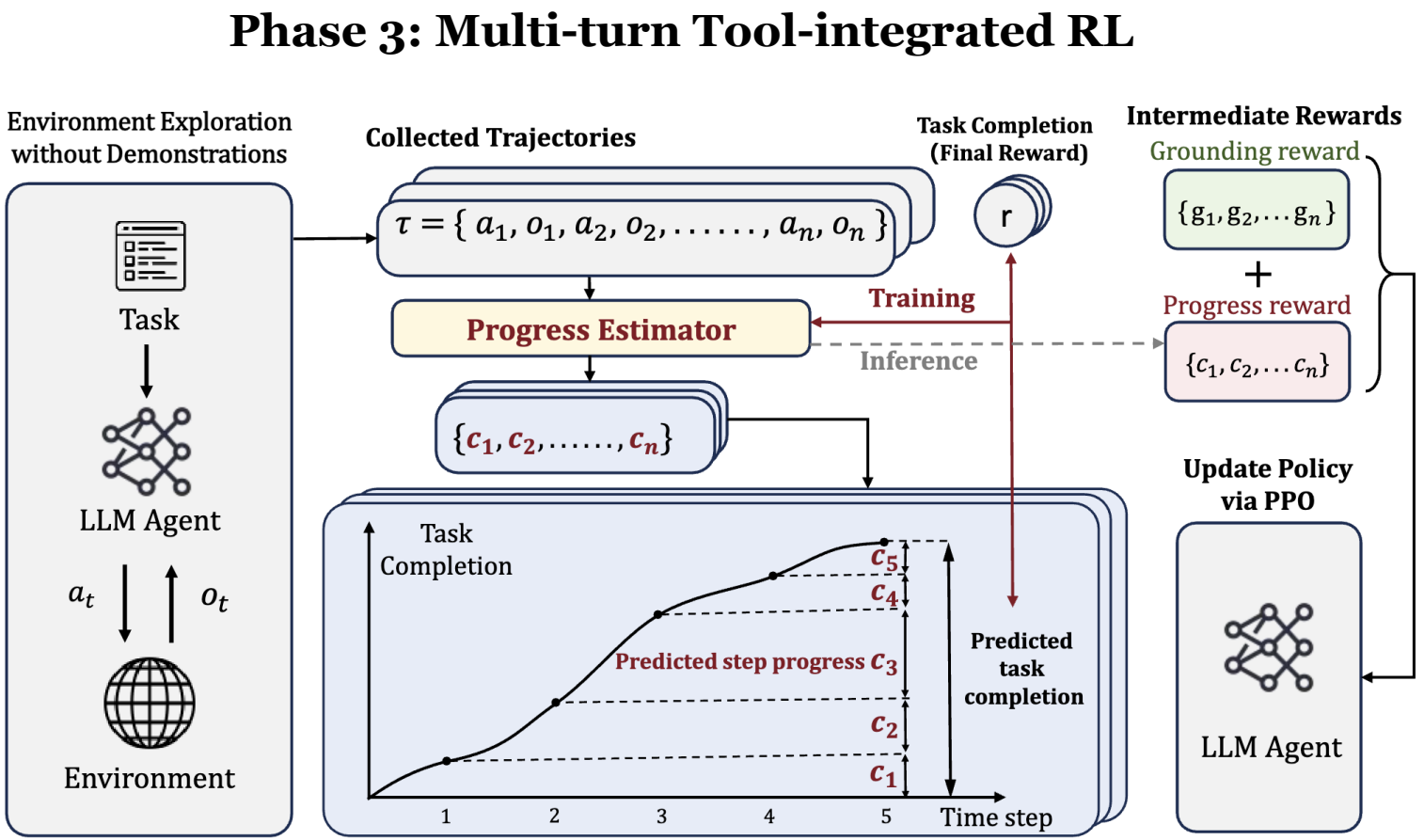


Figure 1: Overview of our proposed Stepwise Progress Attribution (SPA) framework for training LLM agents with RL.

Wang H, Leong C T, Wang J, et al. SPA-RL: Reinforcing LLM Agents via Stepwise Progress Attribution[J]. arXiv preprint arXiv:2505.20732, 2025.

2.3 RL for Agent Memory

Table 3: An overview of three classic categories of agent memory; works marked with † directly employ RL. The list here is not exhaustive, and we refer readers intersted in borader agent memory to [112].

Method	Type	Key Characteristics
	RAG-style Memory	
MemoryBank [113]	External Store	Static memory with predefined storage/retrieval rules
MemGPT [114]	External Store	OS-like agent with static memory components
HippoRAG [115]	External Store	Neuro-inspired memory with heuristic access
Prospect† [116]	RL-Guided Retrieval	Uses RL for reflection-driven retrieval adjustment
Memory-R1† [117]	RL-Guided Retrieval	RL-driven memory management: ADD/UPDATE/DELETE/NOOP
	Token-level Memory	
MemAgent† [118]	Explicit Token	RL controls which NL tokens to retain or overwrite
MEM1† [119]	Explicit Token	Memory pool managed by RL to enhance context handling
Memory Token [120]	Explicit Token	Structured memory for reasoning disentanglement
MemoryLLM [121]	Latent Token	Latent tokens repeatedly integrated and updated
M+ [122]	Latent Token	Scalable memory tokens for long-context tracking
IMM [123]	Latent Token	Decouples word representations and latent memory
Memory [124]	Latent Token	Forget-resistant memory tokens for evolving context
	Structured Memory	
Zep [125]	Temporal Graph	Temporal knowledge graph enabling structured retrieval
A-MEM [126]	Atomic Memory Notes	Symbolic atomic memory units; structured storage
G-Memory [127]	Hierarchical Graph	Multi-level memory graph with topological structure
Mem0 [128]	Structured Graph	Agent memory with full-stack graph-based design

2.3.1 RL for RAG-style Memory

RMM

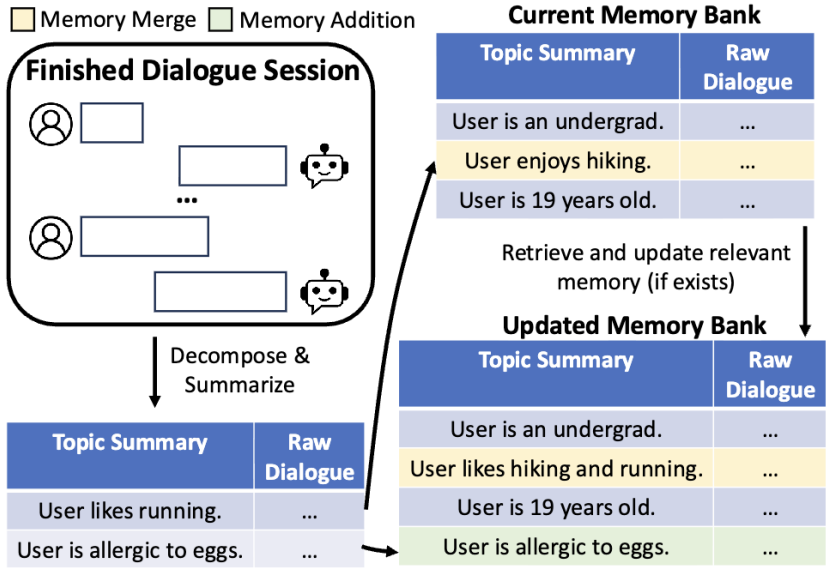


Figure 2 | Illustration of *Prospective Reflection*. After each session, the agent decomposes and summarizes the session into specific topics. These newly generated memories are compared with existing memories in the memory bank. Relevant memories are merged, while others are directly added. Prospective reflection ensures efficient organization of personal knowledge for future retrieval.

Memory R1

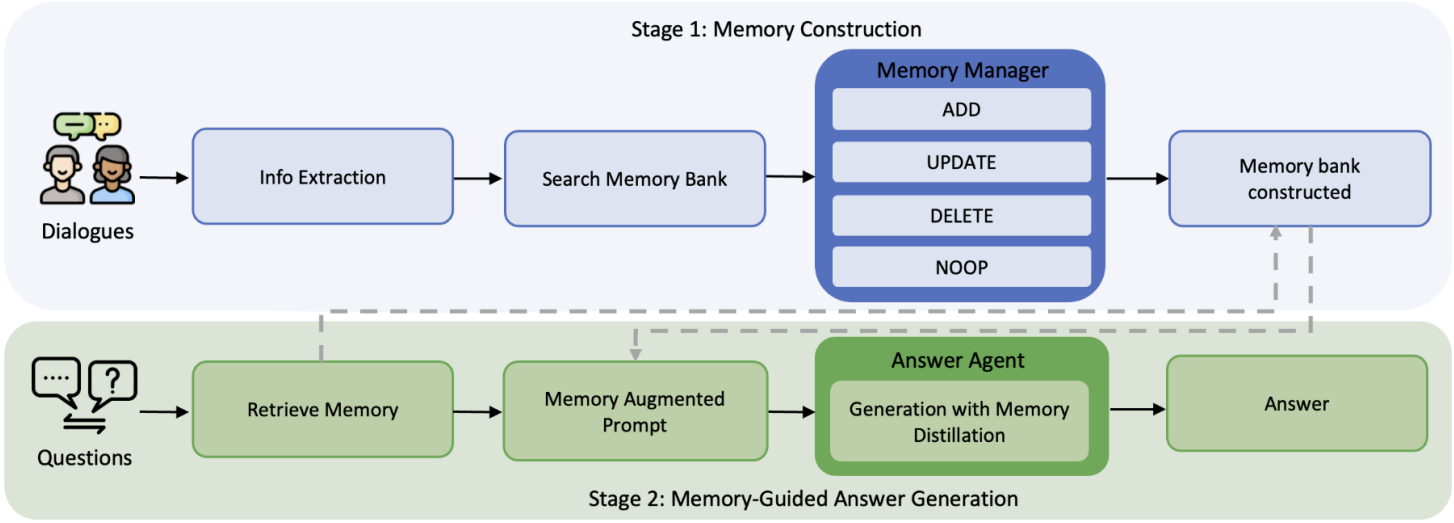
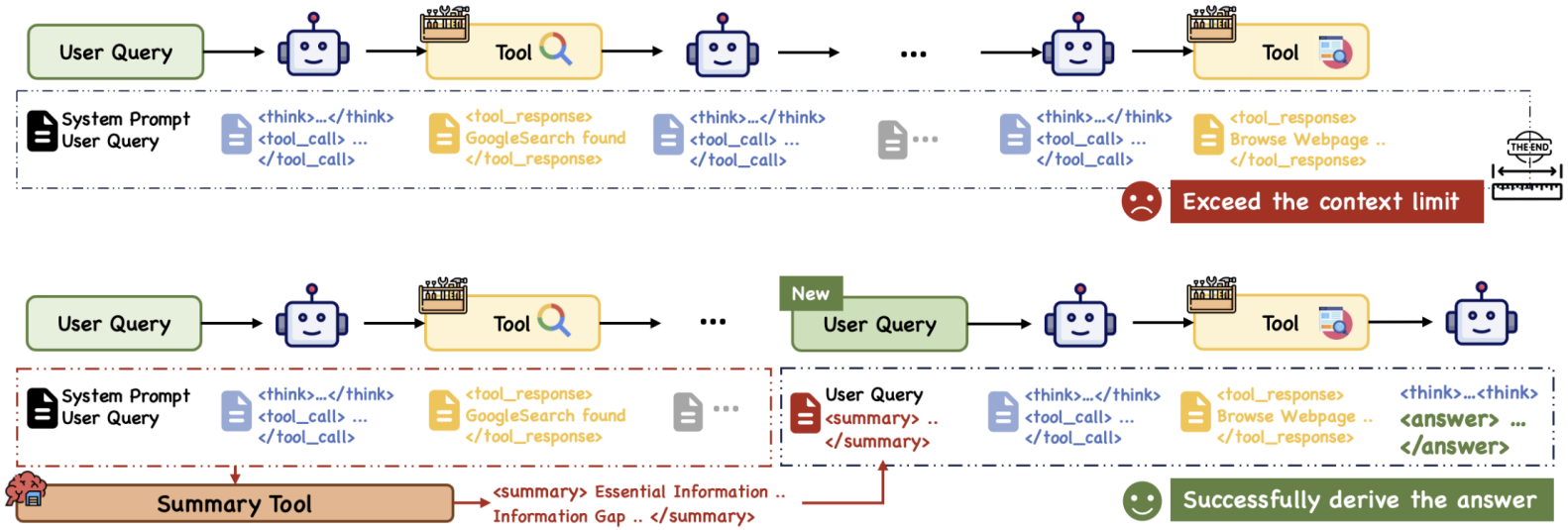


Figure 2: Overview of the Memory-R1 framework. Stage 1 (blue) constructs and updates the memory bank via the RL-fine-tuned Memory Manager, which chooses operations {ADD, UPDATE, DELETE, NOOP} for each new dialogue turn. Stage 2 (green) answers user questions via the Answer Agent, which applies a Memory Distillation policy to reason over retrieved memories.

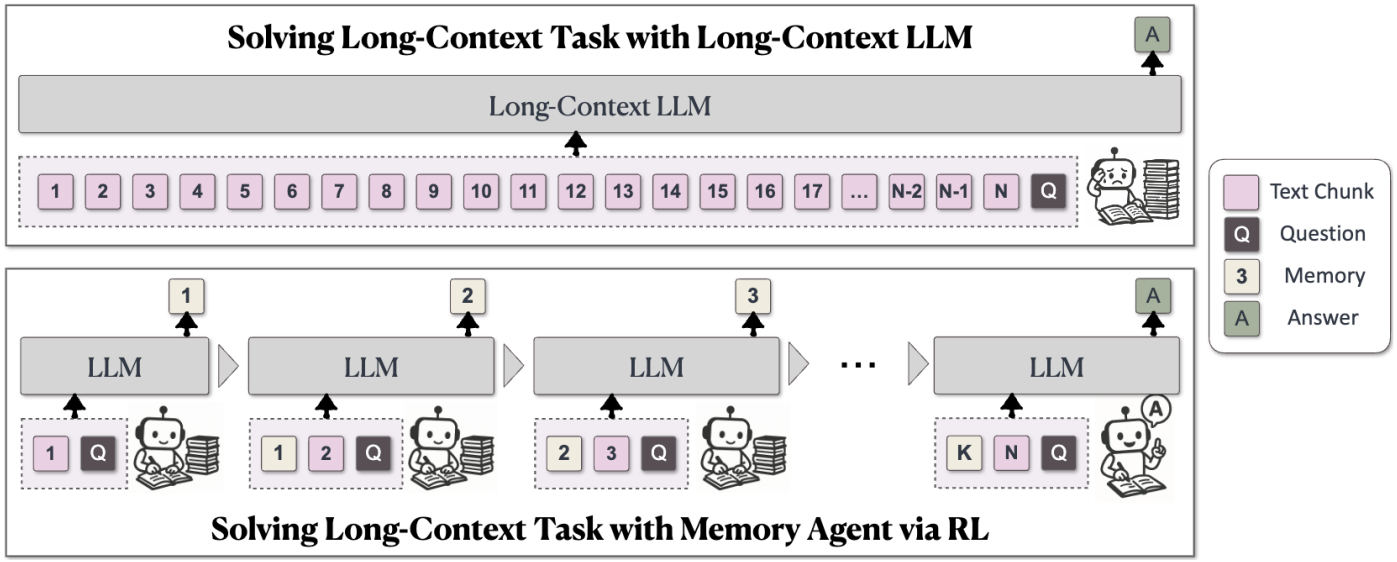
Tan Z, Yan J, Hsu I, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents[J]. arXiv preprint arXiv:2503.08026, 2025.

Yan S, Yang X, Huang Z, et al. Memory-R1: Enhancing Large Language Model Agents to Manage and Utilize Memories via Reinforcement Learning[J]. arXiv preprint arXiv:2508.19828, 2025.

2.3.2 RL for Token-Level Memory



阿里通义
ReSum



字节Seed
MemAgent

2.3.3 RL for Structured Memory

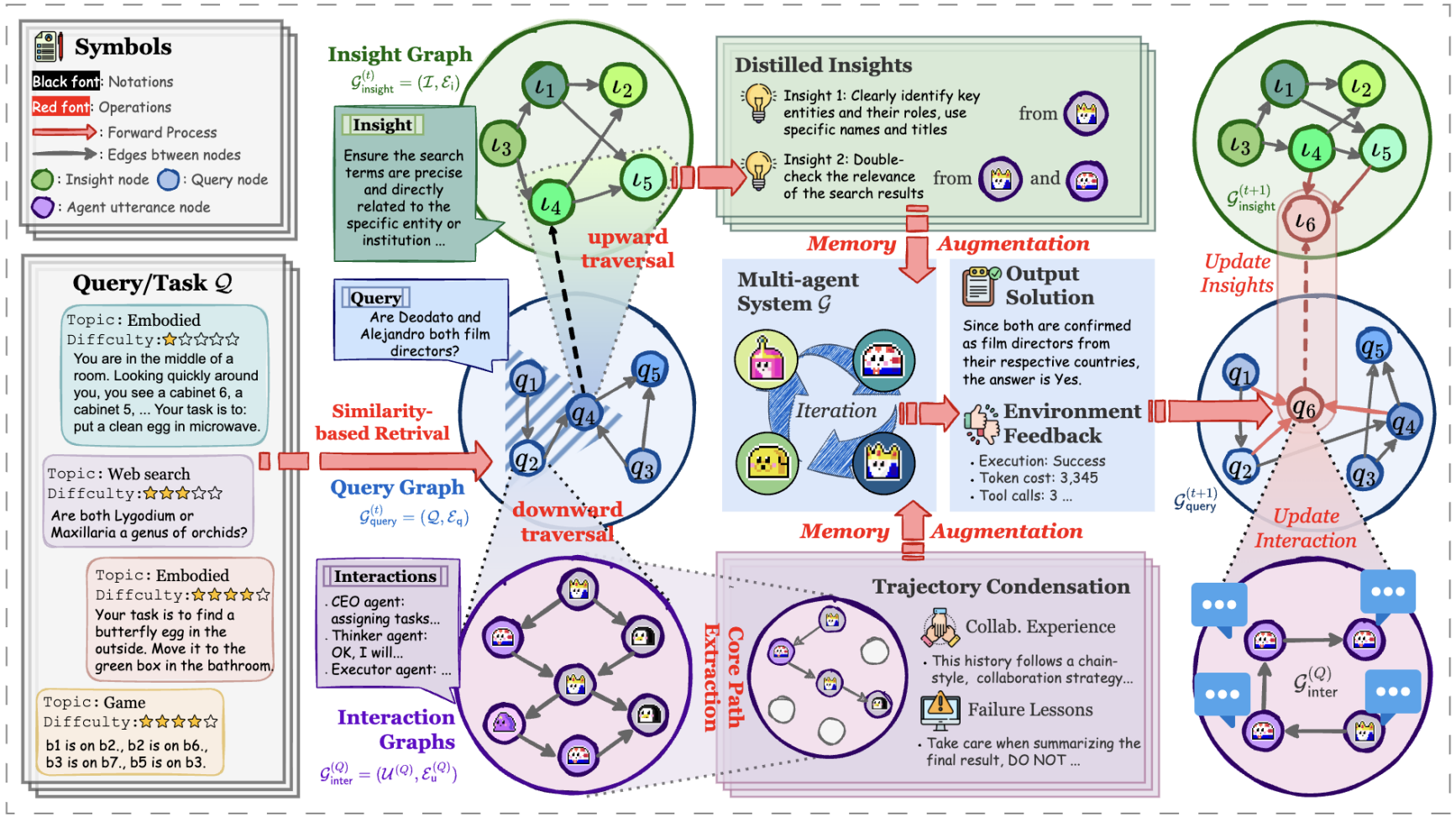
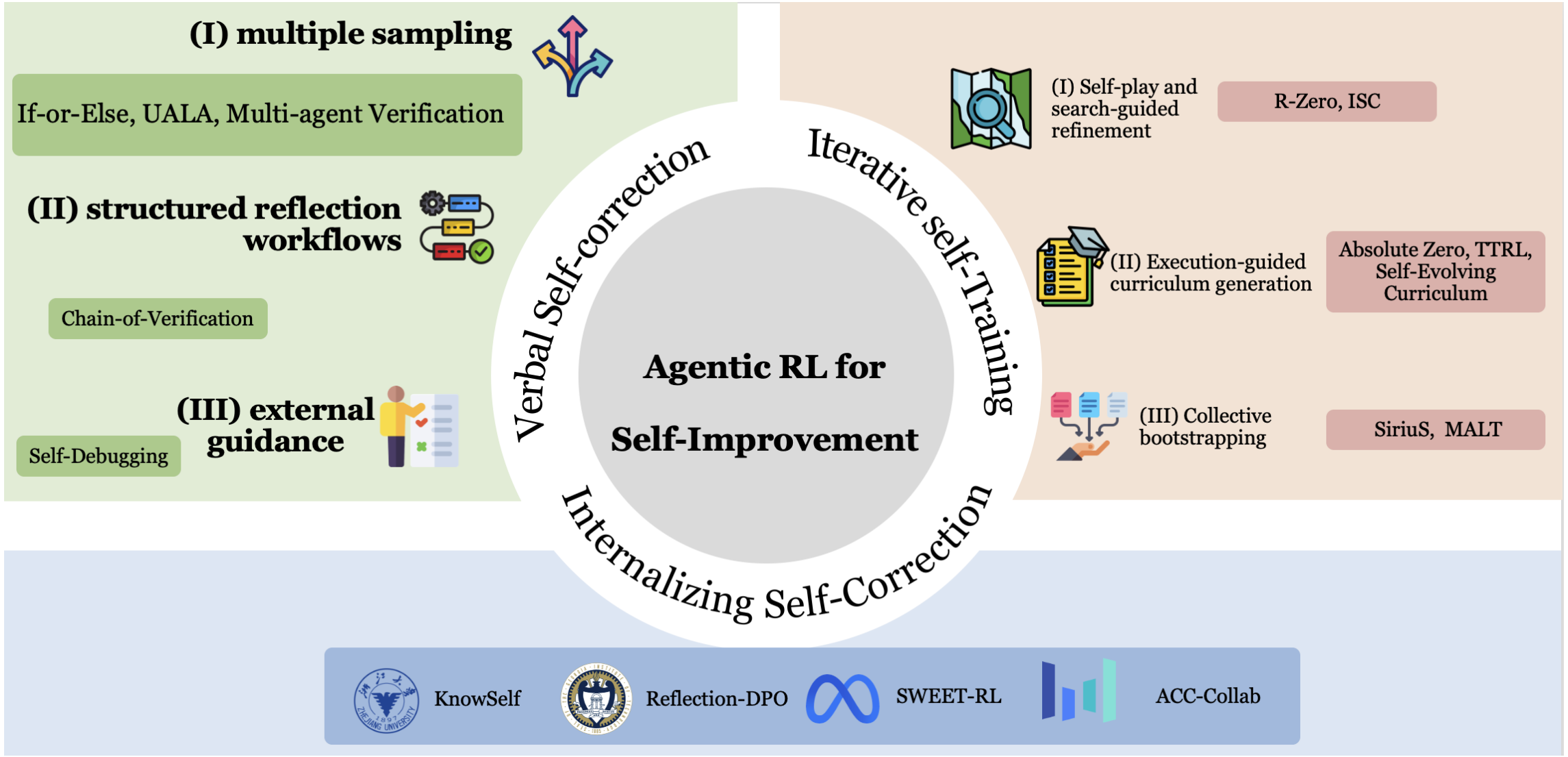


Figure 2: The overview of our proposed **G-Memory**.

2.4 RL for Agent Self-Improving



2.4 RL for Agent Self-Improving

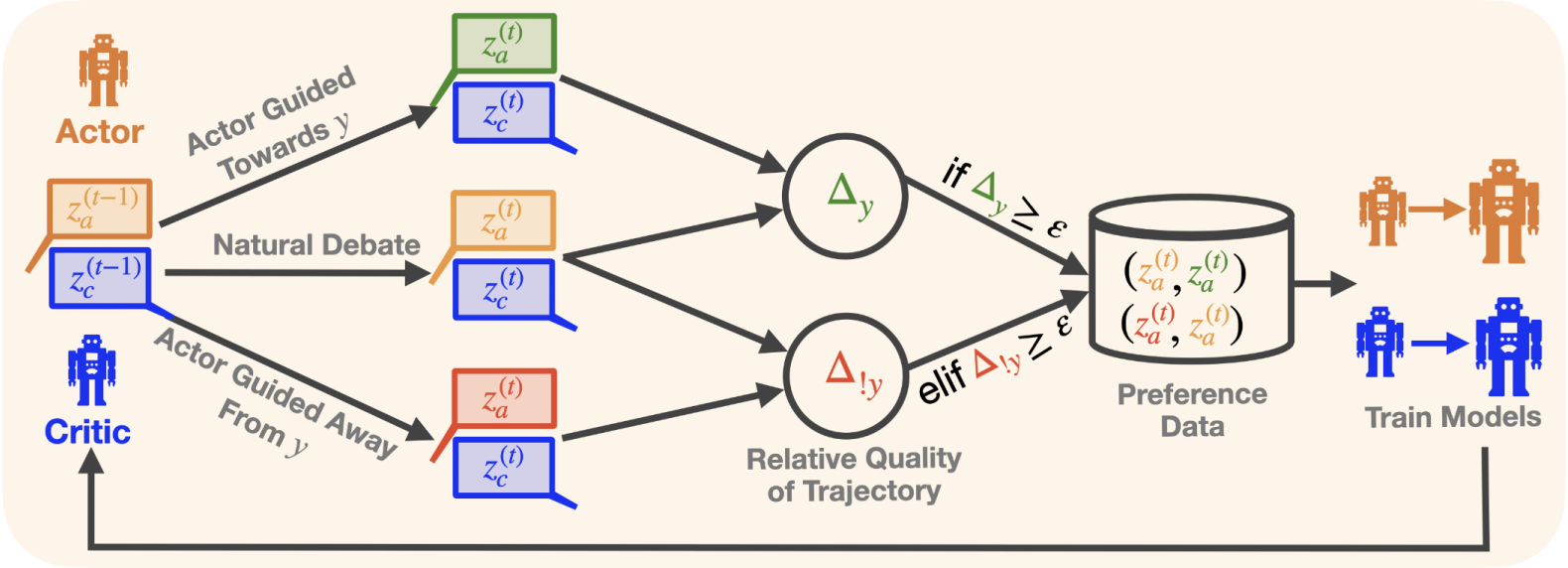


Figure 1: ACC-Collab training pipeline, exemplified for the actor. 1) We generate data from both natural deliberation as well as guided deliberation towards and away from the ground truth answer y using the actor and critic. 2) We compute the relative quality of each trajectory based on the expected quality difference $\Delta_y, \Delta_{!y}$ w.r.t. to the natural response. 3) We store all high-quality pairwise data in our database and train the actor agent. 4) We alternate this procedure for the actor and critic. See Figure 5 of the supplement for the corresponding procedure applied to the critic.

Estornell A, Ton J F, Yao Y, et al. ACC-collab: An actor-critic approach to multi-agent LLM collaboration[J]. arXiv preprint arXiv:2411.00053, 2024.

2.4 RL for Agent Self-Improving

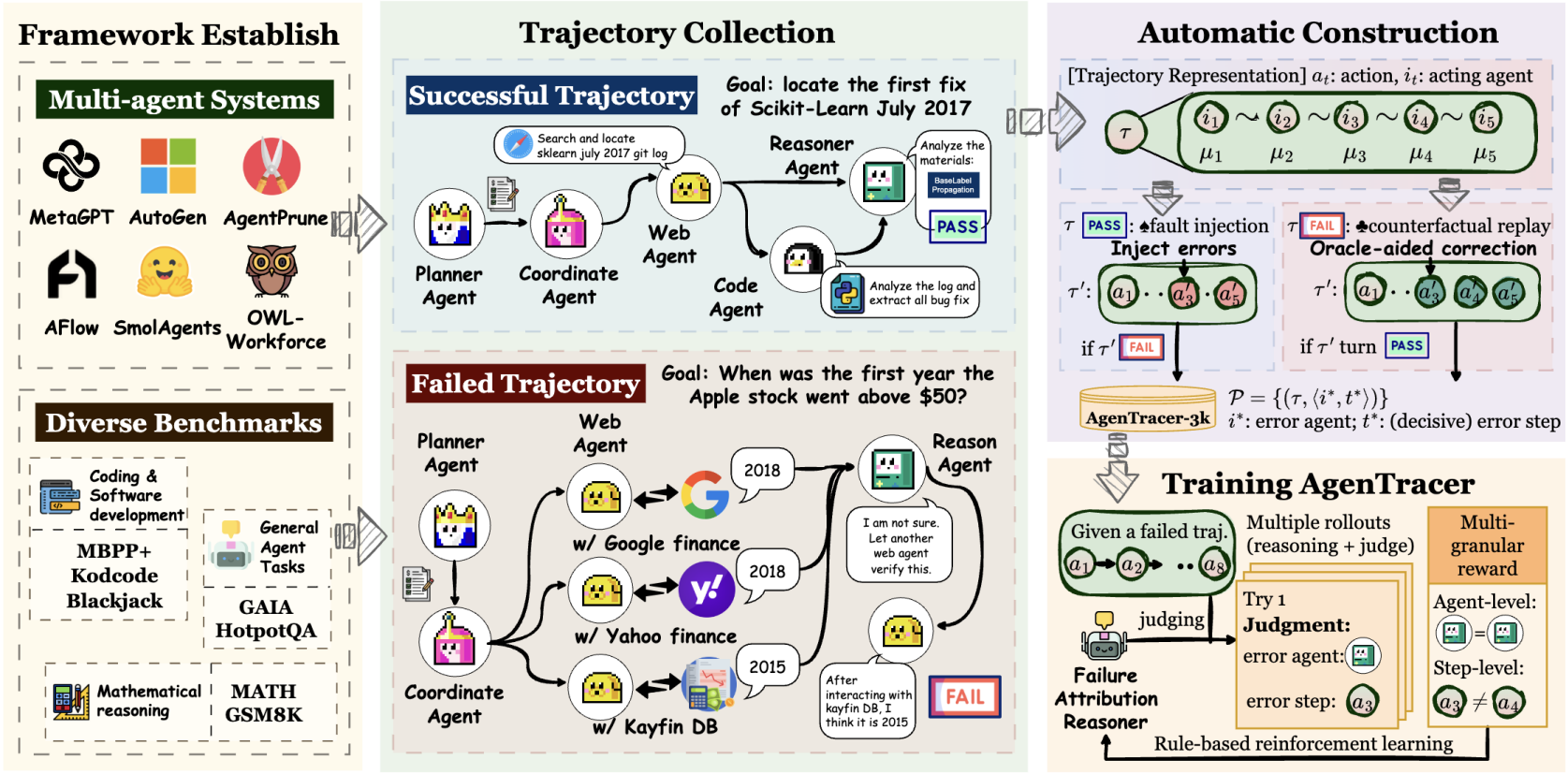


Figure 2: The overview of our proposed **AgenTracer**.

Zhang G, Wang J, Chen J, et al. AgenTracer: Who Is Inducing Failure in the LLM Agentic Systems?[J]. arXiv preprint arXiv:2509.03312, 2025.

2.4 RL for Agent Self-Improving

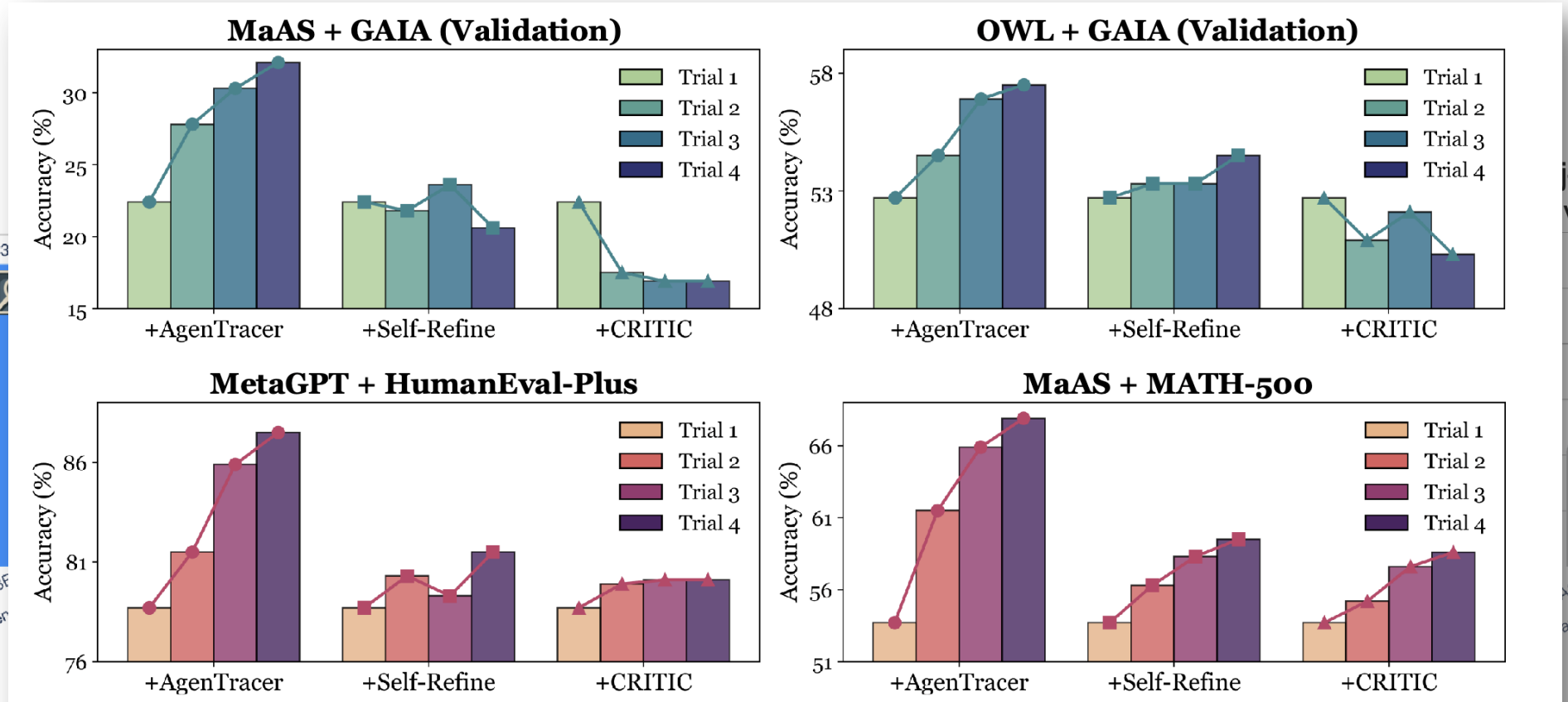


Figure 3: The multi-turn improvement performance brought by **AgenTracer-8B** compared with classical agent reflection baselines, Self-Refine, and CRITIC.

Zhang G, Wang J, Chen J, et al. AgenTracer: Who Is Inducing Failure in the LLM Agentic Systems?[J]. arXiv preprint arXiv:2509.03312, 2025.

2.4 RL for Agent Self-Improving

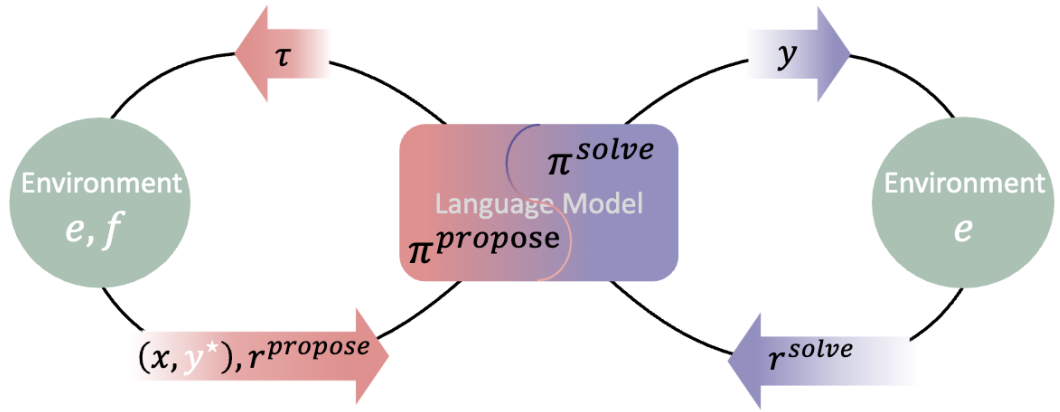
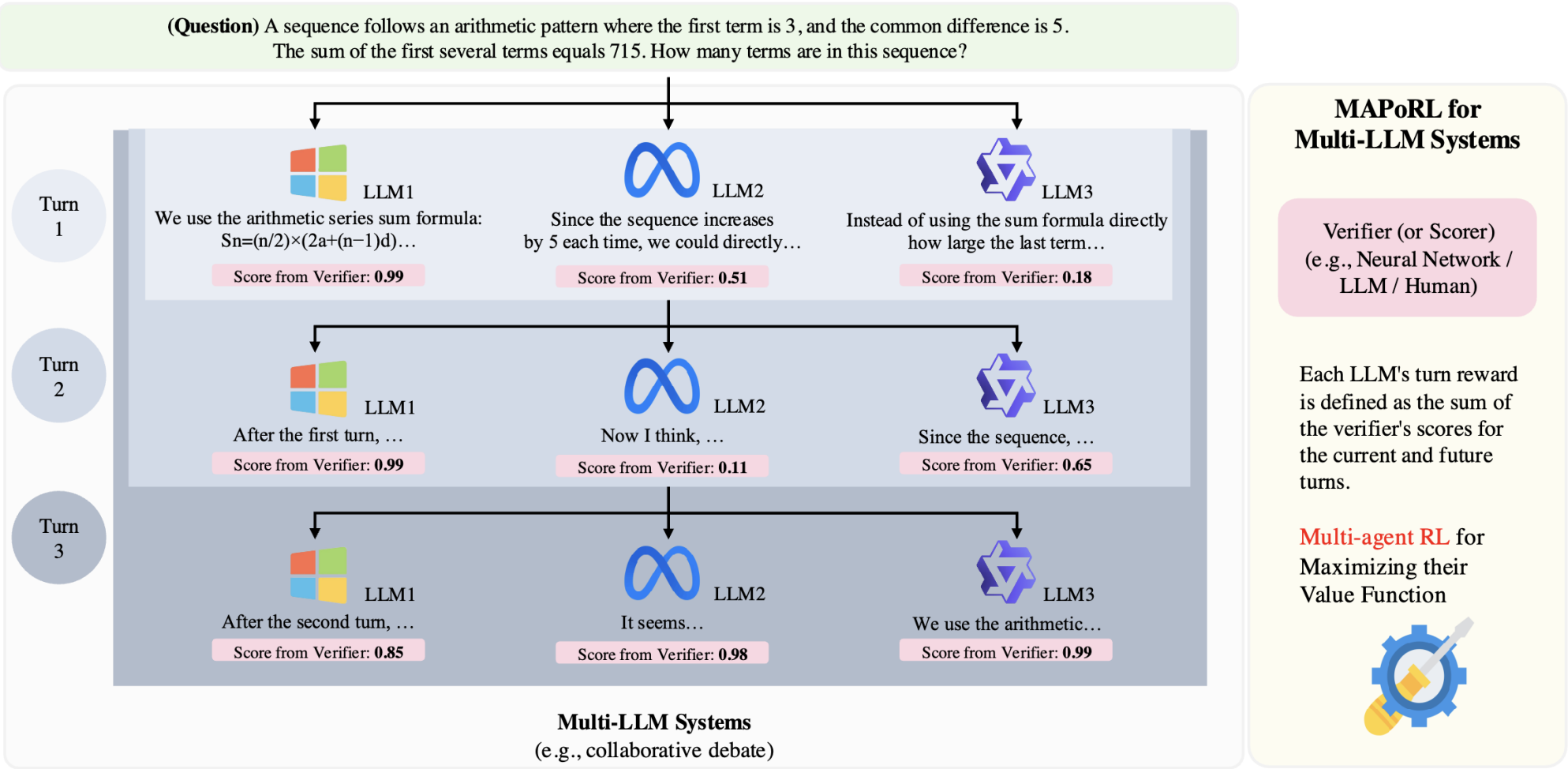


Figure 3. **The Absolute Zero Loop.** The Absolute Zero loop begins with the agent π proposing task τ, which is transformed by f with the environment e into a validated problem $(x, y^\star)$, and also emits a reward r^{propose} for learnability. Then, a standard RL step follows: the agent solves x by producing y, receiving reward r^{solve} from e by matching with $y^\star$. π^{propose} and π^{solve} are jointly trained and this process can be repeated indefinitely.

Zhao A, Wu Y, Yue Y, et al. Absolute zero: Reinforced self-play reasoning with zero data[J]. arXiv preprint arXiv:2505.03335, 2025.

2.4 RL for Agent Self-Improving



Park C, Han S, Guo X, et al. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning[J]. arXiv preprint arXiv:2502.18439, 2025.

2.5 RL for Reasoning & Perception

Fast Reasoning: Intuitive and Efficient Inference Fast reasoning models operate in a manner analogous to System 1 [2] cognition: quick, intuitive, and pattern-driven. They generate immediate responses without explicit step-by-step deliberation, excelling in tasks that prioritize fluency, efficiency, and low latency. Most conventional LLMs fall under this category, where reasoning is implicitly encoded in next-token prediction [13, 156]. However, this efficiency comes at the cost of systematic reasoning, making these models more vulnerable to factual errors, biases, and shallow generalization.

To address the severe hallucination problems in fast reasoning, current research has largely focused on various direct approaches. Some studies attempt to mitigate errors and hallucinations in the next-token prediction paradigm by leveraging internal mechanisms [157, 158, 159] or by simulating human-like cognitive reasoning. Other works propose introducing both external and internal confidence estimation methods [160, 161] to identify more reliable reasoning paths. However, constructing such external reasoning frameworks often risks algorithmic adaptivity issues and can easily fall into the complexity trap.

Slow Reasoning: Deliberate and Structured Problem Solving In contrast, slow reasoning models are designed to emulate System 2 cognition [2] by explicitly producing intermediate reasoning traces. Techniques such as chain-of-thought prompting, multi-step verification [162], and reasoning-augmented reinforcement learning allow these models to engage in deeper reflection and achieve greater logical consistency. While slower in inference due to extended reasoning trajectories, they achieve higher accuracy and robustness in knowledge-intensive tasks such as mathematics, scientific reasoning, and multi-hop question answering [163]. Representative examples include OpenAI's o1 [30] and o3 series [32], DeepSeek-R1 [31], as well as methods that incorporate dynamic test-time scaling [164, 165, 166, 167] or reinforcement learning [168, 169, 170, 171, 172, 173] for reasoning.

Perception:

- Image
  DeepEyes, Groud-R1, BRPO, VisTA, Vtool-R1
- Video
  Video-R1, TinyLLaVa-Video-R1
- Audio
  EchoInk-R1, exc

2.6 RL for Multi-turn Training

ByteDance | Seed

AgentGym-RL:
Training LLM Agents for Long-Horizon Decision Making through Multi-Turn Reinf

Zhiheng Xi[1*†], Jixuan Huang
Baodai Huang[1], Honglin Guo[1], Jiaqi
Jiazheng Zhang[1], Wenxiang Chen[1], We
Zehui Chen[2], Zhengyin Du[2], Xuesong
Tao Gui[1,3†], Zuxuan Wu[1,3], Qi Zhang[1†], Xu

[1]Fudan University, [2]ByteDance Seed, [3]

Group-in-Group Policy Optimization for LLM Agent Training

ong Liu[1] Bo An[1,2]
ity, Singapore
ore

WebResearcher: Unleashing unbounded reasoning capability in Long-Horizon Agents

Zile Qiao*(✉), Guoxin Chen*, Xuanzhong Chen*, Donglei Yu*, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, Rui Min, Minpeng Liao, Yong Jiang(✉), Pengjun Xie, Fei Huang, Jingren Zhou

Tongyi Lab , Alibaba Group

ByteDance | Seed

UI-TARS-2 Technical Report: Advancing GUI Agent with Multi-Turn Reinforcement Learning

ByteDance Seed

See Contributions section for a full author list.

DeepDive: Advancing Deep Search Agents with Knowledge Graphs and Multi-Turn RL

Rui Lu[1*†], Zhenyu Hou[1*], Zihan Wang[2†*], Hanchen Zhang[1], Xiao Liu[1], Yujiang Li[1], Shi Feng[2], Jie Tang[1], Yuxiao Dong[1]

[1] Tsinghua University, [2] Northeastern University
lu-r21@mails.tsinghua.edu.cn, yuxiaod@tsinghua.edu.cn

3.0 RL for Vertical Domains

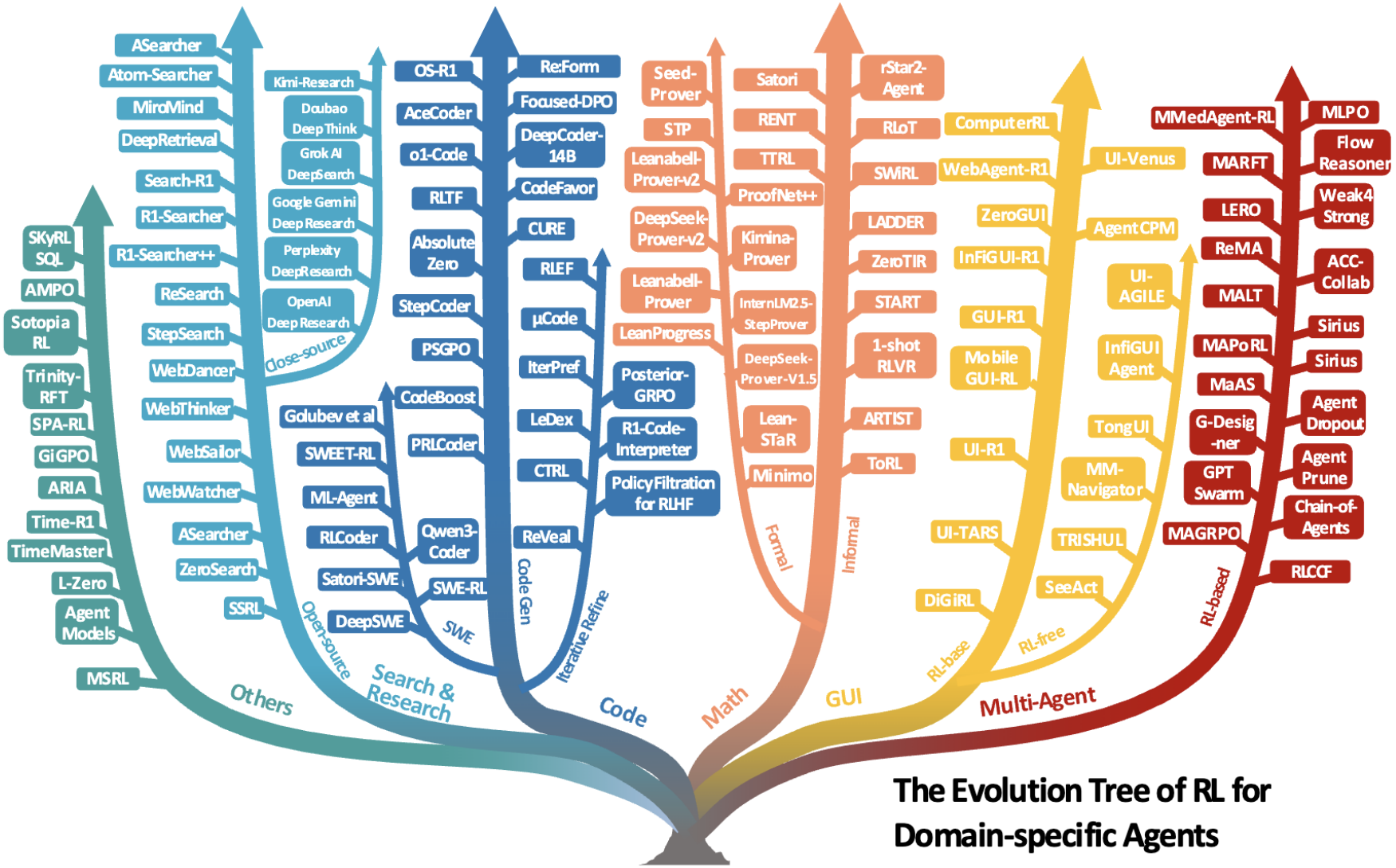


Figure 6: The evolution tree of RL for domain-specific agents.

Thanks for listening

Guibin Zhang @ NUS SoC

2025.9.18