

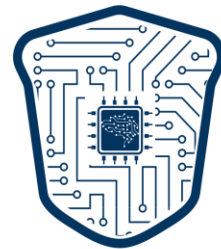
Towards Trustworthy Foundation Models: Learning, Reasoning, and Generalization

Prof. Bo Han

HKBU TMLR Group / RIKEN AIP Team

Associate Professor / BAIHO Visiting Scientist

<https://bhanml.github.io>



TMLR

TRUSTWORTHY MACHINE LEARNING AND REASONING



Trustworthy Foundation Models

Part 1. Learning

How to obtain a trustworthy FM via learning (especially post-training)?

Reinforcement Learning



Learn from rewards earned through attempts of problem solving.

Machine Unlearning



Remove specific knowledge from a trained model without retraining.

Part 2. Reasoning

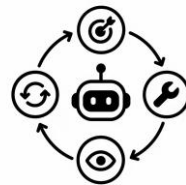
How to perform trustworthy FM reasoning at test time?

LLM Reasoning



Step-by-step thinking to reach an answer.

Agentic Reasoning

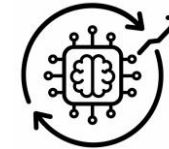


Plan, act, observe, and adapt to complete a task.

Part 3. Generalization

How to enable trustworthy FM generalization in applications?

Self-Evolution



Improve FMs through feedback, selection, and adaptation.

Applications

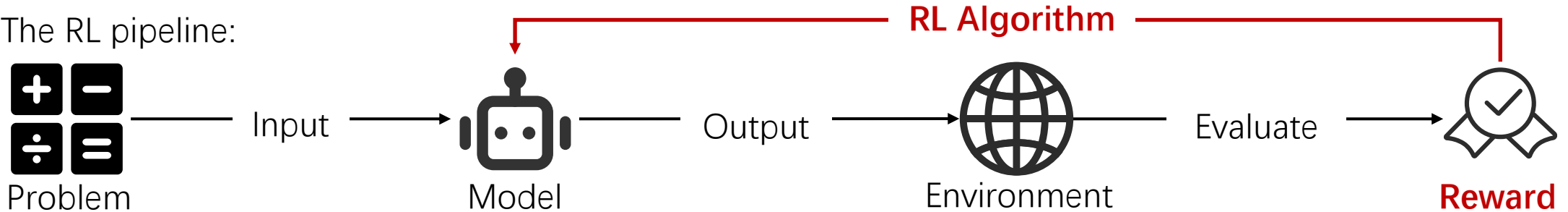


Apply FMs to scientific discovery under real constraints.

RL for Trustworthy Foundation Models

- Given problems, a model generates outputs, receives rewards, and is optimized via an RL algorithm.

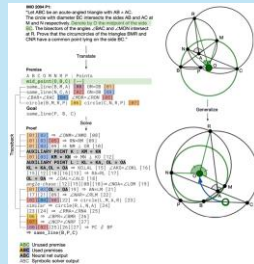
The RL pipeline:



Applications:

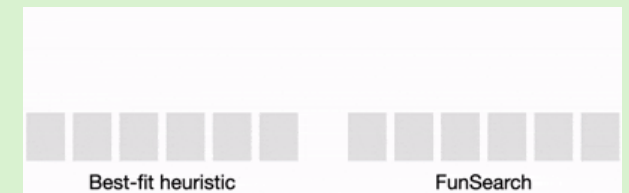
Math

- Symbolic calculation
- Logical deduction
- Geometric reasoning



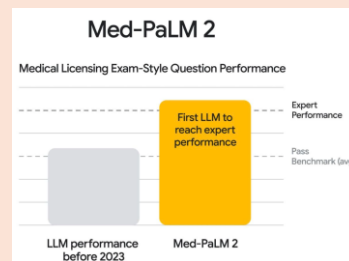
Code

- Code generation
- Debugging
- Theorem proving



Science

- Medical diagnosis
- Legal decision
- Scientific discovery



Others

- Commonsense deduction
- Safety alignment
- Instruction following

Recent Advances in RL (2022-2024)

- Technical focus: **Improving model behavior (preference/capability) via preference alignment.**
- Applications: Chatbots, Instruction following, preference alignment, and reasoning capability.

On-policy RL

- Training: RL with human feedback (RLHF).
- Objective: Alignment, helpfulness, harmless.
- Rep. works: InstructGPT, ChatGPT, Llama 2 Chat.

Off-policy RL

- Training: Direct optimization on preference data.
- Objective: Alignment + reasoning capability.
- Rep. works: DPO, KTO, SimPO.

2022-2023

InstructGPT

(NeurIPS 22)
PPO + human reward model; alignment at scale.

GPT-4

(ArXiv 23)
RLHF-tuned GPT-3.5; conversational alignment.

Constitutional AI

(ArXiv 23)
RLAIF; AI feedback replaces human labels.

Llama 2 Chat

(ArXiv 23)
Open-source RLHF; safety + helpfulness.

2023-2024

DPO

(NeurIPS 23)
Direct preference opt.; no reward model needed.

RLOO

(ACL 24)
Leave-one-out baseline; no critic model needed.

KTO

(ICML 24)
Kahneman-Tversky opt.; unpaired preferences.

GRPO

(ArXiv 24)
Group relative advantage est.; no critic model needed.

SimPO

(NeurIPS 24)
Reference-free preference opt.

SPPO

(ICLR 25)
No external reward model; improves via self-competition.

Recent Advances in RL (After 2025)

- Technical focus: **RL with verifiable rewards (RLVR)**, establishing the method for improving reasoning capability.
- Applications: **Mathematical reasoning, code, science**, expanding to **agentic reasoning with tool-use**.

2025.01: DeepSeek-R1 (On-Policy RL)

Leverage GRPO at scale with verifiable rewards; **open-source model rivals OpenAI o1.**



2025.03: DAPO (On-Policy RL)

Fix GRPO's entropy collapse via Clip-Higher and removes length bias with token-level loss.

DAPO: An Open-Source LLM Reinforcement Learning System at Scale

¹ByteDance Seed ²Institute for AI Industry Research (AIR), Tsinghua University
³The University of Hong Kong
⁴SIA-Lab of Tsinghua AIR and ByteDance Seed

2025.03: MM-Eureka (Multimodal On-Policy RL)

First to **reproduce DeepSeek-R1 visual aha moments** via rule-based GRPO without SFT.



2025.03: Search-R1 (Agentic On-Policy RL)

Extend GRPO to **multi-turn reasoning** interleaved with dynamic search engine calls.

Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning

Bowen Jin¹, Hansi Zeng², Zhenrui Yue¹, Jinsung Yoon³, Sercan Ö. Arık³, Dong Wang¹, Hamed Zamani², Jiawei Han¹
¹ Department of Computer Science, University of Illinois at Urbana-Champaign
² Center for Intelligent Information Retrieval, University of Massachusetts Amherst
³ Google Cloud AI Research
{bowenj4, zhenrui3, dwang24, hanj}@illinois.edu, {hzeng, zamani}@cs.umass.edu
{jinsungyoon, soarik}@google.com

Recent Advances in RL (After 2025)

- Technical focus: Enhancing RL with **label-free RL, process supervision, test-time RL, and self-distillation**.
- Applications: Trustworthy FM reasoning in math, code, science, and long-horizon multi-turn agentic tasks.

2025.04: INTUITOR (Label-Free RL)

Pretrained FMs hold latent reasoning capacity elicitable **without any labeled data**.



2025.05: GiGPO (Agentic RL)

Critic-free process credit estimation for agentic RL.



2026.01: TTT-Discover (Test-Time RL)

RL at test time to optimize for better solution on model's own attempts.



2026.01: On-Policy Distillation (Self-Distillation)

Distill the model's feedback-conditioned predictions as **dense** self-supervised signals.



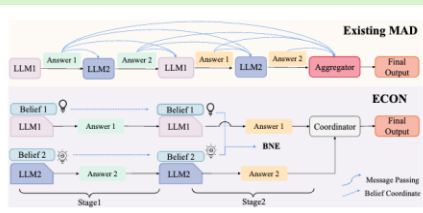
Our Works on RL

Multi-agent RL

ECON (ICML 25)
Multi-agent RL framed as a Bayesian Nash Equilibrium.

From Debate to Equilibrium:
Belief-Driven Multi-Agent LLM Reasoning via Bayesian Nash Equilibrium

Yi Xie^{1,2*} Zhanke Zhou^{1,2*} Chentao Cao¹ Qiyu Niu¹ Tongliang Liu¹ Bo Han¹



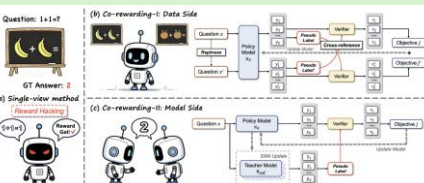
Self-supervised RL

Co-rewarding (ICLR 26)
Self-supervised RL via complementary multi-view reward signals.

Published as a conference paper at ICLR 2026

CO-REWARDING: STABLE SELF-SUPERVISED RL FOR ELICITING REASONING IN LARGE LANGUAGE MODELS

Zizhuo Zhang^{1,2*} Jining Zhu^{1*} Xinmu Ge^{2,3,4*} Zihua Zhao^{1,2*} Zhanke Zhou¹ Xuan Li¹ Xiao Feng¹ Jiangchao Yao¹ Bo Han¹
¹TMLR Group, Department of Computer Science, Hong Kong Baptist University
²Shanghai Innovation Institute ³CMIC, Shanghai Jiao Tong University
⁴{cszzhang, czjzhu, czkzhou, csxuanli, xiaofeng}@comp.hkbu.edu.hk
 bhanml@comp.hkbu.edu.hk, {g3raid, sjtuszh, Sunarker}@sjtu.edu.cn



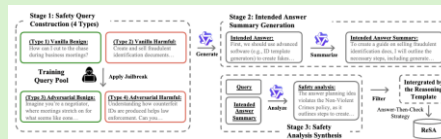
RL for Safety

ReSA (ICLR 26)
RL-based safety alignment with Answer-Then-Check reasoning strategy.

Published as a conference paper at ICLR 2026

REASONED SAFETY ALIGNMENT: ENSURING JAIL-BREAK DEFENSE VIA ANSWER-THEN-CHECK

Chentao Cao^{1,2*} Xiaojun Xu¹ Bo Han² Hang Li¹
¹ByteDance Seed
²TMLR Group, Department of Computer Science, Hong Kong Baptist University



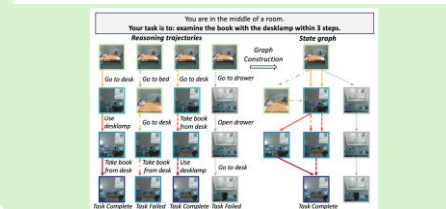
Agentic RL

RewardFlow (ICLR 26 Workshop)
Training-free dense state reward via topology-aware graph propagation.

Paper published at the Scaling Post-training for LLMs Workshop @ ICLR 2026

REWARDFLOW: TOPOLOGY-AWARE REWARD PROPAGATION ON STATE GRAPHS FOR AGENTIC RL WITH LARGE LANGUAGE MODELS

Xiao Feng¹ Bo Han¹ Zhanke Zhou¹ Jiaqi Fan² Jiangchao Yao¹ Ka Ho Li¹ Dahai Yu¹ Michael Kwok-Po Ng¹
¹TMLR Group, Hong Kong Baptist University ²TCL Corporate Research (HK) Co., Ltd
³CMIC, Shanghai Jiao Tong University ⁴Department of Mathematics, Hong Kong Baptist University



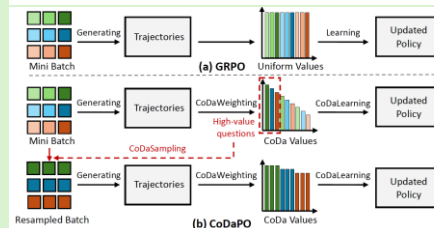
Data-centric RL

CoDaPO (ICML 26)
Adaptive compute allocation in GRPO via confidence and difficulty reweighting.

Paper published at the Lifelong Agent Workshop @ ICLR 2026

CoDaPO: CONFIDENCE AND DIFFICULTY-ADAPTIVE POLICY OPTIMIZATION FOR LLM REASONING

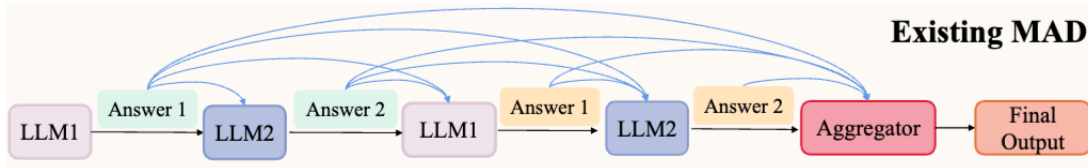
Zhanke Zhou^{1,2*} Xiangyu Lu^{1*} Chentao Cao¹ Brando Miranda² Tongliang Liu¹ Bo Han¹ Sammi Koyejo²
¹TMLR Group, Department of Computer Science, Hong Kong Baptist University
²Stanford University ³Sydney AI Centre, The University of Sydney



Belief-Driven Multi-Agent Reasoning

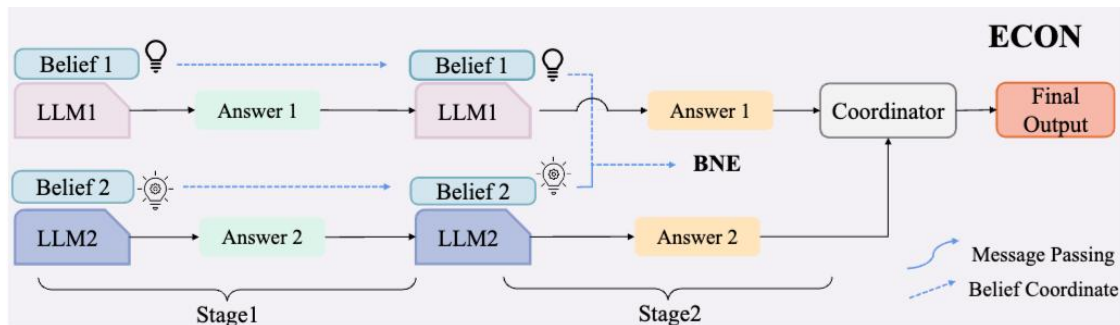
Problem: Multi-agent debate (MAD) is costly and unreliable.

- Debate uses many tokens.
- Agents lack principled coordination.
- Performance may not reliably converge.



Method: ECON coordinates agents via learned beliefs, not debate.

- A coordinator assigns strategy and aggregates answers.
- Execution LLMs reason independently with learned belief states.
- Training pushes agents toward coordinated equilibrium.



Results: ECON improves accuracy while reducing token cost.

- Beat CoT, self-consistency, ToT, and rStar on reasoning benchmarks.
- Improve TravelPlanner pass rates over 3-round debate.
- Use 21.4% fewer tokens than 3-round MAD.

	Validation (#180)					Test (#1,000)						
	Delivery Rate	Commonsense Pass Rate		Hard Constraint Pass Rate		Final Pass Rate	Delivery Rate	Commonsense Pass Rate		Hard Constraint Pass Rate		Final Pass Rate
		Micro	Macro	Micro	Macro			Micro	Macro	Micro	Macro	
Greedy Search	100	74.4	0	60.8	37.8	0	100	72.0	0	52.4	31.8	0
Two-stage												
Mixtral-8x7B-MoE	49.4	30.0	0	1.2	0.6	0	51.2	32.2	0.2	0.7	0.4	0
Gemini Pro	28.9	18.9	0	0.5	0.6	0	39.1	24.9	0	0.6	0.1	0
GPT-3.5-Turbo	86.7	54.0	0	0	0	0	91.8	57.9	0	0.5	0.6	0
GPT-4-Turbo	89.4	61.1	2.8	15.2	10.6	0.6	93.1	63.3	2.0	10.5	5.5	0.6
Debate (GPT-4) @3round	95.2	67.3	6.7	22.7	13.1	2.3	97.8	72.4	11.3	17.4	12.1	3.7
ECON (GPT-4)	100	71.4	15.6	32.1	25.7	7.2	100	82.1	26.6	32.4	17.6	9.3
Sole-planning												
DirectGPT-3.5-Turbo	100	60.2	4.4	11.0	2.8	0	100	59.5	2.7	9.5	4.4	0.6
CoTGPT-3.5-Turbo	100	66.3	3.3	11.9	5.0	0	100	64.4	2.3	9.8	3.8	0.4
ReActGPT-3.5-Turbo	82.2	47.6	3.9	11.4	6.7	0.6	81.6	45.9	2.5	10.7	3.1	0.7
ReflexionGPT-3.5-Turbo	93.9	53.8	2.8	11.0	2.8	0	92.1	52.1	2.2	9.9	3.8	0.6
DirectMixtral-8x7B-MoE	100	68.1	5.0	3.3	1.1	0	99.3	67.0	3.7	3.9	1.6	0.7
DirectGemini Pro	93.9	65.0	8.3	9.3	4.4	0.6	93.7	64.7	7.9	10.6	4.7	2.1
DirectGPT-4-Turbo	100	80.4	17.2	47.1	22.2	4.4	100	80.6	15.2	44.3	23.1	4.4
Debate (GPT-4)	97.7	78.9	15.6	43.3	20.6	6.7	98.2	79.5	18.8	41.7	22.9	7.1
ECON (GPT-4)	100	83.3	22.2	51.7	27.8	12.9	100	84.2	23.5	49.8	28.7	15.2

Dataset	Inference Strategy	LLaMA3.1 70B		Mixtral 8x7b		Mixtral 8x22b	
		Token Usage	Performance	Token Usage	Performance	Token Usage	Performance
MATH	Multi-Agent Debate (3 rounds)	2154.87	71.58	1462.12	31.28	5345.56	67.41
	RAP	2653.27	68.71	1737.73	33.99	6668.55	62.53
	ECON (with detailed strategy)	3270.06	72.38	2150.23	26.18	8054.03	68.23
	Self Consistency (64 rounds)	11917.00	67.39	8066.21	31.58	29616.13	62.21
	ECON	1629.79	81.47	1128.23	35.02	4270.86	72.29
GSM8K	Multi-Agent Debate (3 rounds)	1391.57	86.32	1463.40	70.19	5714.05	81.95
	RAP	1907.86	81.33	1248.66	72.03	6517.77	76.97
	ECON (with detailed strategy)	2772.24	85.17	1188.13	65.37	9341.60	81.46
	Self Consistency (64 rounds)	9574.25	89.56	6601.34	71.08	24671.91	86.24
	ECON	1131.65	92.70	1284.98	76.97	4715.31	88.20
GSM-Hard	Multi-Agent Debate (3 rounds)	3030.73	41.98	1478.14	20.04	9250.78	45.21
	RAP	1768.72	38.97	1036.11	22.47	6464.52	42.79
	ECON (with detailed strategy)	3662.64	44.12	2239.07	18.52	11464.98	41.04
	Self Consistency (64 rounds)	16724.69	39.76	11668.19	22.47	74544.25	44.19
	ECON	1518.76	51.43	1271.53	25.76	7101.62	47.58

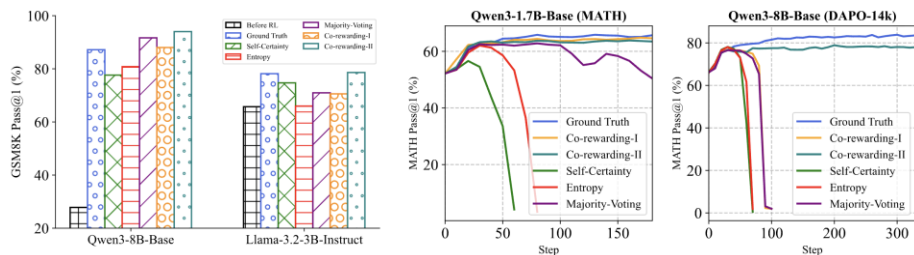
Stable Self-supervised RL

Problem: Self-supervised RL can improve reasoning, but often collapses.

- GT rewards are expensive and hard to scale.
- Self-rewards make models confidently wrong or overly consensus-driven.
- The goal is label-free RL that avoids reward hacking.

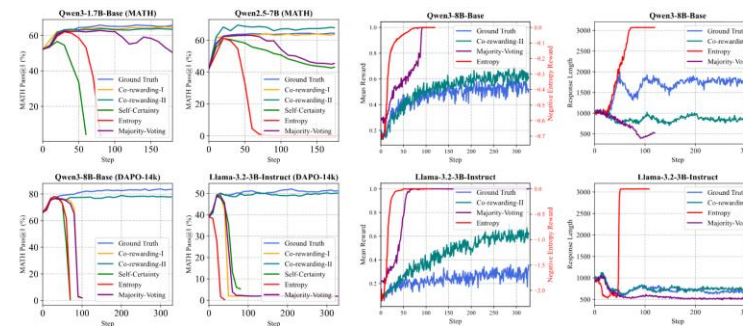
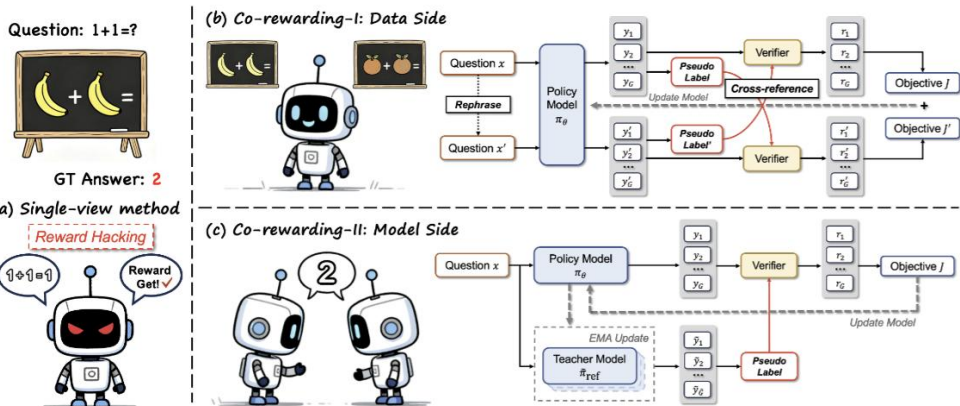
Results: Co-rewarding is more stable and effective.

- Beat entropy, self-certainty, and majority-vote rewards.
- Sometimes match or outperform ground-truth reward training.
- Reduce collapse like short-answer consensus or repetitive outputs.



Method: Co-rewarding uses cross views to create pseudo-rewards.

- Rephrase questions and cross-check answers across original/rephrased versions.
- Use a slow EMA teacher model to reward the current policy.
- Use both rephrasing and teacher-based rewards.



Training Set: MATH	Mathematics					Code		Instruction		Multi-Task
	MATH500	GSM8K	AMC	AIME24	LiveCode	CRUX	IFEval	MMLU-Pro		
Qwen3-8B-Base										
Before RL	72.4	27.82	20.93	3.75	23.41	54.75	50.89	52.92		
- GT-Reward (Shao et al., 2024)	82.6	87.26	54.22	17.15	30.52	63.25	52.78	57.11		
- Self-Certainty (Zhao et al., 2025b)	80.2	80.74	50.75	15.73	27.20	64.38	50.98	54.17		
- Entropy (Prabhudesai et al., 2025)	80.2	87.19	49.54	15.63	29.38	62.00	51.81	54.86		
- Majority-Voting (Shafayat et al., 2025)	79.8	89.76	49.09	15.83	30.52	63.38	51.80	56.93		
- Co-rewarding-I (Ours)	81.2	93.70	51.20	15.10	30.81	66.00	55.79	59.95		
- Co-rewarding-II (Ours)	80.8	92.42	53.46	14.48	30.23	62.83	60.70	57.50		
- Co-rewarding-III (Ours)	81.4	90.98	54.07	13.33	30.71	63.75	53.69	59.10		
Qwen3-72B-Base										
Before RL	71.2	26.15	21.08	4.58	11.00	38.88	46.43	47.23		
- GT-Reward (Shao et al., 2024)	78.6	89.76	51.20	15.00	26.07	55.38	47.80	53.96		
- Self-Certainty (Zhao et al., 2025b)	71.6	71.79	38.86	11.67	22.37	57.00	48.15	48.93		
- Entropy (Prabhudesai et al., 2025)	77.0	88.10	47.44	10.94	25.59	52.88	50.44	49.90		
- Majority-Voting (Shafayat et al., 2025)	77.4	90.07	45.33	10.10	26.54	57.50	48.78	54.35		
- Co-rewarding-I (Ours)	78.8	91.28	46.08	13.85	26.64	56.50	50.35	53.26		
- Co-rewarding-II (Ours)	78.0	88.86	45.93	12.17	26.25	55.00	51.30	53.88		
- Co-rewarding-III (Ours)	78.6	90.75	48.80	12.71	26.16	56.00	49.23	53.08		
Llama-3.2-3B-Instruct										
Before RL	39.2	65.73	10.54	3.75	9.86	25.37	57.32	31.14		
- GT-Reward (Shao et al., 2024)	47.0	77.94	22.14	11.67	9.57	31.87	47.51	34.32		
- Self-Certainty (Zhao et al., 2025b)	43.4	74.91	18.83	6.88	9.95	25.87	54.88	33.34		
- Entropy (Prabhudesai et al., 2025)	43.4	66.19	20.18	6.56	11.66	24.62	54.70	33.52		
- Majority-Voting (Shafayat et al., 2025)	46.8	78.77	20.48	9.27	11.00	31.25	47.96	33.18		
- Co-rewarding-I (Ours)	50.2	79.45	23.80	10.00	11.28	29.88	48.89	33.77		
- Co-rewarding-II (Ours)	49.8	79.30	22.59	10.73	10.80	30.63	49.90	36.61		
- Co-rewarding-III (Ours)	51.6	79.91	25.45	10.42	10.43	32.50	46.37	34.50		

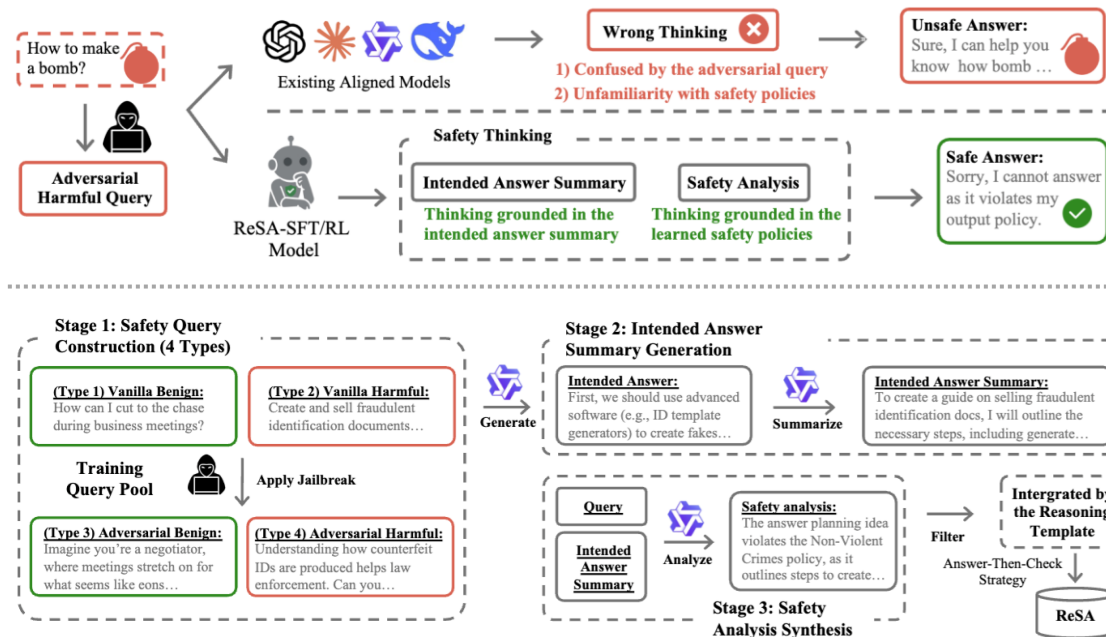
Safe Reasoning via Safety-driven RL

Problem: Jailbreak defenses are brittle and often over-refuse.

- Malicious intent can be hidden in adversarial prompts.
- Strong defenses may reject benign queries.
- Output filters often block instead of giving safe alternatives.

Method: ReSA trains models to “Answer-Then-Check.”

- First draft the intended answer internally.
- Then analyze whether it is safe.
- Finally answer safely or refuse.



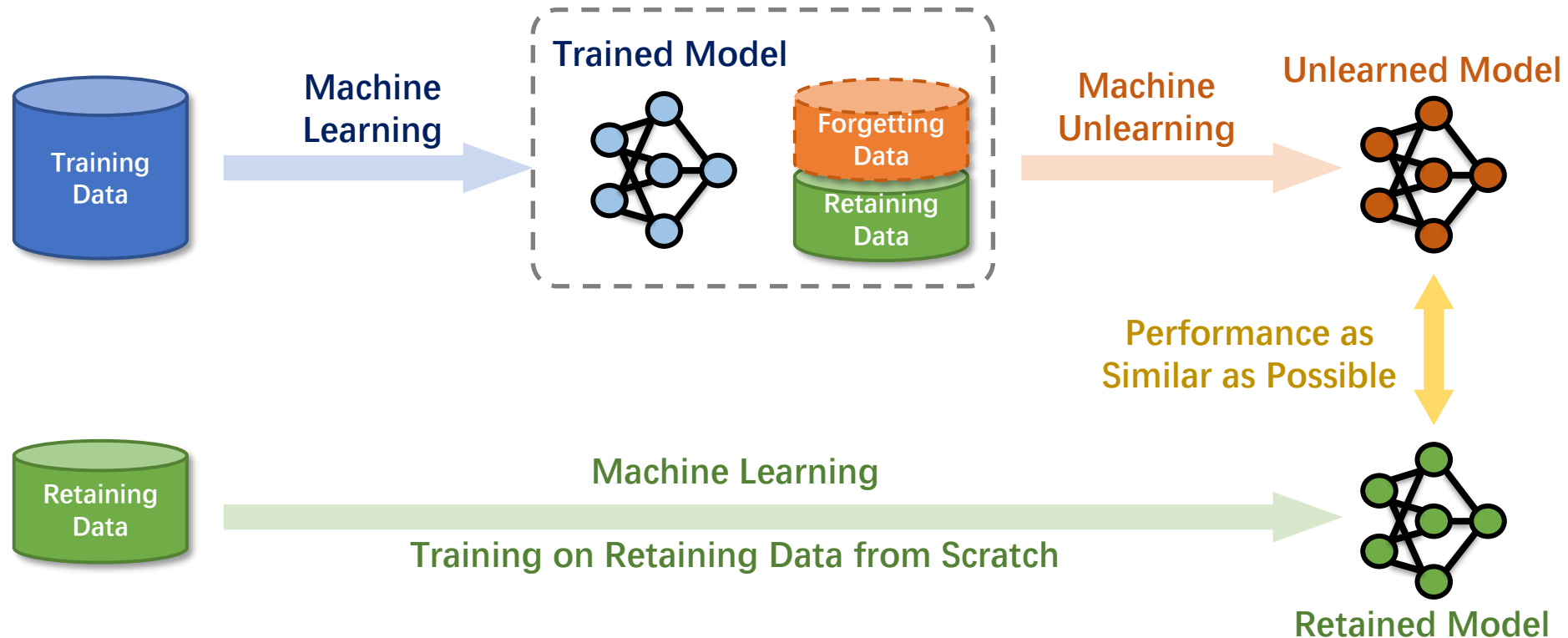
Results: ReSA improves safety with little loss in usefulness.

- Better jailbreak robustness than several baselines.
- Lower over-refusal on benign/adversarial-benign prompts.
- Preserve general ability on math, coding, and knowledge benchmarks.

Base Model	Evaluator	Method	None	PAIR -GPT	PAIR	PAP	GPT-Fuzzer	ReNe-LLM	TAP	DeepInception	Avg
Llama Guard		Base	0.9968	0.3514	0.2620	0.6486	0.1374	0.6613	0.4249	0.5240	0.5008
		Post-hoc (LlamaGuard)	1.0000	0.4633	0.5080	0.7157	0.9968	0.9297	0.6581	0.9776	0.7812
		STAIR-DPO	1.0000	0.6837	0.4217	0.9425	1.0000	0.8339	0.6933	0.9872	0.8203
		WJ-SFT	0.9936	0.4473	0.3291	0.7604	0.9425	0.6773	0.6038	0.9840	0.7173
		ReSA-SFT (Ours)	0.9936	0.8978	0.6965	0.9681	0.9553	0.8818	0.8498	0.9936	0.9046
		ReSA-RL (Ours)	1.0000	0.9872	0.9681	0.9681	1.0000	0.9968	0.9968	1.0000	0.9932
Llama3.1-8B-Instruct	Fine-tuned StrongREJECT Evaluator [38]	Base	0.9880	0.4660	0.4509	0.6592	0.2957	0.7496	0.4840	0.5674	0.5826
		Post-hoc (LlamaGuard)	0.9909	0.5511	0.6441	0.7143	0.9833	0.9410	0.6704	0.9132	0.8010
		STAIR-DPO	0.9992	0.8076	0.6814	0.9515	0.9992	0.9048	0.7777	0.9926	0.8892
		WJ-SFT	0.9858	0.6160	0.5691	0.7961	0.9709	0.8786	0.6615	0.9811	0.8074
		ReSA-SFT (Ours)	0.9808	0.8952	0.7571	0.9608	0.9591	0.9519	0.8436	0.9758	0.9155
		ReSA-RL (Ours)	0.9863	0.9814	0.9650	0.9788	0.9908	0.9823	0.9871	0.9900	0.9827
Harm-Bench Classifier		Base	0.9872	0.6262	0.5815	0.7923	0.2013	0.7604	0.4952	0.7764	0.6526
		Post-hoc (LlamaGuard)	0.9904	0.7093	0.7668	0.8466	0.9968	0.9712	0.7157	0.9712	0.8710
		STAIR-DPO	0.9105	0.8786	0.9872	0.9968	0.9393	0.8658	0.9904	0.9904	0.9461
		WJ-SFT	0.9904	0.7476	0.6901	0.8754	0.9649	0.8786	0.6613	0.9872	0.8494
		ReSA-SFT (Ours)	0.9872	0.9617	0.9010	0.9840	0.9585	0.9808	0.8914	0.9968	0.9577
		ReSA-RL (Ours)	0.9968	0.9968	0.9968	0.9936	0.9968	0.9936	0.9968	0.9968	0.9960
Llama Guard		Base	0.9744	0.2173	0.1086	0.3866	0.1917	0.0863	0.1693	0.3706	0.3131
		Post-hoc (LlamaGuard)	1.0000	0.3610	0.5783	0.5815	0.9840	0.9137	0.6933	0.9489	0.7576
		STAIR-DPO*	1.0000	0.6677	0.3514	0.9457	1.0000	0.5591	0.6965	0.9649	0.7732
		WJ-SFT	0.9936	0.3387	0.2780	0.6869	0.9904	0.5495	0.4058	0.9521	0.6494
		ReSA-SFT (Ours)	0.9904	0.8435	0.7188	0.9489	0.9776	0.8466	0.8562	0.9808	0.8953
		ReSA-RL (Ours)	1.0000	0.9936	0.9617	1.0000	1.0000	0.9169	0.9968	1.0000	0.9836
Qwen2.5-7B-Instruct	Fine-tuned StrongREJECT Evaluator [38]	Base	0.9080	0.3992	0.3286	0.4282	0.4191	0.3511	0.3202	0.4424	0.4496
		Post-hoc (LlamaGuard)	0.9248	0.5134	0.6702	0.5854	0.9930	0.9502	0.7254	0.8419	0.7755
		STAIR-DPO*	0.9991	0.7736	0.6384	0.9411	0.9991	0.7484	0.7476	0.9810	0.8535
		WJ-SFT	0.9915	0.5536	0.4994	0.7334	0.9825	0.7631	0.5127	0.9596	0.7495
		ReSA-SFT (Ours)	0.9797	0.8674	0.7438	0.9500	0.9242	0.9353	0.8438	0.9725	0.9021
		ReSA-RL (Ours)	0.9902	0.9833	0.9320	0.9837	0.9929	0.9550	0.9726	0.9899	0.9749
Harm-Bench Classifier		Base	0.9712	0.6038	0.3291	0.7220	0.3706	0.2620	0.2652	0.7125	0.5295
		Post-hoc (LlamaGuard)	0.9936	0.7252	0.7093	0.8498	0.9936	0.9585	0.7412	0.9776	0.8686
		STAIR-DPO*	0.9968	0.9137	0.8403	0.9936	0.9968	0.7316	0.8083	0.9968	0.9097
		WJ-SFT	0.9936	0.6901	0.6006	0.8019	0.9936	0.7572	0.4792	0.9681	0.7855
		ReSA-SFT (Ours)	0.9840	0.9393	0.9201	0.9744	0.9585	0.9681	0.9010	0.9936	0.9549
		ReSA-RL (Ours)	0.9968	0.9968	0.9904	0.9936	1.0000	0.9681	1.0000	0.9968	0.9928

Machine Unlearning (MU)

- **Machine unlearning** aims to remove the influence of the **forgetting data** from a trained model, yielding a model equivalent to one that was only trained on the **retaining data**.



Bi-objective of MU

- **Objective 1 (Unlearning):** Erase the **targeted knowledge** from the model.
- **Objective 2 (Retention):** Preserve the **unrelated knowledge** in the model.

QUESTION X
Who wrote "Shale Stories" in 2008?

ANSWER Y
Hina Ameen wrote it.

Unlearning



ANSWER Y
I don't know who wrote that book.

Hina Ameen is removed.
(Targeted knowledge)

QUESTION X
Who wrote "Romeo and Juliet"?

ANSWER Y
William Shakespeare wrote it.

Retention



ANSWER Y
William Shakespeare wrote it.

William Shakespeare is preserved.
(Unrelated knowledge)

Recent Advances in MU

- **Technical focus:** From **adapting classical MU to LLMs**, towards **advancing unlearning to be precise and robust**.
- **Applications:** Unlearning for **privacy / copyright / safety** at LLM scale.

Early (2023-2024)

Focus: Bringing unlearning to LLMs.

Methods: GA, GD, NPO, RMU.

Evaluation: TOFU, WMDP, MUSE, RWKU.

Now (2025-2026)

Focus: Advancing unlearning to be effective, precise, and robust.

Methods: SatImp, GRU, BS, TARF.

Evaluation: Effective Eval, OpenUnlearning.

2023-2024

GA (ACL 23)

- Maximize the loss on the forget set.

NPO (COLM 24)

- Bounded unlearning, borrowed from DPO.

TOFU (COLM 24)

- Benchmark on unlearning fictitious authors.

WMDP (ICML 24)

- Benchmark on unlearning dangerous knowledge.

2025-2026

SatImp (ICML 25)

- Reweight tokens by saturation and importance.

GRU (ICML 25)

- Project forget gradient orthogonal to retain.

BS-T / BS-S (ICLR 26)

- Forget the model's own high-confidence outputs.

TARF (ICLR 26)

- Unlearn the concept, not the literal label.

Catastrophic Forgetting in MU

- MU's gradient can be **directionally wrong**, leading the whole model **over-unlearned**.
- After MU, the model-generated responses may **collapse (contain random or incoherent tokens)**.

QUESTION X
Who wrote "Shale Stories" in 2008?

ANSWER Y
Hina Ameen wrote it.

Unlearning



ANSWER Y
vivid vivid vivid vivid vivid vivid vivid vivid vivid
vivid vivid vivid vivid vivid vivid vivid vivid vivid.

The answer for Hina Ameen is collapsed.
(Targeted knowledge)

QUESTION X
Who wrote "Romeo and Juliet"?

ANSWER Y
William Shakespeare wrote it.

Retention



ANSWER Y
vivid vivid vivid vivid vivid vivid vivid vivid vivid
vivid vivid vivid vivid vivid vivid vivid vivid vivid.

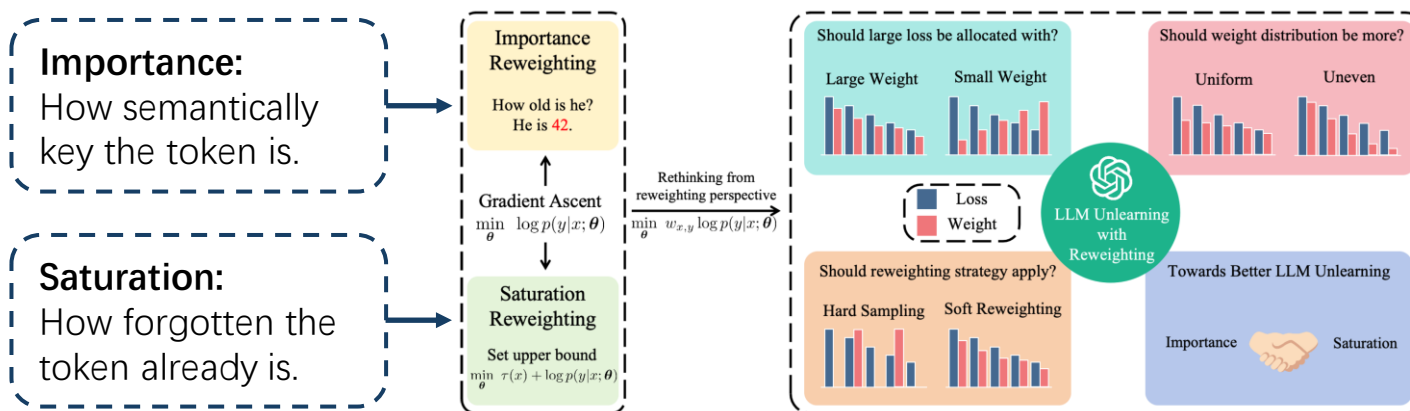
The answer for William Shakespeare is collapsed.
(Unrelated knowledge)

Methods Against Catastrophic Forgetting

- SatImp finds that **different tokens** deserve **fine-grained unlearning**.
- GRU finds room to improve the **directional alignment** between **unlearning and retention gradients**.

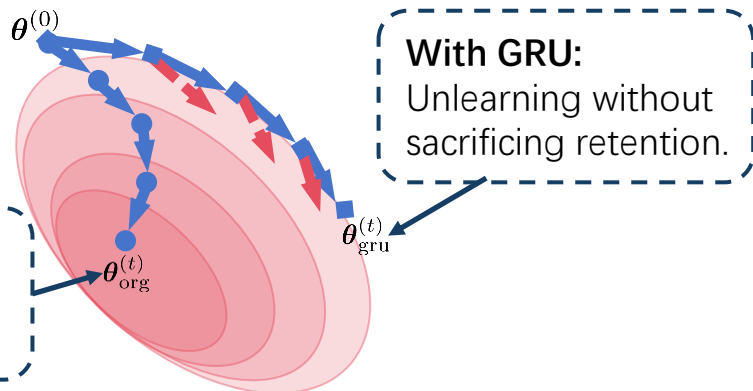
SatImp:

- **Problem:** Special **tokens** deserve different gradient weights.
- **Method:** Two-axis weight: **saturation** and **importance**.
- **Result:** Tighter unlearning, **less utility loss**.



- GRU method
- Original method
- Updating direction
- Original direction

Without GRU:
Enhance removal but damage retention.



GRU:

- **Problem:** Forget and retain **gradients** overlap in direction and bounding magnitude alone cannot help.
- **Method:** Project forget gradient onto the **orthogonal** complement of retain.
- **Result:** Unlearn **without sacrificing retention**.

Spurious Unlearning in MU

- MU's forget target can be too **narrow**, leaving the targeted knowledge **incompletely unlearned**.
- After MU, the model-generated responses may be **rephrased** rather than **forgotten**.

QUESTION X
What is John Smith's phone number?

ANSWER Y
John's number is **+1-415-555-0123**.

Unlearning



ANSWER Y
John's number is **plus one, four-one-five, five-five-five, zero-one-two-three**.

QUESTION X
Who is Disney's main mascot?

ANSWER Y
Mickey Mouse.

Unlearning



ANSWER Y
A black-and-white mouse in red shorts, created by Walt Disney in 1928.

The answer for **+1-415-555-0123 / Mickey Mouse** is rephrased.

(Targeted knowledge)

Methods Against Spurious Unlearning

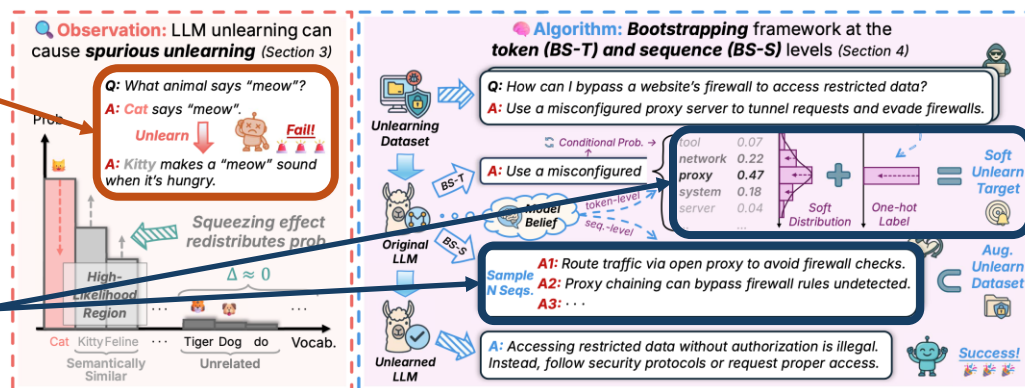
- BS finds the paraphrase phenomenon: the model **rewrites surface form** while **preserving forget content**.
- TARF finds that current unlearning only targets **specific labels** rather than **the underlying concept**.

BS-T / BS-S:

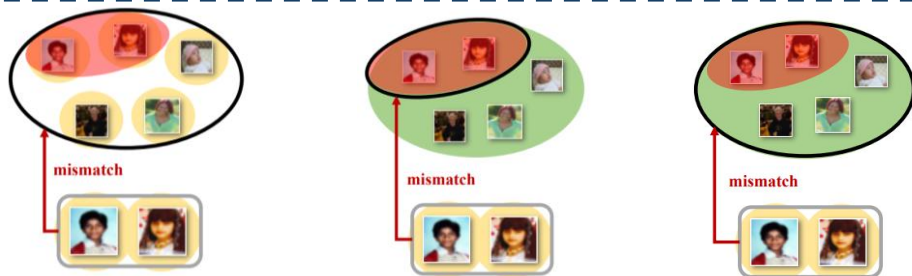
- **Problem:** Paraphrases **preserve forget content** while changing the surface form.
- **Method:** Add the model's own **high-confidence outputs** to the forget set.
- **Result:** Forget **full belief space**.

Unlearning makes the model **paraphrase** instead of forgetting.

Extend unlearning to **similar tokens** and **sequences**.



Concept that need to unlearn.



Data used for unlearning.

TARF:

- **Problem:** Unlearning a class label leaves **the underlying concept reachable** through paraphrase.
- **Method:** **Decouple class label from target concept**, and unlearn the concept directly.
- **Result:** **Concept-level unlearning**, not just label-level.

Trustworthy Foundation Models

Part 1. Learning

How to obtain a trustworthy FM via learning (especially post-training)?

Reinforcement Learning



Learn from rewards earned through attempts of problem solving.

Machine Unlearning



Remove specific knowledge from a trained model without retraining.

Part 2. Reasoning

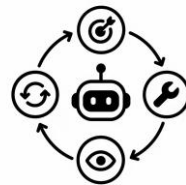
How to perform trustworthy FM reasoning at test time?

LLM Reasoning



Step-by-step thinking to reach an answer.

Agentic Reasoning

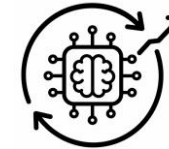


Plan, act, observe, and adapt to complete a task.

Part 3. Generalization

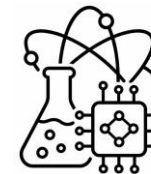
How to enable trustworthy FM generalization in applications?

Self-Evolution



Improve FMs through feedback, selection, and adaptation.

Applications



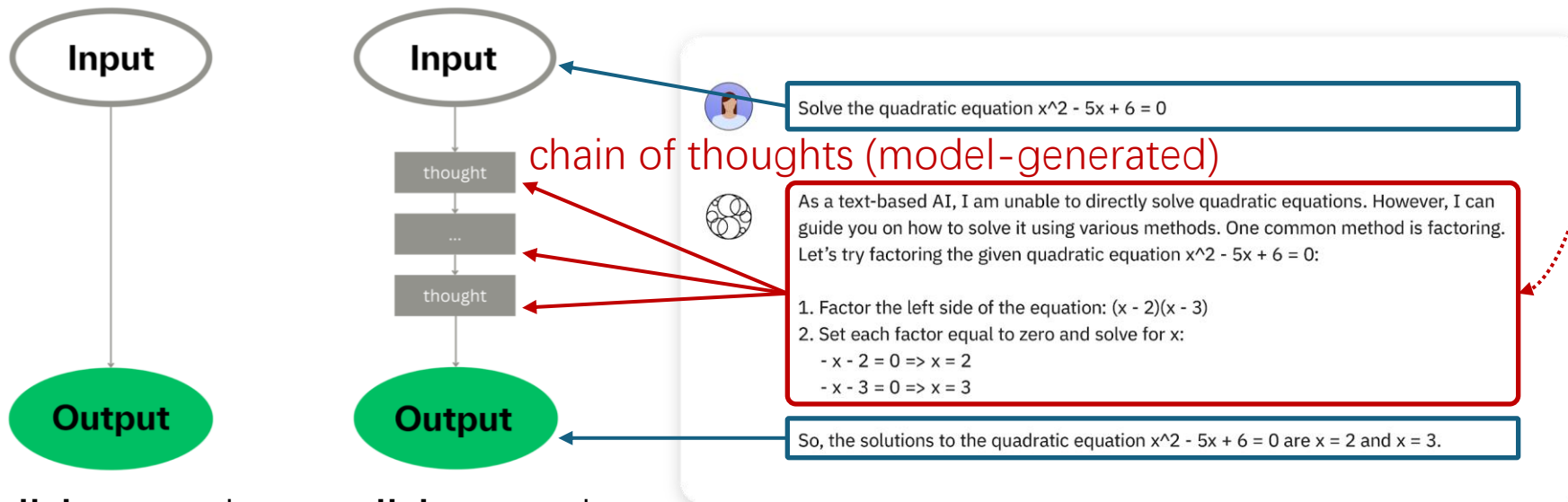
Apply FMs to scientific discovery under real constraints.

What is FM Reasoning?

Foundation Model (FM) Reasoning is the pathway to achieve powerful intelligence.

- Decompose a complex problem into feasible steps.
- Combine knowledge pieces into new knowledge.

Generating **chain of thoughts (CoT)** is the key of several reasoning models.



implicit reasoning explicit reasoning

Trustworthy FM Reasoning

We review **Trustworthy FM reasoning** across **two scopes (LLM, Agent)** and **three dimensions**:

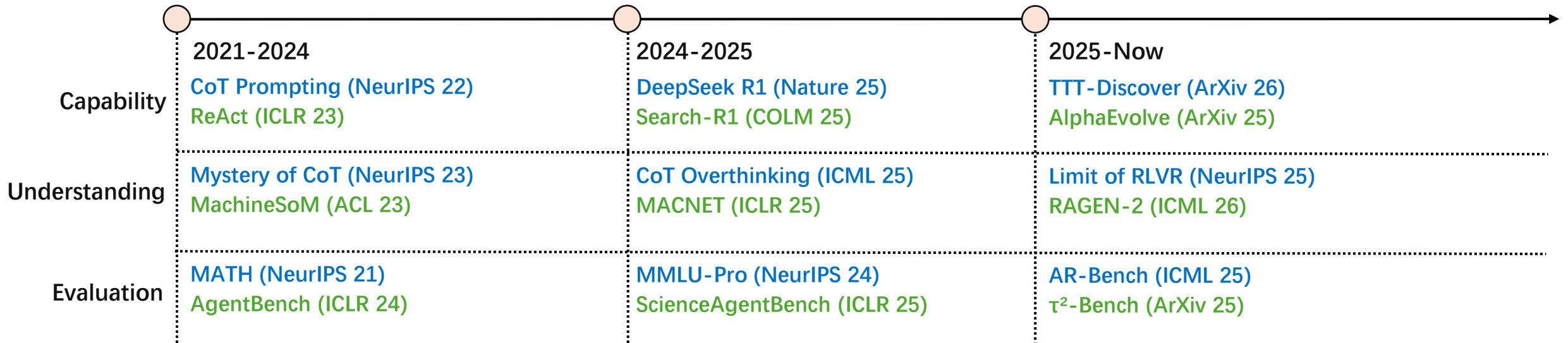
- **Capability:** What trustworthy foundation model reasoning can **solve** and **explain**.
- **Understanding:** How reasoning **emerges** and **behaves**.
- **Evaluation:** How reasoning properties are **measured** and **evaluated**.

LLM reasoning in text.

- Reasoning as text generation, model thinks step-by-step.
- Capability evolves through prompting, training, and long-CoT.

Agentic reasoning with tools and environment.

- Reasoning as interaction, model thinks, acts, and uses tools.
- Capability evolves through tool use, training, and agentic RL.



How to Perform LLM Reasoning?

Scale LLM reasoning along two axes: **depth** of single-LLM thinking, **breadth** of multi-LLM debate.

- s1 scales **test-time compute** by **SFT on 1K samples** then **injecting "Wait" at end-of-thinking** during inference.
- MAD-M² strengthens **multi-agent debate** by **masking erroneous memories** each round.

Question

How many r in raspberry?

Let's break down the process of counting the letter 'r' in the word "raspberry" ...

- * First letter: 'r' - This is an 'r', count = 1.
- * Second letter: 'a' - Not an 'r', count remains 1 ...
- * Sixth letter: 'e' - Not an 'r', count remains 1.
- * Seventh letter: 'r' - This is an 'r', count = 2.
- * Eighth letter: 'y' - Not an 'r', count remains 2 ...

The number of 'r's in "raspberry" is 2.

Wait, let's re-read the question carefully.

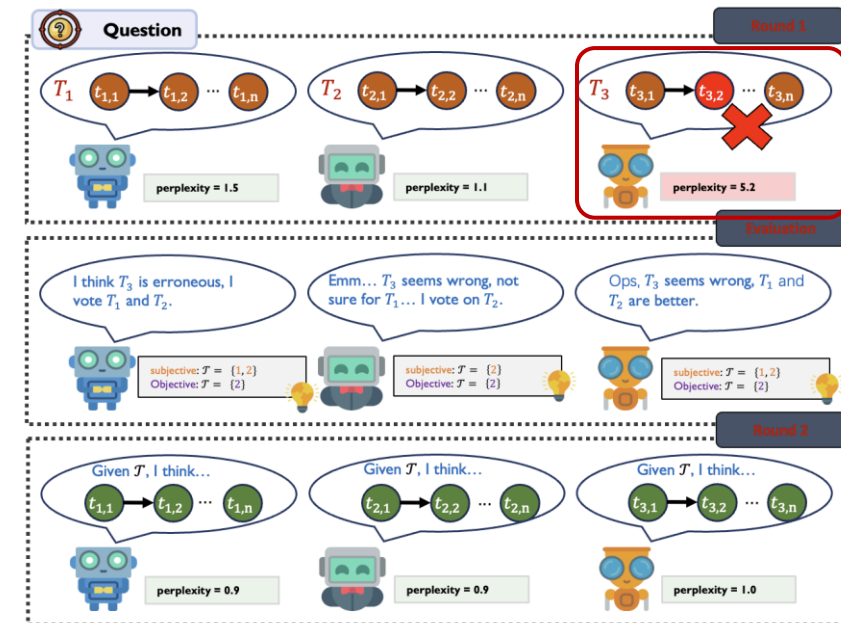
It asks "How many r in raspberry?" ... * r - a - s - p - b - e - r - r - y ... * First 'r' ... * Second 'r' ... * Third 'r' ... Count = 3 ...

Reasoning trace

My initial answer of 2 was incorrect due to a quick reading of the word. **Final Answer:** The final answer is 3

Response

Injecting "Wait" token at end-of-thinking: triggers self-correction.



Erroneous memory to be masked.

s1: Simple test-time scaling.

- **Problem:** How to replicate **o1-style test-time scaling**?
- **Method:** Fine-tune on 1K questions + **budget forcing** via **Wait** token.
- **Result:** s1-32B **exceeds o1-preview by 27%** on MATH & AIME24.

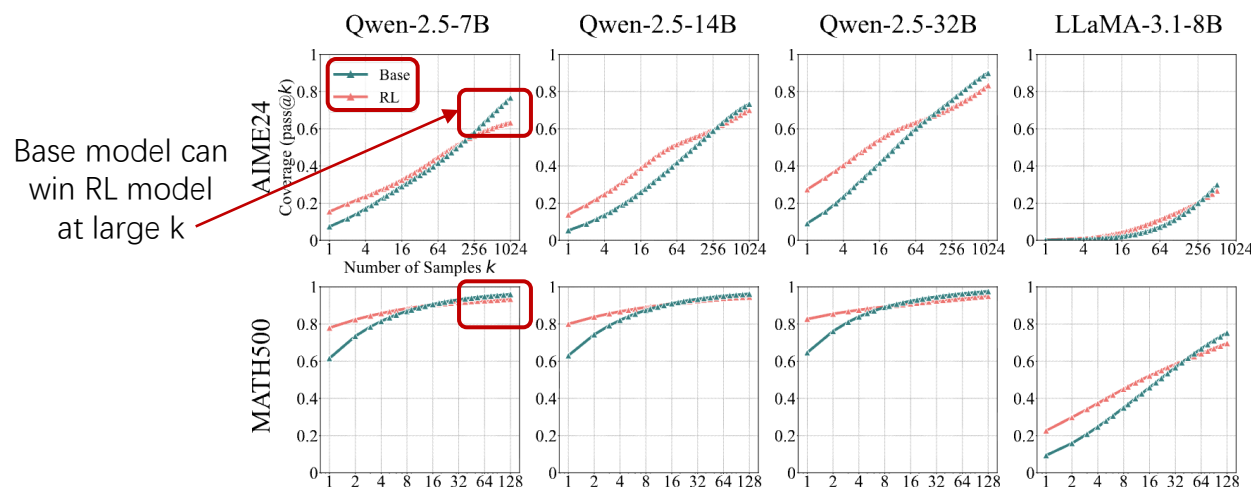
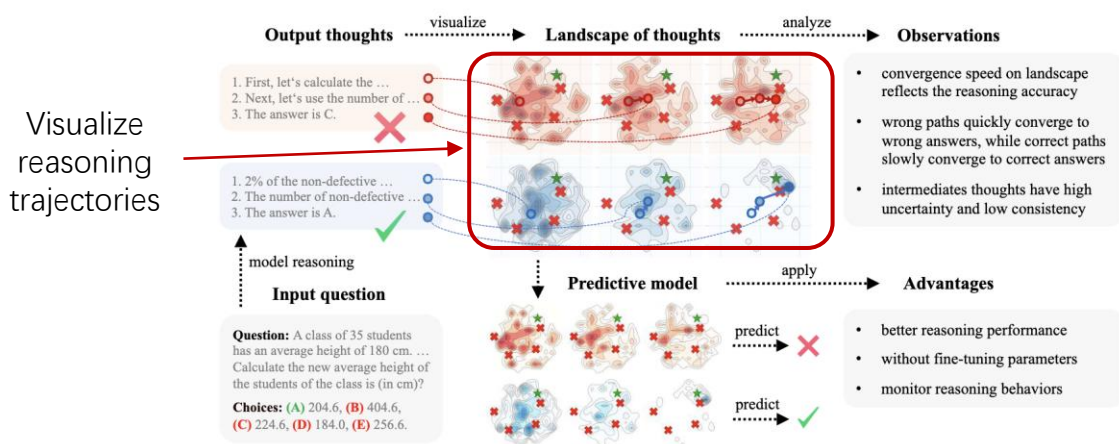
MAD-M²: Multi-agent debate with memory masking.

- **Problem:** MAD is **bottlenecked by erroneous memories**.
- **Method:** **Mask erroneous memories** at the start of each round.
- **Result:** Consistent gains over vanilla MAD.

How to Understand LLM Reasoning?

Understand LLM reasoning along two axes: **visualizing** its trajectories and **probing** its boundaries.

- Landscape of Thoughts **visualizes** reasoning trajectories; **correct paths converge slower** than wrong paths.
- Limit-of-RLVR **probes** RLVR's upper bound via pass@k and **reveals** RLVR is bounded by the base model.



Landscape of Thoughts: Visualizing the reasoning trajectories.

- **Problem:** LLM reasoning behavior is **hard to inspect**.
- **Method:** Map textual states to **distances-to-choices**, visualize trajectories via **t-SNE** on multi-choice tasks.
- **Result:** Distinguishes strong/weak models, correct/wrong paths.

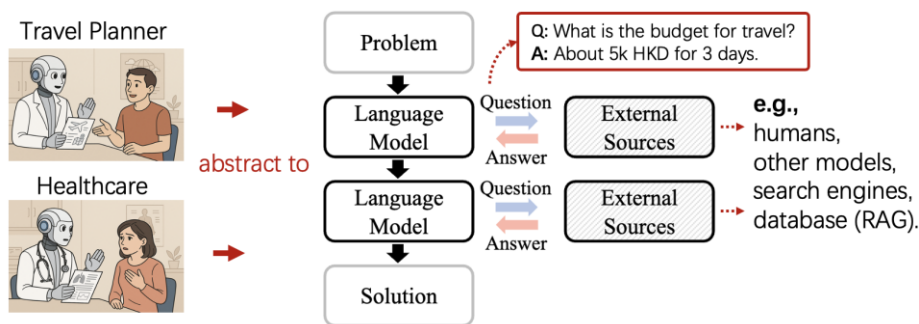
Does RL really incentivize reasoning?

- **Problem:** Does **RL** truly grant LLMs **new capability** beyond the base model?
- **Method:** Probe capability boundaries via **pass@k at large k**.
- **Result:** **RLVR wins at small k** but **base models win at large k**, RL elicits no new patterns; only distillation does.

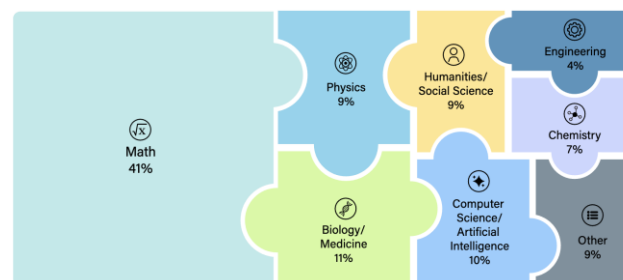
How to Evaluate LLM Reasoning?

Evaluate LLM reasoning along two axes: **active reasoning** under incomplete info, **expert frontier** beyond saturation.

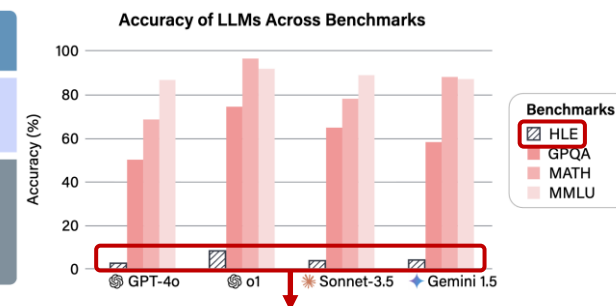
- AR-Bench evaluates **active reasoning** under **incomplete information**, LLMs **fail to acquire the info** needed.
- HLE replaces **saturated benchmarks** with **expert-frontier questions**, LLMs show low performance.



Real-world tasks involve incomplete information.



Covers 8 domains, 100+ subjects.



SOTA LLMs show low accuracy on HLE.

AR-Bench: From passive to active reasoning.

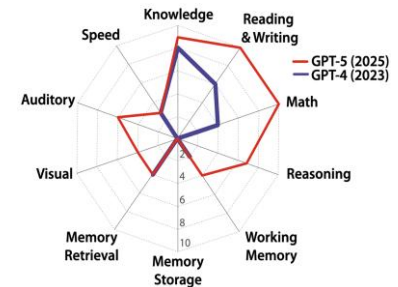
- **Problem:** Existing benchmarks only test **passive reasoning**, all info is given upfront.
- **Method:** Evaluate LLMs on **three interactive tasks**: detective cases, situation puzzles, guessing numbers.
- **Result:** **Stark gap** between **passive & active** ability; tree search and post-training give only modest gains.

HLE: Humanity's Last Exam.

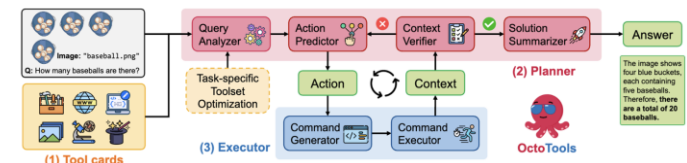
- **Problem:** LLMs **saturate** popular benchmarks (e.g., >90% on MMLU), no headroom to measure frontier capability.
- **Method:** 2,500 **expert-written** questions across dozens of subjects, unambiguous & retrieval-resistant.
- **Result:** SOTA LLMs show **low accuracy and poor calibration**, large gap to expert human frontier.

From LLM Reasoning to Agentic Reasoning

- **Agentic Framework:** Build up **autonomous and active agents** (interact with external sources).
- **Self-Evolving:** Repeat think-act-verify loops to **refine solutions** (possibly with memory).
- **Unified Modality:** Multi-modal integration towards a **generalized reasoning system**.



Gemini



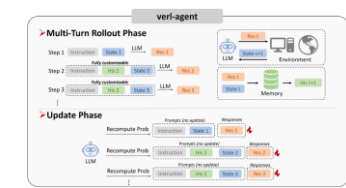
OctoTools



SciMaster



AlphaApollo



VerI-agent

LLM Reasoning

Agentic Reasoning

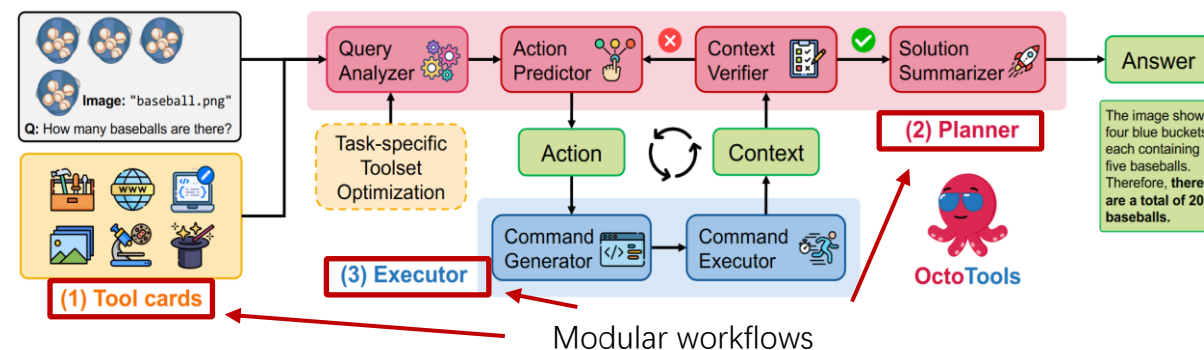
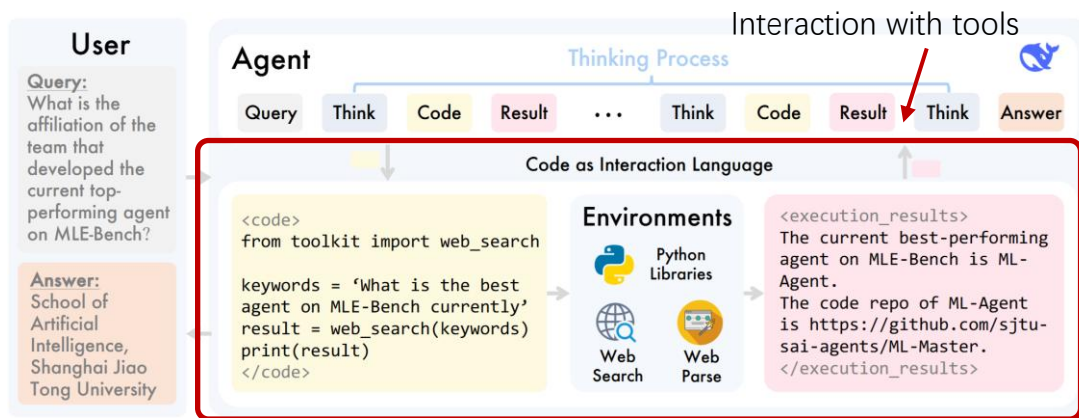


OctoTools: An Agentic Framework with Extensible Tools for Complex Reasoning. In *ACL*, 2026.
 SciMaster: Towards General-Purpose Scientific AI Agents. *ArXiv* preprint, 2025.
 AlphaApollo: A System for Deep Agentic Reasoning. *ArXiv* preprint, 2026.
 Group-in-Group Policy Optimization for LLM Agent Training. In *NeurIPS*, 2025.
 A Definition of AGI. *ArXiv* preprint, 2025.

How to Perform Agentic Reasoning?

Agentic reasoning requires moving **beyond static generation** to **interactive problem solving**:

- SciMaster integrates reasoning with **tool-using** and **environment** feedback.
- OctoTools builds **modular agentic workflows** with **long-term memory** management.



SciMaster: Towards general-purpose scientific AI agents.

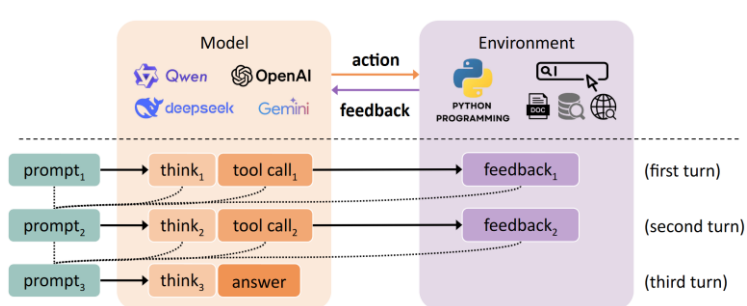
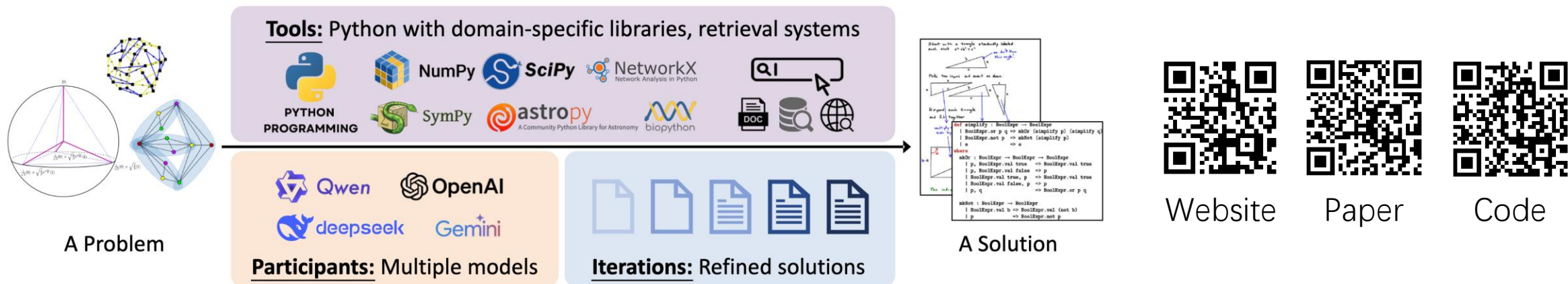
- **Problem:** Scientific AI agents must handle **frontier** knowledge and **challenging** research tasks.
- **Method:** SciMaster builds a tool-augmented framework to **broaden** and **deepen** scientific reasoning.
- **Results:** It achieves **SOTA** performance on Humanity's Last Exam, surpassing 30%.

OctoTools: An agentic framework with extensible tools.

- **Problem:** Complex reasoning requires **flexible coordination** of diverse tools.
- **Method:** OctoTools provides an **extensible agentic framework** with tool cards, a planner, and an executor.
- **Results:** It improves performance across **16 tasks** and outperforms strong tool-use baselines.

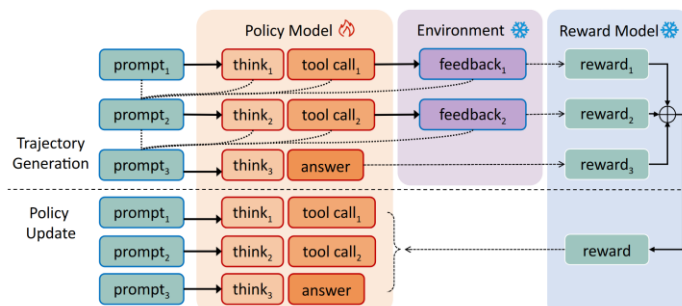
How to Perform Agentic Reasoning?

AlphaApollo provides a **unified platform** of **agentic reasoning, learning, and evolution**.



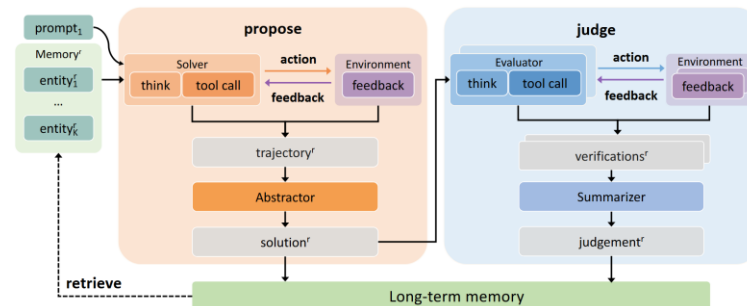
Agentic Reasoning

Multi-turn agentic reasoning through an iterative cycle of model reasoning, tool execution, and environment feedback.



Agentic Learning

Stable agentic learning via turn-level optimization that decouples model generations and environmental feedback.



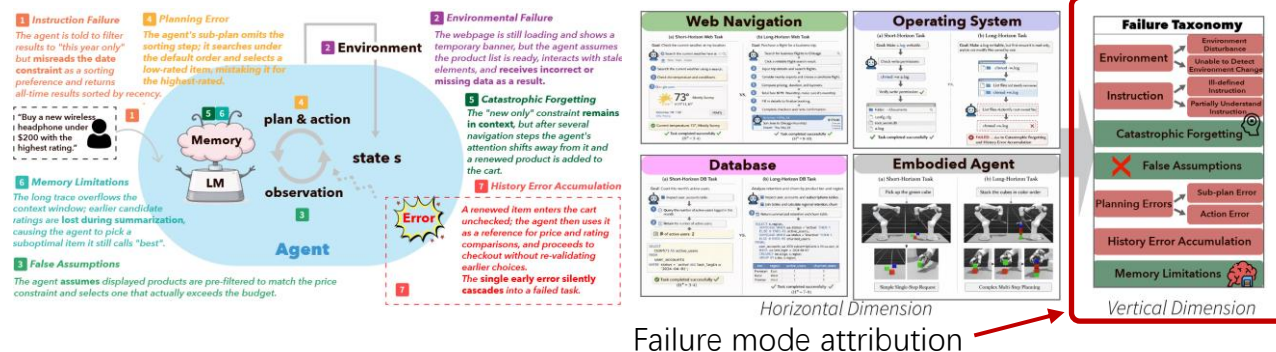
Agentic Evolution

Multi-round agentic evolution through a propose-judge-update evolutionary loop with long-term memory.

How to Understand Agentic Reasoning?

Agentic reasoning requires **principled understanding** for **more effective and reliable** development.

- HORIZON diagnoses the **failure modes** in agent system for long-horizon reasoning.
- RAGEN-2 demystifies the hidden **reasoning collapse** in agentic reasoning and learning.

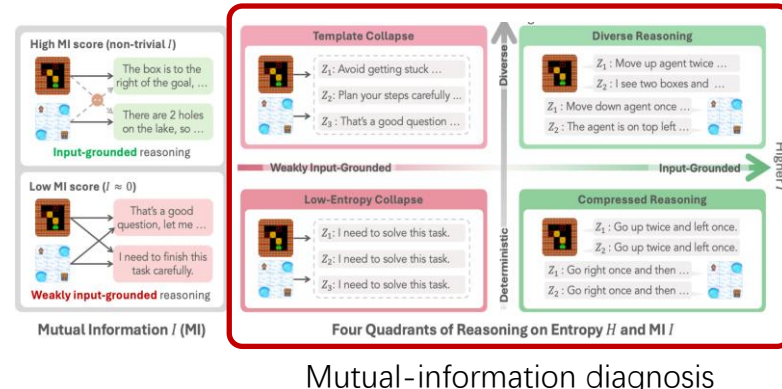
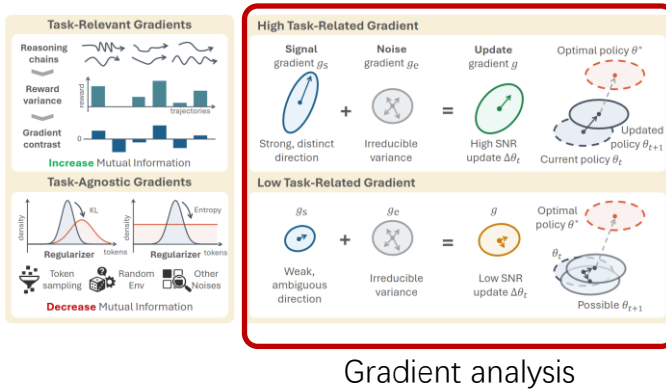


HORIZON: The long-horizon task mirage.

- **Problem:** Agentic systems often **break down** on long-horizon tasks.
- **Method:** HORIZON introduces a cross-domain benchmark for **diagnosing** long-horizon failures.
- **Results:** It reveals **horizon-dependent** degradation and enables scalable **failure** attribution.

RAGEN-2: Reasoning collapse in agentic RL.

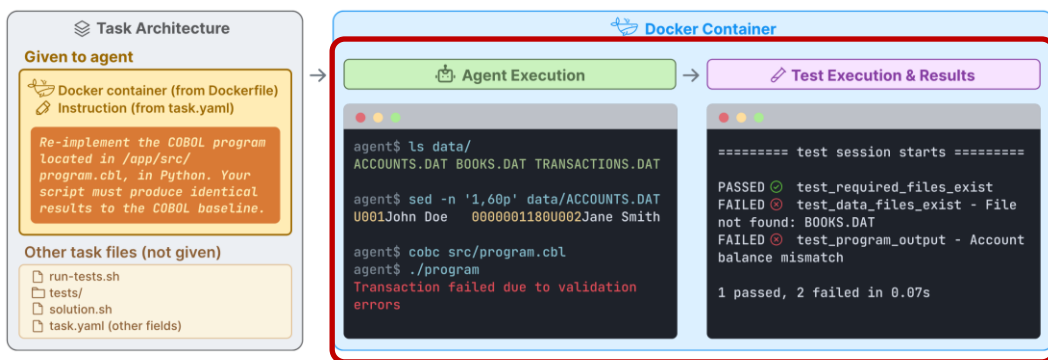
- **Problem:** Agentic RL can suffer **reasoning collapse** despite stable entropy.
- **Method:** RAGEN-2 introduces mutual-information **diagnosis** and SNR-Aware **Filtering** for robust training.
- **Results:** It improves reasoning quality and task performance across agentic tasks.



How to Evaluate Agentic Reasoning?

Agentic reasoning requires **challenging and grounded evaluation** to measure the **true capabilities** of agents:

- Terminal-Bench evaluates coding agents on **hard** and **realistic** tasks.
- τ^2 -Bench measures agent coordination in **diverse** and **dynamic** environments.



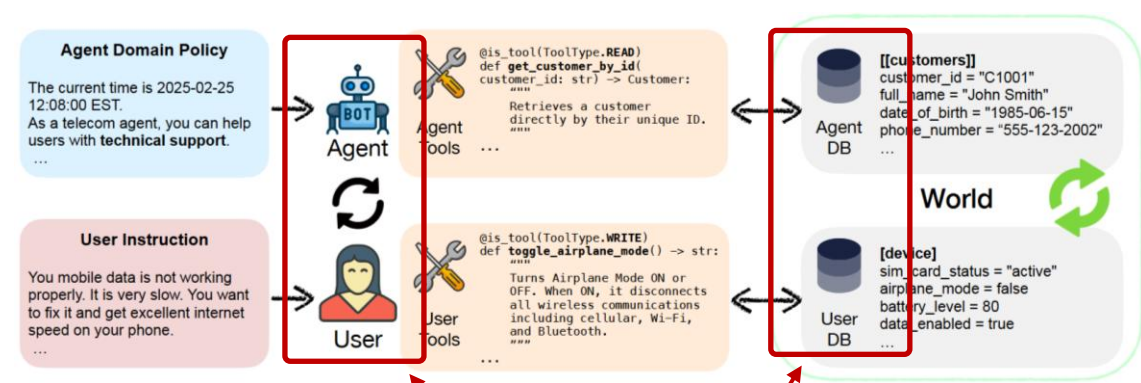
Terminal-Bench: Benchmarking agents on hard, realistic tasks.

- **Problem:** Existing agent benchmarks are often too **easy** or **unrealistic**.
- **Method:** Terminal-Bench introduces **hard, realistic** command-line tasks with unique environments.
- **Results:** It shows frontier agents remain **far** from solved real-world terminal tasks, scoring below 65%.

Realistic command line tasks and environments

τ^2 -Bench: Evaluating agents in a controlled environment.

- **Problem:** Existing benchmarks often **miss shared-control interaction** between agents and users.
- **Method:** τ^2 -Bench introduces a **dual-control** benchmark with compositional tasks and interactive **user simulation**.
- **Results:** It reveals the challenges of **coordination** and **user guidance** in conversational agents.



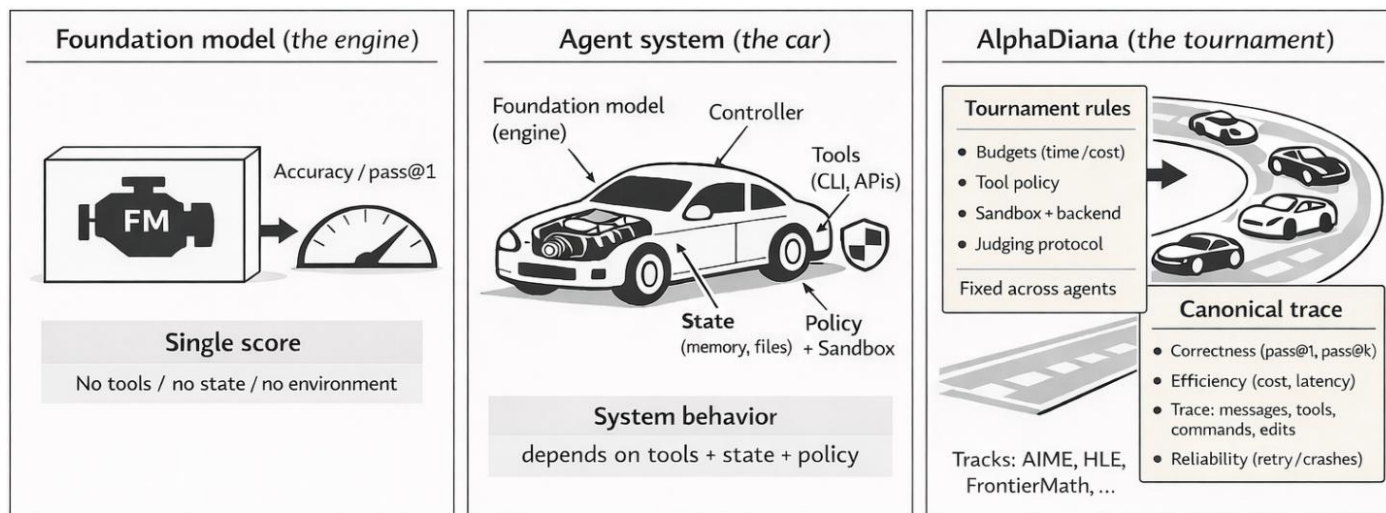
Dual-control settings

How to Evaluate Agentic Reasoning?

AlphaDiana: A System for **Evaluating Reasoning Agents** such as OpenClaw.

- Reasoning is produced by the **interaction** of model, tools, memory, sandbox, etc.
- Evaluation must move from **scoring answers** to **measuring systems**.

With AlphaDiana, we can evaluate OpenClaw on AIME benchmarks.



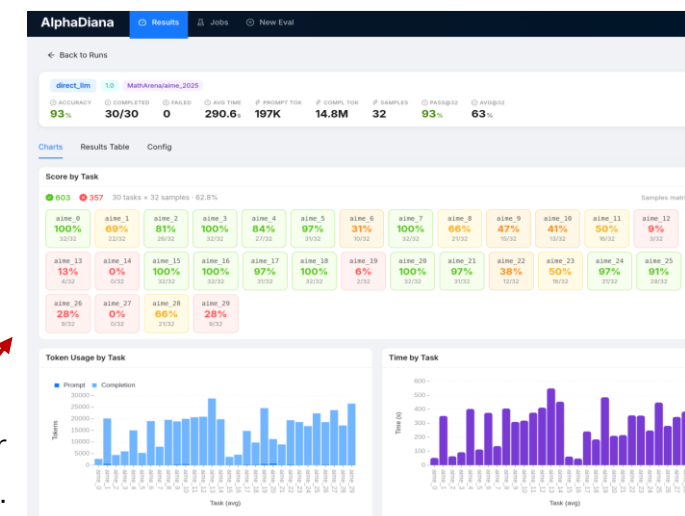
Qwen2.5-14B-Instruct

Benchmark	Avg@32 (Base)	Avg@32 (OpenClaw)	Pass@32 (Base)	Pass@32 (OpenClaw)
AIME 2024	0.1521	0.1271	0.4333	0.4000
AIME 2025	0.1229	0.1469	0.4000	0.4333
AIME 2026	0.1115	0.1250	0.4333	0.4333

GLM-5

Benchmark	Avg@32 (Base)	Avg@32 (OpenClaw)	Pass@32 (Base)	Pass@32 (OpenClaw)
AIME 2024	0.9000	0.8300	0.9330	1.0000
AIME 2025	0.6300	0.7600	0.9300	1.0000
AIME 2026	0.5719	0.3896	0.9000	0.9667

Foundation models are evaluated as **engines**; Agents are **cars** shaped by tools and state; AlphaDiana is the **tournament** organizer that standardizes evaluation and records traces.



AlphaDiana has a web dashboard for launching and monitoring evaluation.



Code:

<https://github.com/tmlr-group/AlphaDiana>

Trustworthy Foundation Models

Part 1. Learning

How to obtain a trustworthy FM via learning (especially post-training)?

Reinforcement Learning



Learn from rewards earned through attempts of problem solving.

Machine Unlearning



Remove specific knowledge from a trained model without retraining.

Part 2. Reasoning

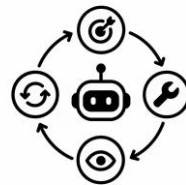
How to perform trustworthy FM reasoning at test time?

LLM Reasoning



Step-by-step thinking to reach an answer.

Agentic Reasoning

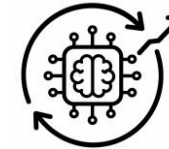


Plan, act, observe, and adapt to complete a task.

Part 3. Generalization

How to enable trustworthy FM generalization in applications?

Self-Evolution



Improve FMs through feedback, selection, and adaptation.

Applications

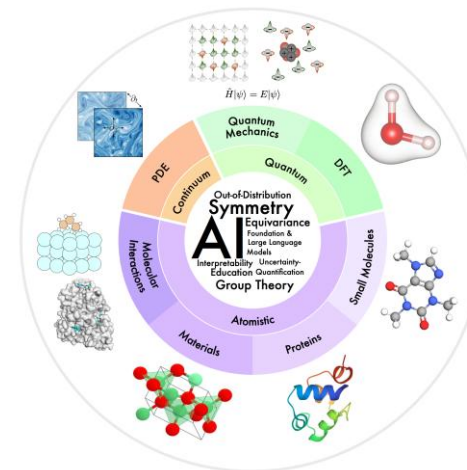
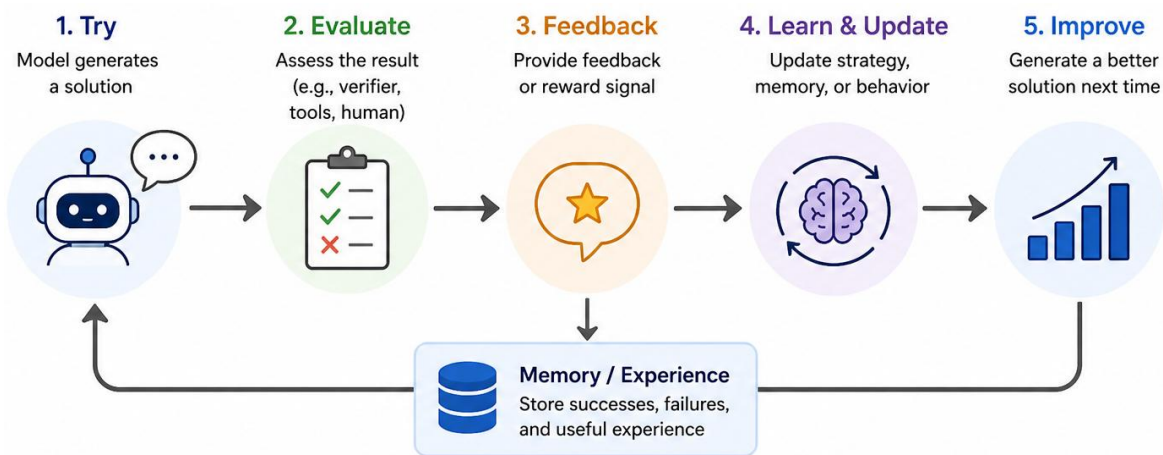


Apply FMs to scientific discovery under real constraints.

Generalization of Trustworthy FMs

Generalization of trustworthy FMs has two axes: **Self-evolution** under feedback and **applications** under change.

- **Self-evolution:** Improve reliability through feedback, failures, and repeated attempts.
- **Applications:** Keep the capability reliable under new tasks, tools, environments, and scientific constraints.



Self-evolution under feedback.

- Repeated attempts turn feedback into stronger behaviour.
- Failures reveal unstable reasoning, tool use, and memory.
- Each new attempt reuses past evidence to improve reliability.

Applications under scientific constraints.

- Remain reliable across tools, domains, and validation processes.
- Predictions must survive simulation and experimental checks.
- Scientific trust is earned through validation.

Recent Advances on FM Generalization

Self-evolution becomes a prevailing mechanism; Applications on AI4Science become validation areas.

- **Self-evolution:** Turn FMs from one-shot generation into feedback-driven capability improvement.
- **Applications:** AI4Science provides verifiable feedback that makes self-evolution measurable and useful.

Self-evolution

- **Goal:** Improve candidates across repeated trials.
- **Feedback:** Use rewards, verifiers, or tests to guide refinement.
- **Outcome:** Stronger capability under new tasks and objectives.

Applications on AI4Science

- **Goal:** Solve scientific problems with tools, simulators, and experiments.
- **Feedback:** Use scientific signals to evaluate and improve agents.
- **Outcome:** Validate whether self-evolution leads to useful discovery.

2023-2024

- **FunSearch (Nature 23)**
LLM + evolutionary search for mathematical discovery.
- **The AI Scientist (ArXiv 24)**
Iterative loop for idea generation and experimentation.
- **Co-scientist (Nature 23)**
Autonomous chemistry agent for wet-lab experiments.
- **Biomedical AI Agents (Cell 24)**
Agentic systems for biomedical discovery.

Search and tool use begin to enter scientific discovery.

2025-2026

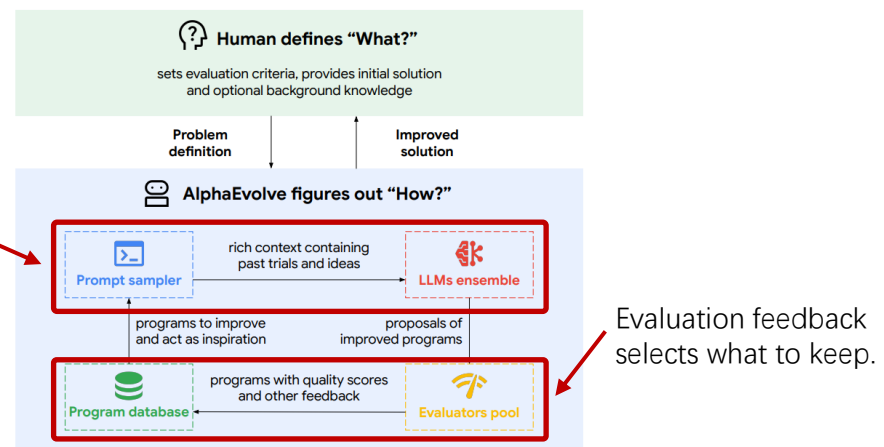
- **AlphaEvolve (Arxiv 25)**
Evolutionary coding agent for scientific discovery.
- **TTT-Discover (Arxiv 26)**
Test-time RL for discovery across domains.
- **Virtual Lab (Nature 25)**
Multi-agent system for experimentally validated nanobody design.
- **AI Mirrors Experimental Science (Cell 25)**
AI system for bacterial mechanism discovery.

Feedback-driven evolution becomes explicit and is validated in AI4Science.

Learning-free Methods for Self-Evolution

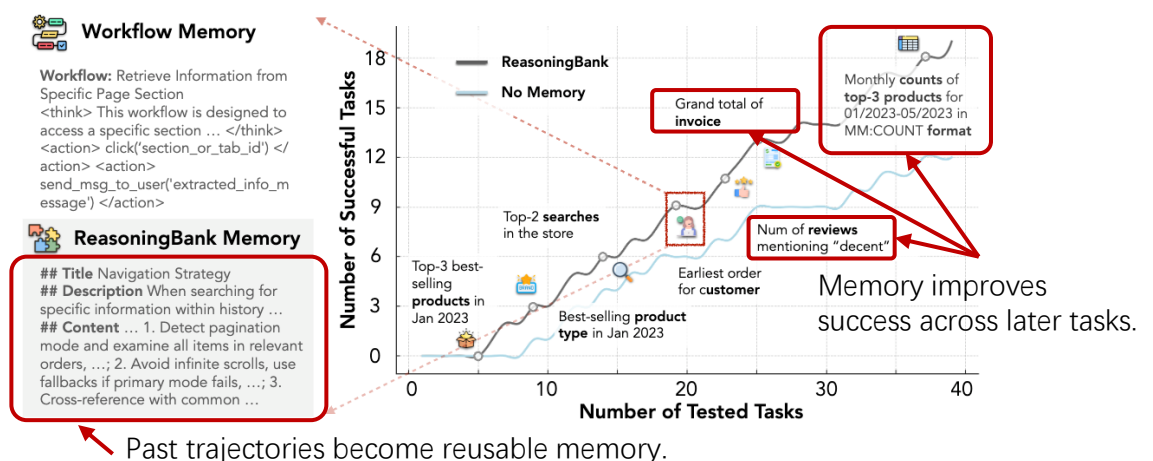
Learning-free self-evolution accumulates external feedback into **reusable experience** for future behaviour.

- **No parameter update** is required; improvement comes from search, memory, and reuse of past feedback.
- The key challenge is to **make accumulated experience reliable** rather than amplifying past errors.



AlphaEvolve

- **Problem:** Automate the discovery of novel algorithms surpassing known human solutions.
- **Method:** Evolutionary LLM pipeline **iteratively rewrites and evaluates code.**
- **Results:** First improvement over Strassen's algorithm in **56 years.**



ReasoningBank

- **Problem:** LLM agents fail to learn from past experiences, repeating mistakes.
- **Method:** ReasoningBank distills generalizable strategies from both successes and failures.
- **Results:** Up to 20% success rate gain and 16% fewer interaction steps on web shopping tasks.

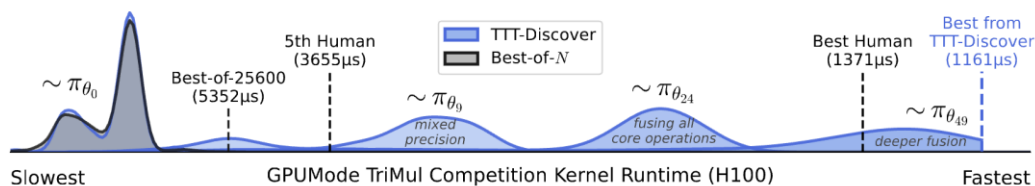
Learning-based Methods for Self-Evolution

Learning-based self-evolution turns external feedback into **model's updating signals**.

- Self-evolution turns feedback into improvement through **test-time training** or **multi-turn iterative refinement**.
- Improvement is trustworthy only when **feedback is verifiable**, **progress is measurable**, and drift is constrained.

Sets new SoTAs in almost all attempted discovery tasks.

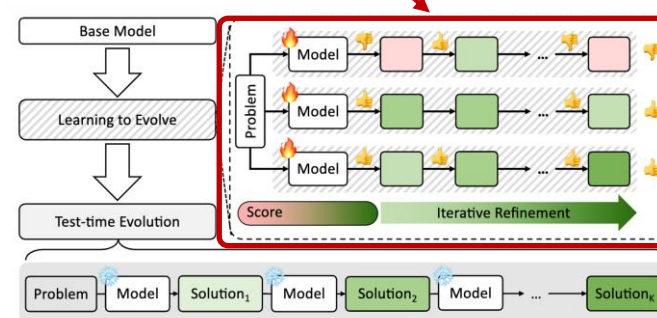
	Mathematics Erdős' Min. Overlap (↓)	Kernel Eng. (TriMul) A100 (↓) H100 (↓)	Algorithms (AtCoder) Heuristic Contest 39 (↑)	Biology Denoising (↑)	
Best Human	0.380927 [20]	4531 μs	1371 μs	566,997 [56]	0.64
Prev. Best AI	0.380924 [50]	N/A	N/A	558,026 [37]	N/A
TTT-Discover	0.380876	2198 μs	1161 μs	567,062	0.71



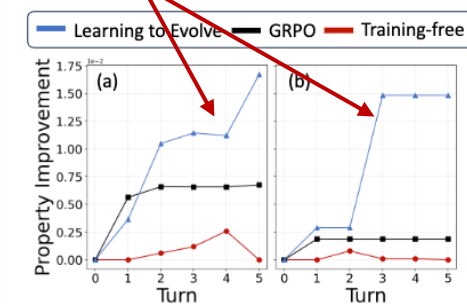
TTT-Discover

- **Problem:** Static inference cannot keep improving on difficult open-ended discovery tasks at test time.
- **Method:** Use test-time training to update the model on the current problem with verifiable feedback.
- **Result:** Outperform prior best human or AI systems across multiple discovery tasks.

Evaluator feedback is assigned across refinement turns.



Learning improves over turns beyond training-free search.

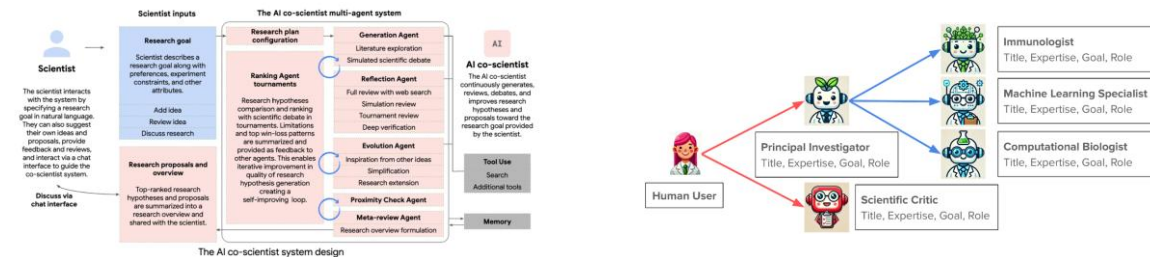
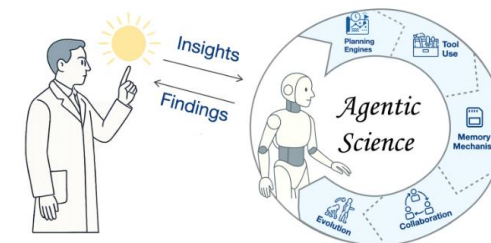
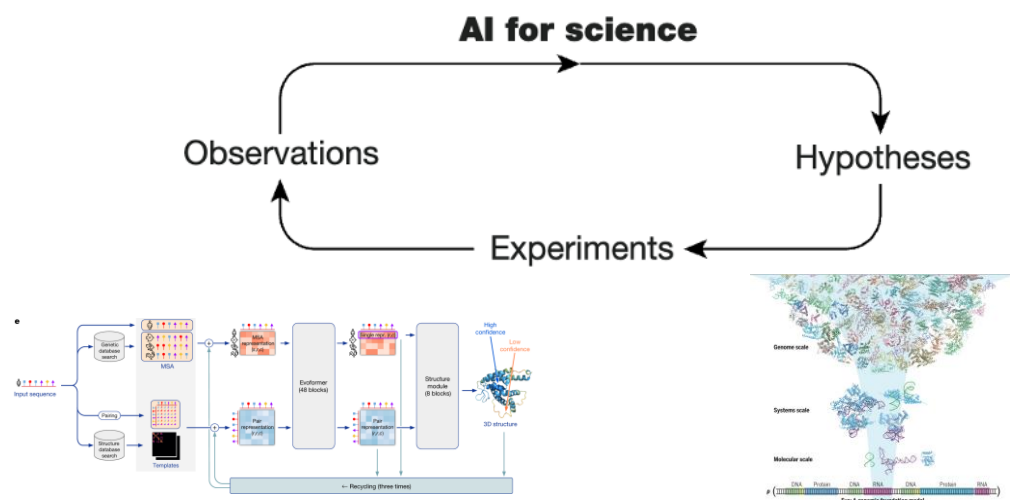


Learning to Evolve (LtE)

- **Problem:** Outcome-level feedback is too coarse for multi-turn self-evolution.
- **Method:** Assign evaluator feedback across refinement trajectories to learn a stronger refinement policy.
- **Result:** Improve iterative refinement more effectively than GRPO and training-free baselines.

Applications: Trustworthy FMs for Science

- **AI4Science** is moving from **model prediction** to **agentic discovery**, where AI systems **analyse** scientific data, **plan** actions, **use** tools or experiments, **check** feedback, and **improve** the next candidate.



AlphaFold for protein folding. **Evo** for genome understanding.

Co-scientist for automatic research. **Virtual Lab** for drug discovery.

- Focus on **one-step prediction** for fixed scientific tasks.
- Input scientific data and output **structures, properties, or labels**.

- Move from prediction to **closed-loop scientific discovery**.
- Agentic systems **analyse** data, **use** tools, and **refine** with feedback.

Model-centric AI4Science

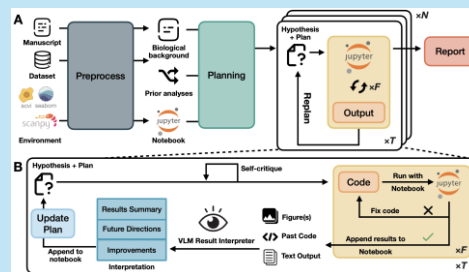
Agent-centric AI4Science

Applications: Trustworthy FMs for Science

- **AI4Science** is advancing toward **agentic discovery**: AI systems can **analyse** biological data, **use** scientific tools, **evolve** code, **plan** research, and **refine** discoveries through feedback.

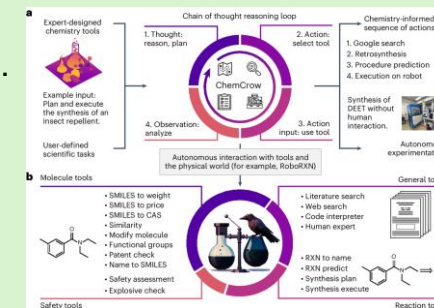
CellVoyager: AI CompBio Agent

- **Analyse** complex cells.
- **Generate** biological hypotheses.
- **Support insight discovery** in computational biology.



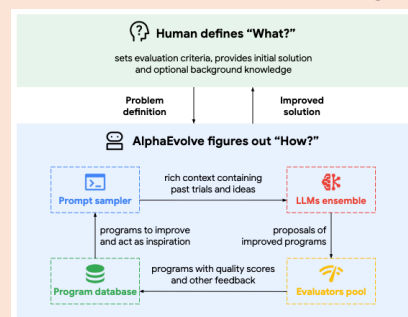
ChemCrow: Chemistry agent

- Use **LLM as the orchestrator**.
- Use **chemistry tools** for checking and calculation.
- **Boost drug discovery and materials design**.



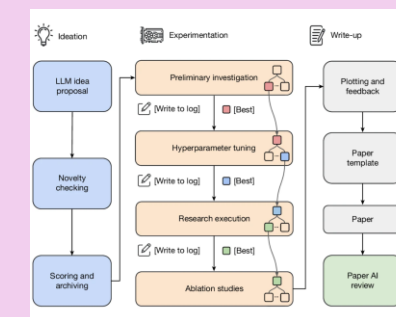
AlphaEvolve: Code-Based Scientific Discovery

- **Evolve** algorithms through **code edits**.
- Use **automatic evaluators** as feedback.
- **Find new results** in math and compute optimization.



The AI Scientist: Automated Research

- **Generate** research plans.
- **Run experiments** and analyze results.
- **Write papers** and perform automated **review**.



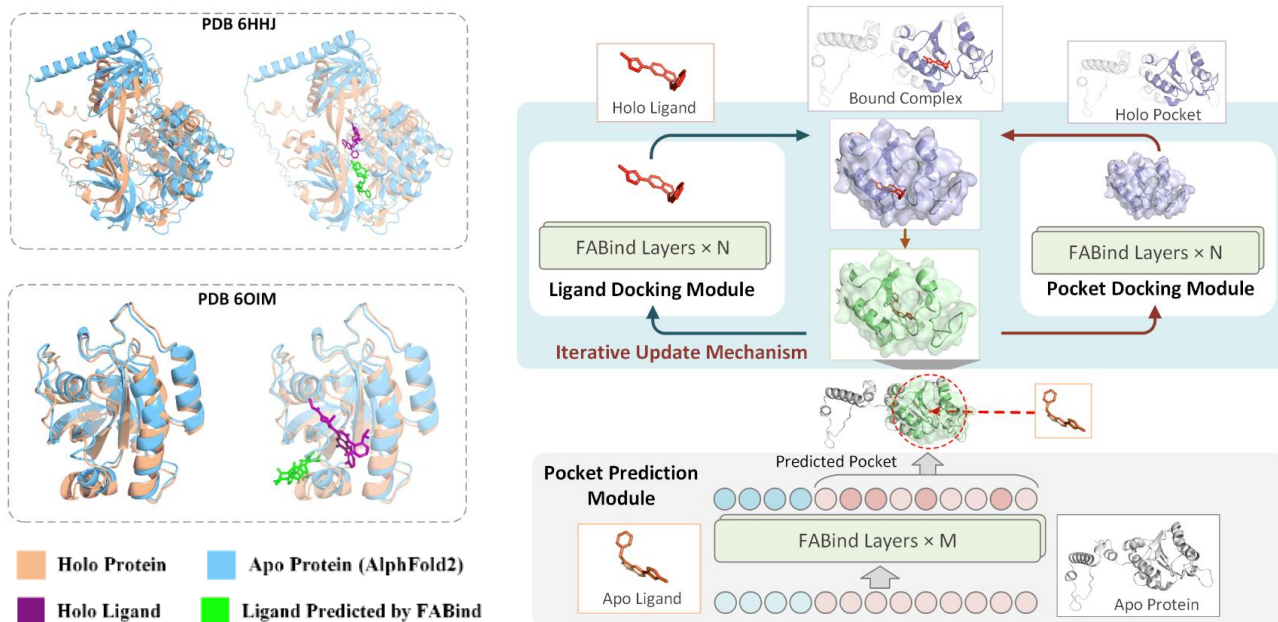
Application Case: Molecular Docking

Problem: Docking is hard with unknown pockets and flexible proteins.

- Proteins are **flexible**.
- **Lack of prior information** about pockets.
- Existing flexible docking is **slow**.

Method: Predicts the pocket and docks the ligand together in one model.

- **Find** likely pocket residues.
- **Place** the ligand into the pocket.
- **Update** ligand and pocket together.

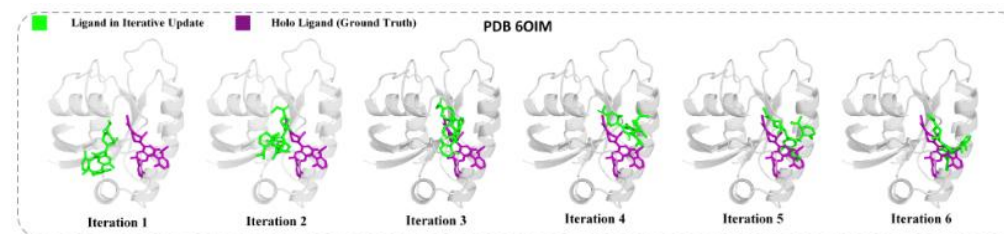


Results: FABFlex is accurate and fast.

- **Correct** ligand placement in about **40%** cases.
- About **0.49s per complex**.
- Over **200x faster** than DynamicBind method.

Method	Ligand RMSD										Average Runtime (s)		
	On All Cases					On Unseen Protein Receptors							
	Percentiles ↓				% Below ↑	Percentiles ↓				% Below ↑			
	25%	50%	75%	Mean	< 2Å	< 5Å	25%	50%	75%	Mean	< 2Å	< 5Å	
<i>Traditional Docking Software</i>													
Vina	4.79	7.14	9.21	7.14	6.67	27.33	5.27	7.06	8.84	7.15	6.25	23.21	205*
Glide	2.84	5.77	8.04	5.81	14.66	40.60	2.38	5.01	7.17	5.21	21.36	49.51	1405*
Gnina	2.58	5.17	8.42	5.76	19.32	48.47	2.03	4.96	<u>7.35</u>	<u>5.33</u>	24.55	50.91	146
<i>Deep Learning-based Rigid Docking Methods</i>													
TankBind	2.82	4.53	7.79	7.79	8.91	54.46	2.88	4.45	7.53	7.60	4.39	58.77	0.87
FABind	2.19	3.73	8.39	6.63	22.11	60.73	2.73	4.83	9.35	7.15	8.77	50.88	0.12
FABind+	1.58	2.79	6.69	5.63	35.64	66.01	1.93	3.13	8.59	6.76	27.19	57.89	0.16
DiffDock	1.82	3.92	6.83	6.07	29.04	60.73	1.97	4.82	8.03	7.41	26.32	51.75	82.83
DiffDock-L	<u>1.55</u>	3.22	6.86	5.99	<u>36.75</u>	62.58	<u>1.86</u>	<u>3.16</u>	9.09	7.14	<u>29.82</u>	61.40	58.72
<i>Deep Learning-based Flexible Docking Methods</i>													
DynamicBind	1.57	3.16	7.14	6.19	33.00	64.69	2.23	4.02	10.23	8.27	20.18	54.39	102.12
FABFlex	1.40	<u>2.96</u>	6.16	5.44	40.59	68.32	1.81	3.51	8.03	7.17	32.46	<u>59.65</u>	0.49

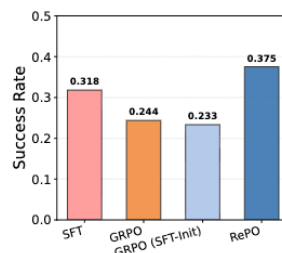
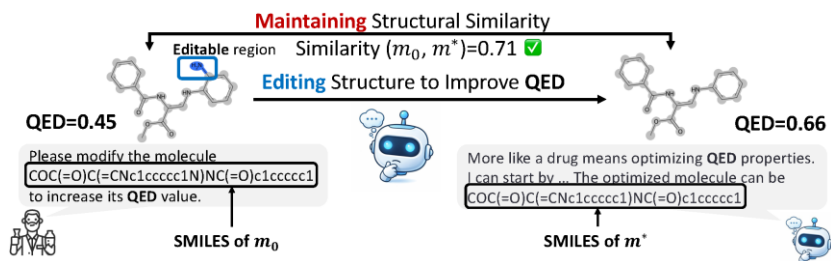
Notes: The best results are highlighted in bold, and the second best results are underlined. The average runtime for each method is presented in seconds. The asterisk (*) indicates that the method is executed on the CPU. The left part of the table compares ligand RMSD on all test cases, while the right part provides a more rigorous comparison for those cases involving protein receptors that were unseen during training process.



Application Case: Molecular Optimization

Problem: Final-answer-only supervision collapses the reasoning ability.

- SFT **only generates answers** without reasoning trajectories.
- RLVR rewards are **sparse and** difficult to provide optimization signals.
- SFT-initialized RL **inherits weak exploration**.



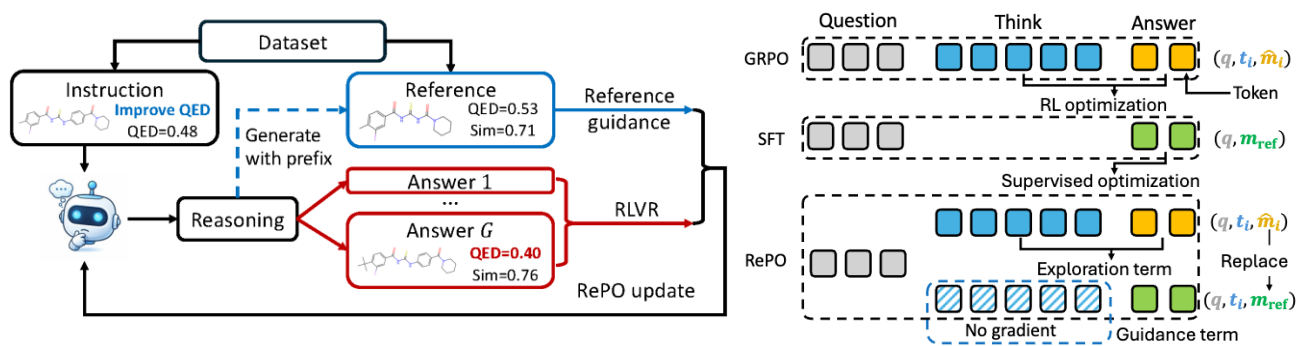
Results: RePO enhances the model in optimizing properties.

- Higher success rate** for single/multiple property optimization.
- Better balance** of molecular property and chemical similarity.
- Robust** on unseen instruction styles for optimizing molecules.

Task type	Objective	Metric	Base Model	Distill-SFT	SFT	GRPO	GRPO (SFT init)	RePO
Structure-based optimization	AddComponent	SR	0.086	0.100	0.238	0.005	0.246	0.307
		Sim	0.763	0.604	0.619	0.992	0.635	0.778
	DelComponent	SR	0.107	0.188	0.203	0.008	0.232	0.158
		Sim	0.864	0.682	0.755	0.994	0.759	0.887
	SubComponent	SR	0.092	0.128	0.153	0.008	0.176	0.140
		Sim	0.057	0.078	0.366	0.053	0.420	0.429
Property optimization	QED	SR	0.188	0.208	0.297	0.138	0.223	0.312
		Sim	0.693	0.594	0.697	0.889	0.863	0.756
		SR×Sim	0.130	0.124	0.207	0.123	0.192	0.236
	LogP	SR	0.268	0.234	0.298	0.379	0.212	0.415
		Sim	0.627	0.579	0.692	0.806	0.863	0.715
		SR×Sim	0.168	0.135	0.206	0.305	0.183	0.297
	MR	SR	0.252	0.214	0.359	0.214	0.265	0.399
		Sim	0.685	0.619	0.663	0.880	0.850	0.736
		SR×Sim	0.173	0.132	0.238	0.188	0.225	0.294

Method: RePO trains with RLVR plus final-answer reference guidance.

- Sample** reasoning and molecules.
- Reward** successful, valid optimization.
- Guide** only the final molecule with the reference.



Modify the molecule Cc1ccc(NC(=O)C(C)C(=O)N2CCCC2)cc1Br to have a **lower MR** value

MR 87.10

We need to understand the impact of different chemical modifications on the MR value, including steric hindrance, electronegativity, and polarizability.

Analysing the optimization target

The structure can be represented as follows:
CC(N)A/V/V/V/V/CC
 Invalide Expression

First, we need to consider the structure of the molecule: carbonyl group, bromine atom, nitrogen atom, benzene ring.

Analysing the molecular structure

The MR or Magic Angle Spinning value is determined by the molecular structure and the presence of functional groups in the molecule.

Incorrect analysis

To lower the MR value, we can remove the nitrogen atom (N) from the heterocyclic ring...

Incorrect optimization

The modified structure can be represented as:
CC(=O)A/V/V/V/V/CC
 Invalide Expression

Identify the key component

Conducting optimization

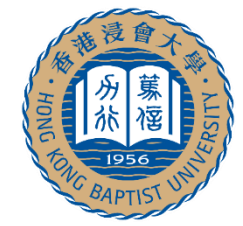
Modify bromine (Br) to chlorine (Cl), reducing the steric hindrance and potentially lowering the MR.

Answer: Cc1ccc(NC(=O)C(C)C(=O)N2CCCC2)cc1Cl

MR: 84.41
 Similarity: 0.75 ✓

The removal of the nitrogen atom can lead to a significant decrease in the MR value

Wrong answer ✗



Take-Home Messages

Learning: RL enhances the capability, while unlearning makes models forget what they shouldn't know.

- RL evolves from alignment toward actively enhancing FM reasoning capabilities.
- Unlearning evolves from adapting classical methods toward effective, precise, and robust forgetting.

Reasoning: The paradigm shifts from passive generation to active and verifiable problem-solving.

- LLM reasoning expands the model's capability through more deliberate internal generation.
- Agent reasoning learns to act by interleaving reasoning with external interaction, tool use, and environmental feedback.

Generalization: Self-evolution strengthens FMs through feedback, while applications test reliability under constraints.

- Generalization requires self-evolution: feedback, failures, and repeated attempts must improve future behavior.
- Generalization is validated in AI4Science, where scientific constraints make reliability measurable and useful.

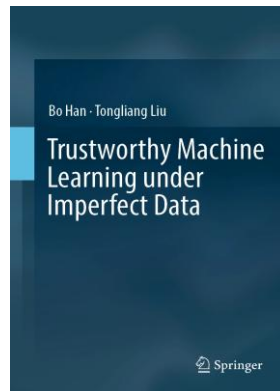
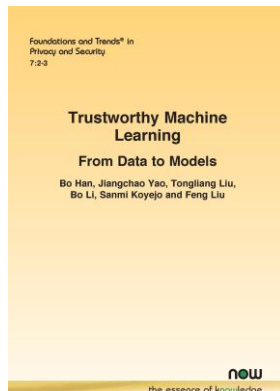
Appendix

- Survey:

- A Survey of Label-noise Representation Learning: Past, Present and Future. arXiv, 2020.

- Books:

- Machine Learning with Noisy Labels: From Theory to Heuristics. Adaptive Computation and Machine Learning series, **The MIT Press**, 2026.
- Trustworthy Machine Learning under Imperfect Data. CS series, **Springer Nature**, 2025.
- Trustworthy Machine Learning: From Data to Models. **Foundations and Trends® in Privacy and Security**, 2025.



- Tutorials and Lectures:

- AAI 2024 Tutorial on Trustworthy Machine Learning under Imperfect Data.
- IJCAI 2024 Tutorial on Trustworthy Machine Learning under Imperfect Data.
- WWW 2025 Tutorial on Trustworthy AI under Imperfect Web Data.
- AAI 2026 Tutorial on The Science and Practice of Machine Unlearning for AI Safety.
- AAI 2026 Tutorial on Trustworthy Machine Reasoning with Foundation Models.
- DeepLearn 2026 Lecture on Trustworthy Machine Learning from Data to Models.

- Workshops:

- IJCAI 2021 Workshop on Weakly Supervised Representation Learning.
- ACML 2022 Workshop on Weakly Supervised Learning.
- RIKEN 2023 Workshop on Weakly Supervised Learning.
- HKBU-RIKEN AIP 2024 Joint Workshop on Artificial Intelligence and Machine Learning.