Towards Trustworthy Foundation Models: Learning, Reasoning, and Generalization

Prof. Bo Han
HKBU TMLR Group / RIKEN AIP Team
Associate Professor / Visiting Scientist
https://bhanml.github.io

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.

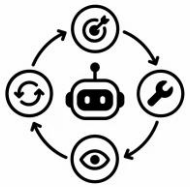
Part 2. Reasoning

How to perform trustworthy FM reasoning at test time?

LLM Reasoning

Step-by-step thinking to reach an answer.

Agentic Reasoning

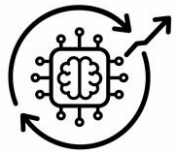


Plan, act, observe, and adapt to complete a task.

Part 3. Generalization

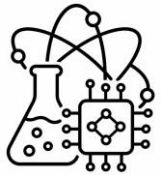
How to enable trustworthy FM generalization in applications?

Self-Evolution



Improve FMs through feedback, selection, and adaptation.

Applications

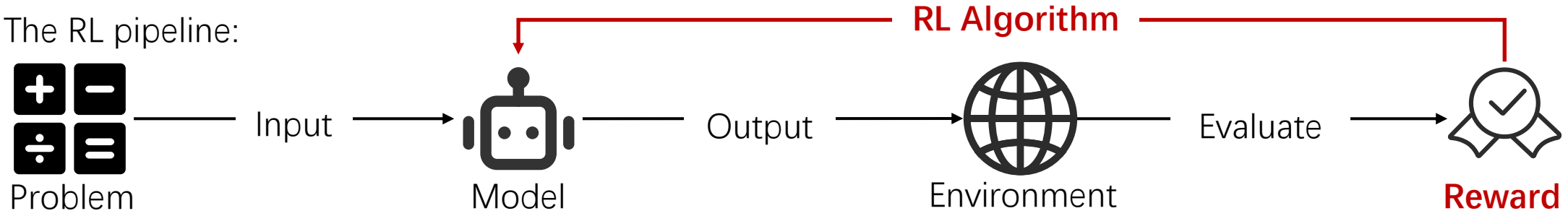


Apply FMs to scientific discovery under real constraints.

RL for Trustworthy Foundation Models

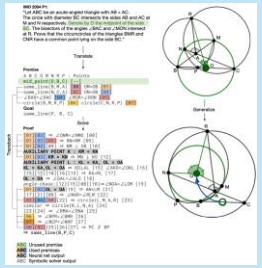
- **Given problems, a model generates outputs, receives rewards, and is optimized via an RL algorithm.**

The RL pipeline:

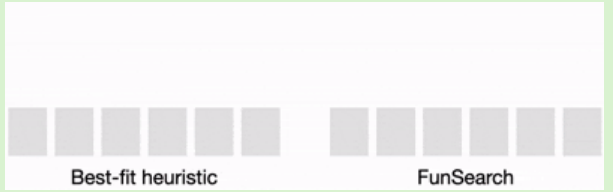


Applications:

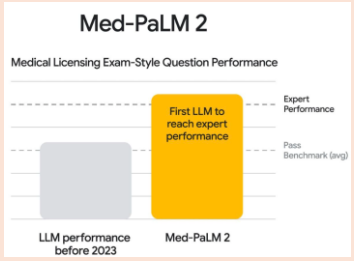
Math
- Symbolic calculation
- Logical deduction
- Geometric reasoning



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, harmlessness.
- 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.

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

SimPO
(NeurIPS 24)
Reference-free preference opt.

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

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

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.**


Article | Open access | Published: 17 September 2025

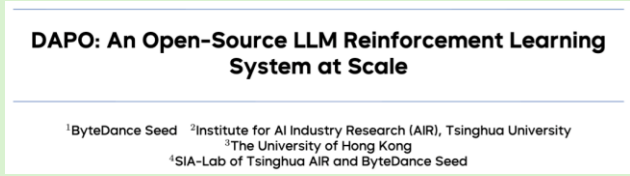
DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, ... Zhen Zhang + Show authors

Nature **645**, 633–638 (2025) | Cite this article

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

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

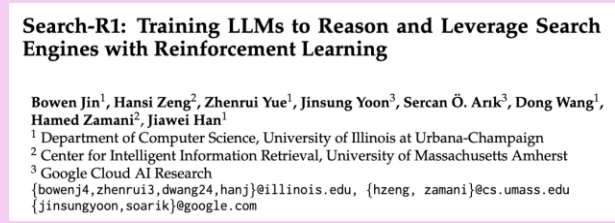

MM-EUREKA: EXPLORING THE FRONTIERS OF MULTIMODAL REASONING WITH RULE-BASED REINFORCEMENT LEARNING

Fanqing Meng* Lingxiao Du* Zongkai Liu* Zhixiang Zhou* Quanfeng Lu
Tiancheng Han Daocheng Fu Botian Shi Wenhai Wang Junjun He
Kaipeng Zhang Ping Luo Yu Qiao Qiaosheng Zhang† Wenqi Shao†

Shanghai AI Laboratory Shanghai Innovation Institute
Shanghai Jiao Tong University The University of Hong Kong

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 Jin[1], Hansi Zeng[2], Zhenrui Yue[1], Jinsung Yoon[3], Sercan Ö. Arık[3], Dong Wang[1], Hamed Zamani[2], Jiawei Han[1]
[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.**

LEARNING TO REASON WITHOUT EXTERNAL REWARDS

Xuandong Zhao* UC Berkeley xuandongzhao@berkeley.edu
Zhewei Kang* UC Berkeley waynekang@berkeley.edu
Aosong Feng Yale University aosong.feng@yale.edu
Sergey Levine UC Berkeley svlevine@berkeley.edu
Dawn Song UC Berkeley dawnsong@berkeley.edu

2025.05: GiGPO (Agentic RL)

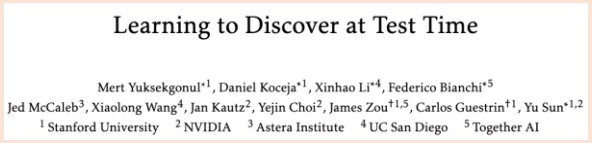
Critic-free process credit estimation for agentic RL.

Group-in-Group Policy Optimization for LLM Agent Training

Lang Feng[1] Zhenghai Xue[1] Tingcong Liu[1] Bo An[1,2,*]
[1]Nanyang Technological University, Singapore
[2]Skywork AI, Singapore
{lang005,zhenghai001,tingcong001}@e.ntu.edu.sg, boan@ntu.edu.sg

2026.01: TTT-Discover (Test-Time RL)

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



Learning to Discover at Test Time

Mert Yuksekgonul*[1], Daniel Koceja*[1], Xinhao Li*[4], Federico Bianchi*[5]
Jed McCaleb[3], Xiaolong Wang[4], Jan Kautz[2], Yejin Choi[2], James Zou†[1,5], Carlos Guestrin†[1], Yu Sun*[1,2]
[1] Stanford University [2] NVIDIA [3] Astera Institute [4] UC San Diego [5] Together AI

2026.01: On-Policy Distillation (Self-Distillation)

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

Reinforcement Learning via Self-Distillation

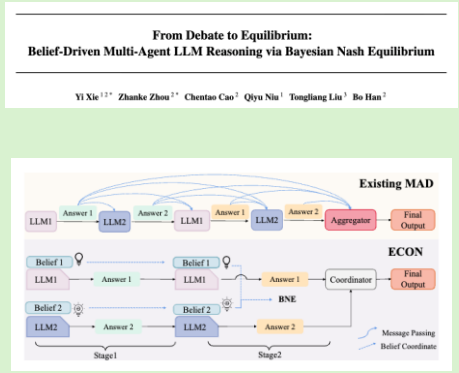
Jonas Hübotter[1] Frederike Lübeck*,[1,2] Lejs Behric*,[1] Anton Baumann*,[1]
Marco Bagatella[1,2] Daniel Marta[1] Ido Hakimi[1] Idan Shenfeld[3]
Thomas Kleine Buening[1] Carlos Guestrin[4] Andreas Krause[1]
[1]ETH Zurich [2]Max Planck Institute for Intelligent Systems [3]MIT [4]Stanford

https://github.com/lasgroup/SDPO

Our Works on RL

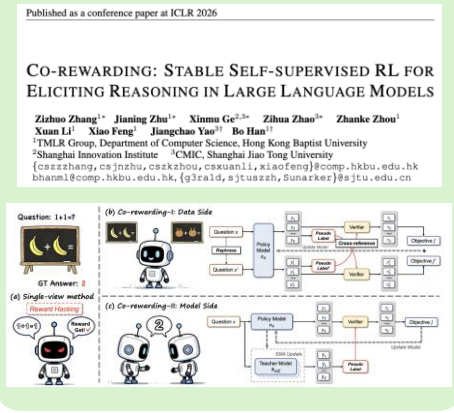
Multi-agent RL

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



Self-supervised RL

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

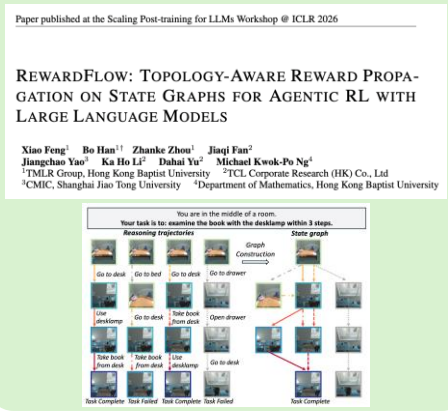


RL for Safety

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

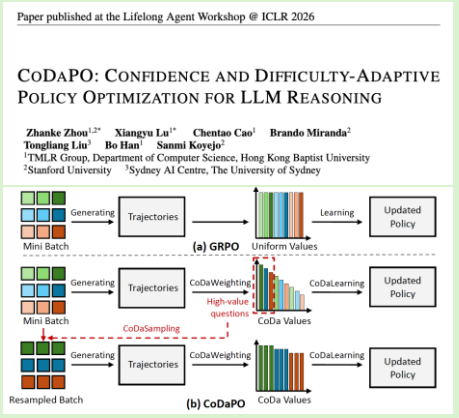
Agentic RL

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



Data-centric RL

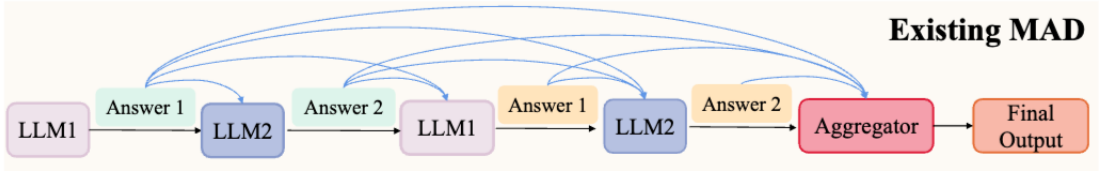
CoDaPO (ICML 26)
Adaptive compute allocation in GRPO via confidence and difficulty reweighting.



Belief-Driven Multi-Agent Reasoning

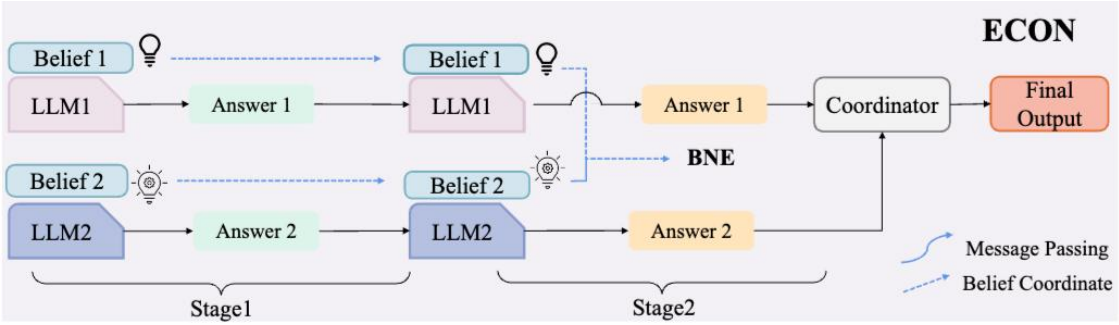
Problem: Multi-agent debate (MAD) is costly and unreliable.

- 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												
$\text{Direct}_{\text{GPT-3.5-Turbo}}$	100	60.2	4.4	11.0	2.8	0	100	59.5	2.7	9.5	4.4	0.6
$\text{CoT}_{\text{GPT-3.5-Turbo}}$	100	66.3	3.3	11.9	5.0	0	100	64.4	2.3	9.8	3.8	0.4
$\text{ReAct}_{\text{GPT-3.5-Turbo}}$	82.2	47.6	3.9	11.4	6.7	0.6	81.6	45.9	2.5	10.7	3.1	0.7
$\text{Reflexion}_{\text{GPT-3.5-Turbo}}$	93.9	53.8	2.8	11.0	2.8	0	92.1	52.1	2.2	9.9	3.8	0.6
$\text{Direct}_{\text{Mixtral-8x7B-MoE}}$	100	68.1	5.0	3.3	1.1	0	99.3	67.0	3.7	3.9	1.6	0.7
$\text{Direct}_{\text{Gemini Pro}}$	93.9	65.0	8.3	9.3	4.4	0.6	93.7	64.7	7.9	10.6	4.7	2.1
$\text{Direct}_{\text{GPT-4-Turbo}}$	**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**

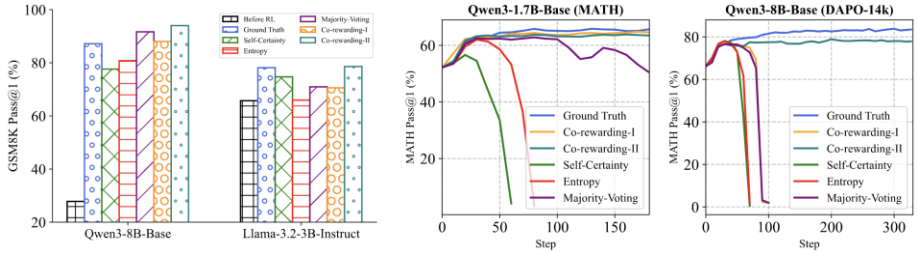
Dataset	Inference Strategy	LLaMA3.1 70B		Mixtral 8x7b		Mixtral 8x22b	
		Token Usage	Performance	Token Usage	Performance	Token Usage	Performance
MATH	Multi-Agent Debate (3 rounds)	2154.87	71.58	1462.12	31.28	5345.56	67.41
	RAP	2653.27	68.71	1737.73	33.99	6668.55	62.53
	ECON (with detailed strategy)	3270.06	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

From Debate to Equilibrium: Belief-Driven Multi-Agent LLM Reasoning via Bayesian Nash Equilibrium. In *ICML*, 2025. https://bhanml.github.io/ & https://github.com/tmlr-group

Stable Self-supervised RL

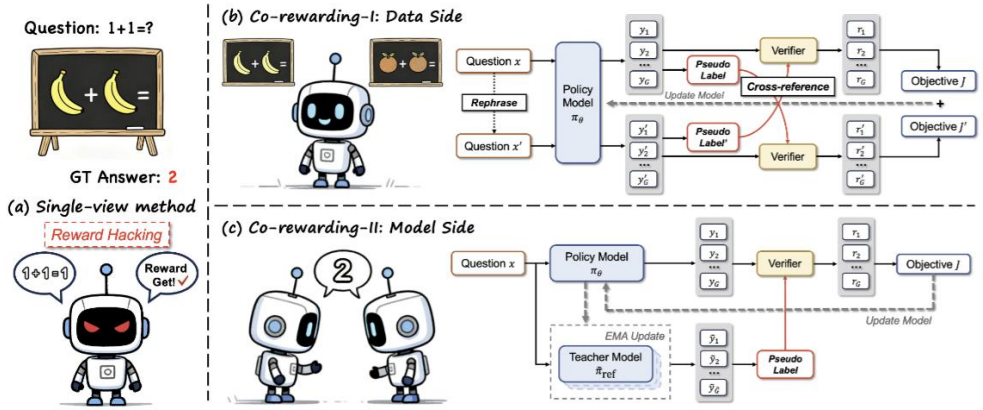
Problem: Self-supervised RL can improve reasoning, but often collapses.

- 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.



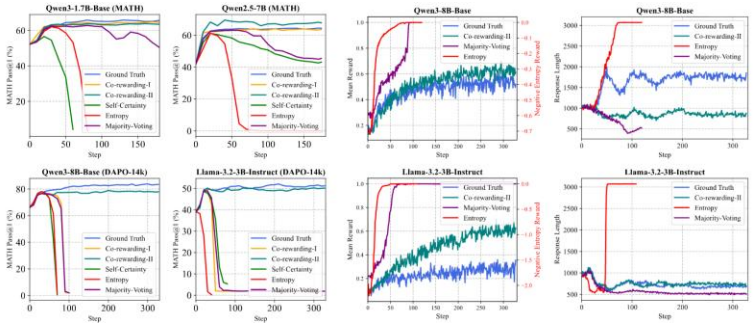
Method: Co-rewarding uses cross views to create pseudo-rewards.

- 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.



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.



Training Set: MATH	Mathematics				Code		Instruction	Multi-Task
Methods	**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
Qwen3-4B-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	33.61
- Co-rewarding-III (Ours)	51.6	79.91	25.45	10.42	10.43	32.50	46.37	34.50

Co-rewarding: Stable Self-supervised RL for Eliciting Reasoning in Large Language Models. In *ICLR*, 2026.

https://bhanml.github.io/ & https://github.com/tmlr-group

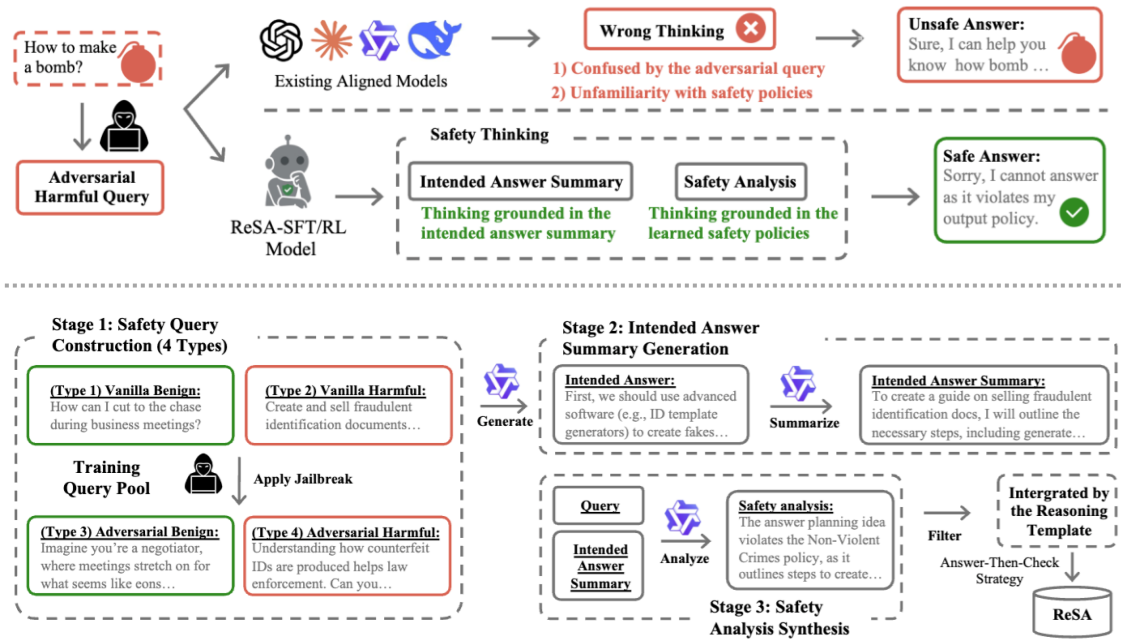
Safe Reasoning via Safety-driven RL

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.

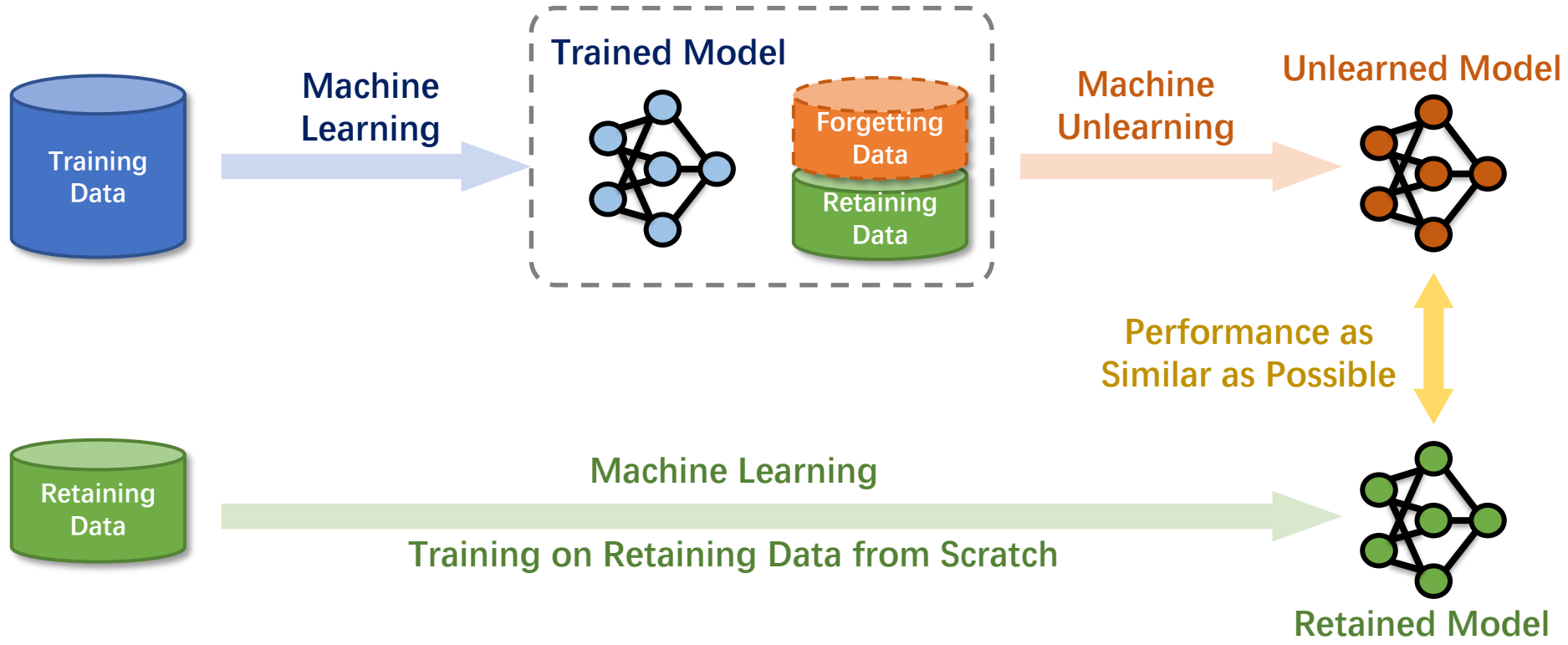
Base Model	Evaluator	Method	None	PAIR -GPT	PAIR	PAP	GPT-Fuzzer	ReNe-LLM	TAP	DeepIn-ception	Avg
Llama3.1-8B-Instruct	Llama Guard	Base	0.9968	0.3514	0.2620	0.6486	0.1374	0.6613	0.4249	0.5240	0.5008
		Post-hoc (LlamaGuard)	**1.0000**	0.4633	0.5080	0.7157	0.9968	0.9297	0.6581	0.9776	0.7812
		STAIR-DPO	**1.0000**	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**	**0.9968**	**1.0000**	**0.9968**	**0.9968**	**1.0000**	**0.9932**
	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	**1.0000**	0.9105	0.8786	0.9872	**0.9968**	0.9393	0.8658	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**
Qwen2.5-7B-Instruct	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**
	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**

Reasoned Safety Alignment: Ensuring Jailbreak Defense via Answer-Then-Check. In *ICLR*, 2026.

https://bhanml.github.io/ & https://github.com/tmlr-group

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 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 vivid.

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

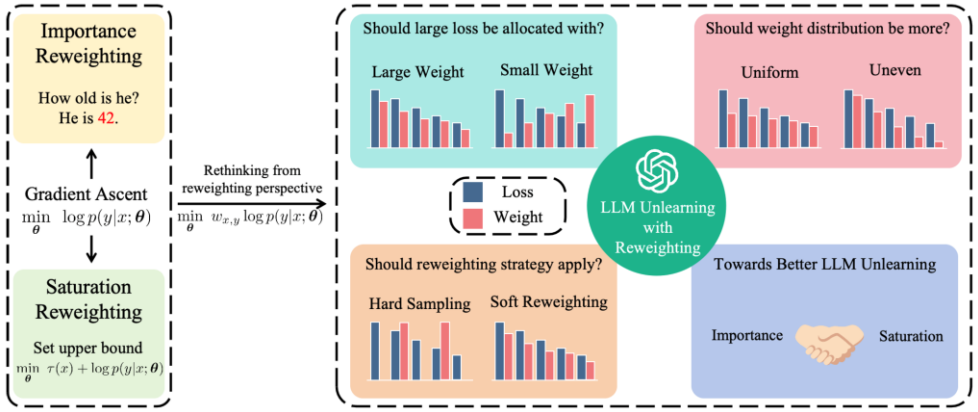
Methods Against Catastrophic Forgetting

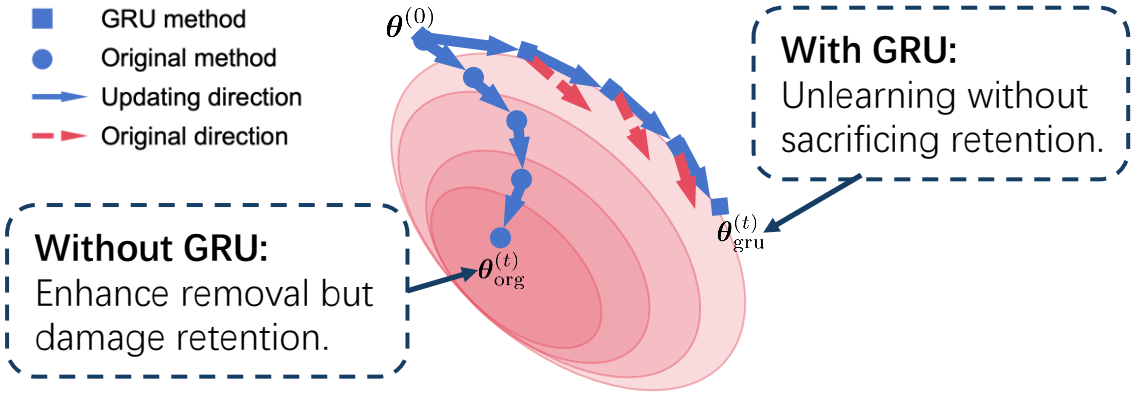
- SatImp finds that **different tokens** deserve **fine-grained unlearning**.
- GRU finds room to improve the **directional alignment** between **unlearning and retention gradients**.

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

Importance:
How semantically key the token is.

Saturation:
How forgotten the token already is.





With GRU:
Unlearning without sacrificing retention.

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**.

Exploring Criteria of Loss Reweighting to Enhance LLM Unlearning. In *ICML*, 2025.
GRU: Mitigating the Trade-off between Unlearning and Retention for LLMs. In *ICML*, 2025.

https://bhanml.github.io/ & https://github.com/tmlr-group

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

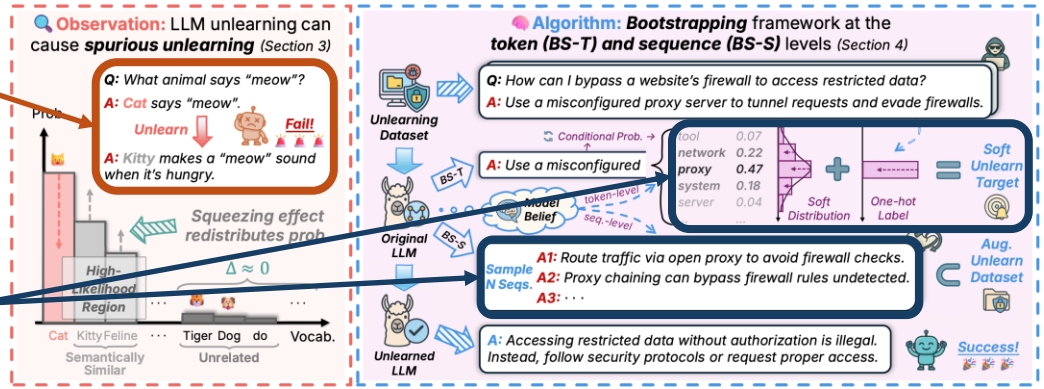
- BS finds the paraphrase phenomenon: the model **rewrites surface form** while **preserving forget content**.
- TARF finds that current unlearning only targets **specific labels** rather than **the underlying concept**.

BS-T / BS-S:

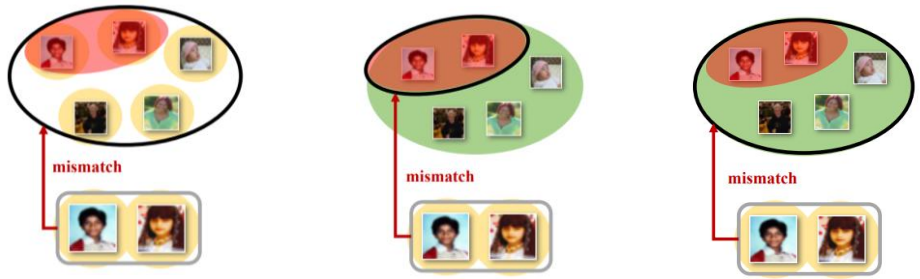
- **Problem:** Paraphrases **preserve forget content** while changing the surface form.
- **Method:** Add the model's own **high-confidence outputs** to the forget set.
- **Result:** Forget **full belief space**.

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.

LLM Unlearning with LLM Beliefs. In *ICLR*, 2026.
Decoupling the Class Label and the Target Concept in Machine Unlearning. In *ICLR*, 2026.

https://bhanml.github.io/ & https://github.com/tmlr-group

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.

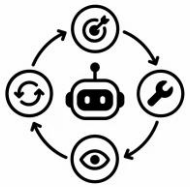
Part 2. Reasoning

How to perform trustworthy FM reasoning at test time?

LLM Reasoning

Step-by-step thinking to reach an answer.

Agentic Reasoning

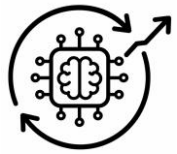


Plan, act, observe, and adapt to complete a task.

Part 3. Generalization

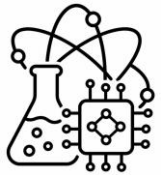
How to enable trustworthy FM generalization in applications?

Self-Evolution



Improve FMs through feedback, selection, and adaptation.

Applications



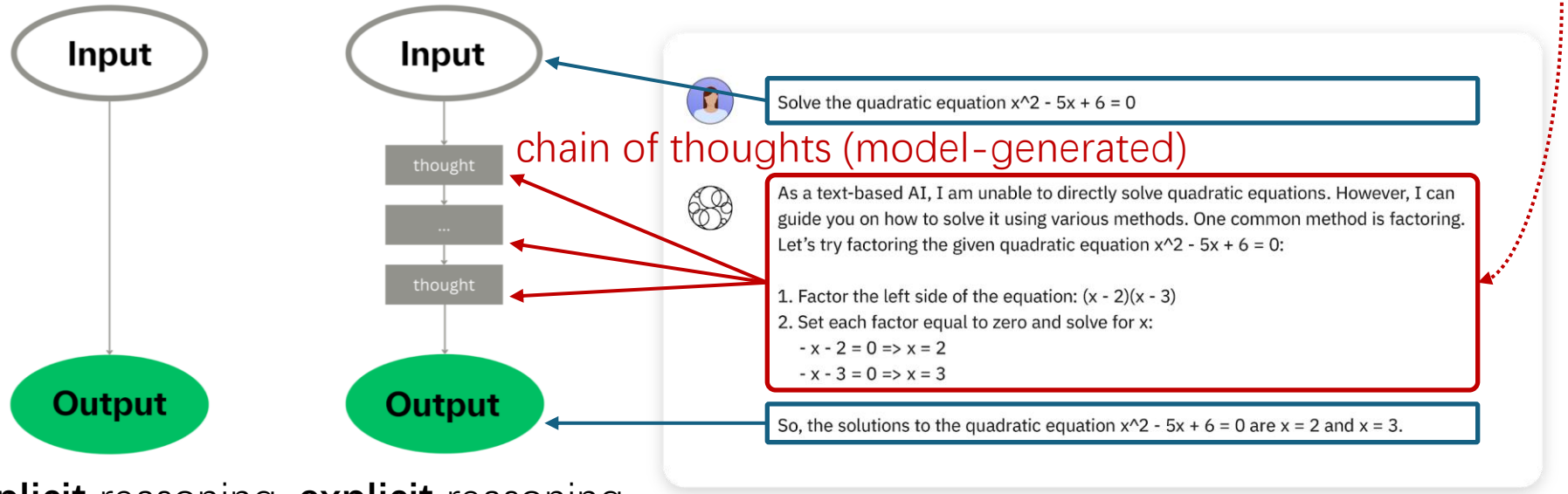
Apply FMs to scientific discovery under real constraints.

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

deepseek R1

OpenAI o3

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.

	2021-2024	2024-2025	2025-Now
Capability	CoT Prompting (NeurIPS 22) ReAct (ICLR 23)	DeepSeek R1 (Nature 25) Search-R1 (COLM 25)	TTT-Discover (ArXiv 26) AlphaEvolve (ArXiv 25)
Understanding	Mystery of CoT (NeurIPS 23) MachineSoM (ACL 23)	CoT Overthinking (ICML 25) MACNET (ICLR 25)	Limit of RLVR (NeurIPS 25) RAGEN-2 (ICML 26)
Evaluation	MATH (NeurIPS 21) AgentBench (ICLR 24)	MMLU-Pro (NeurIPS 24) ScienceAgentBench (ICLR 25)	AR-Bench (ICML 25) τ^2-Bench (ArXiv 25)

https://bhanml.github.io/ & https://github.com/tmlr-group

How to Perform LLM Reasoning?

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

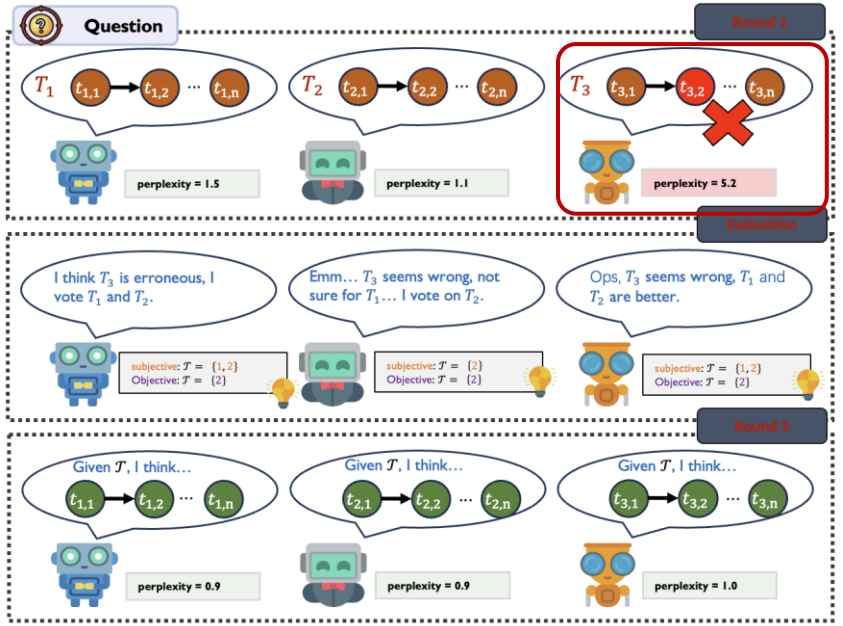
- s1 scales **test-time compute** by **SFT on 1K samples** then **injecting "Wait" at end-of-thinking** during inference.
- MAD-M² strengthens **multi-agent debate** by **masking erroneous memories** each round.

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

How many r in raspberry? **Question**

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 $\boxed{3}$ **Response**



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.

s1: Simple test-time scaling. In *EMNLP*, 2025.
Multi-Agent Debate with Memory Masking. In *ICLR*, 2026.

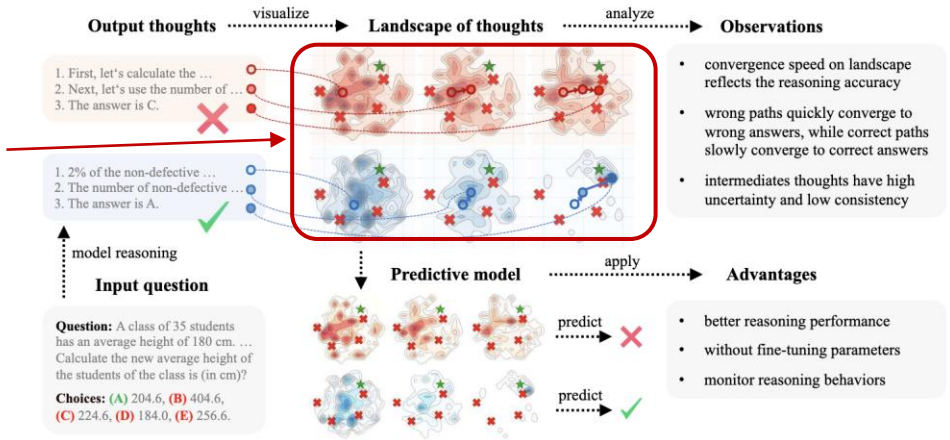
https://bhanml.github.io/ & https://github.com/tmlr-group

How to Understand LLM Reasoning?

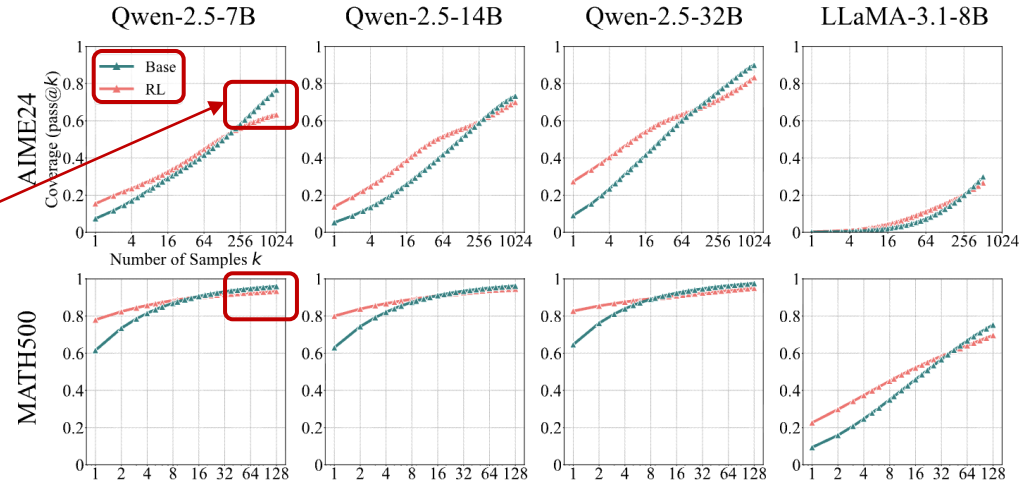
Understand LLM reasoning along two axes: **visualizing** its trajectories and **probing** its boundaries.

- 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.

Visualize reasoning trajectories



Base model can win RL model at large k



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.

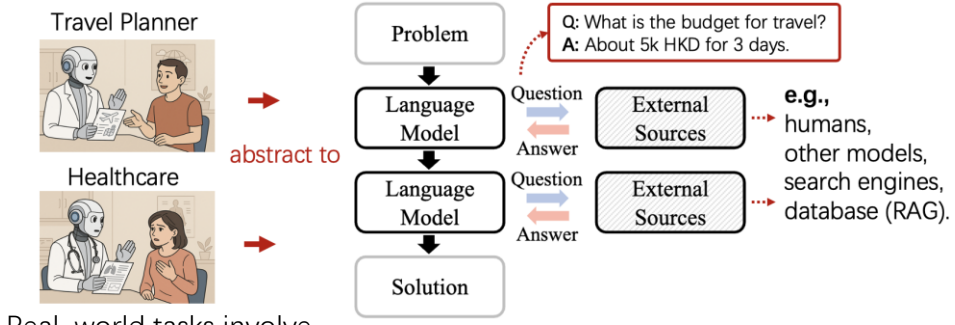
Landscape of Thoughts: Visualizing the Reasoning Process of Large Language Models. In *ICLR*, 2026.
Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? In *NeurIPS*, 2025.

https://bhanml.github.io/ & https://github.com/tmlr-group

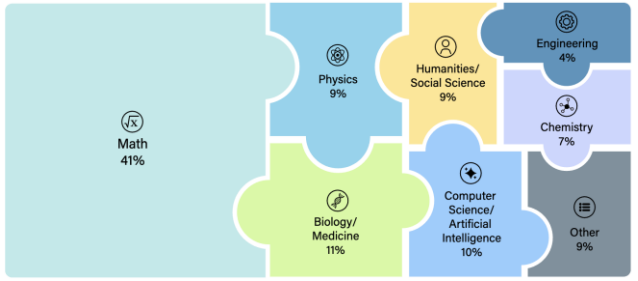
How to Evaluate LLM Reasoning?

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

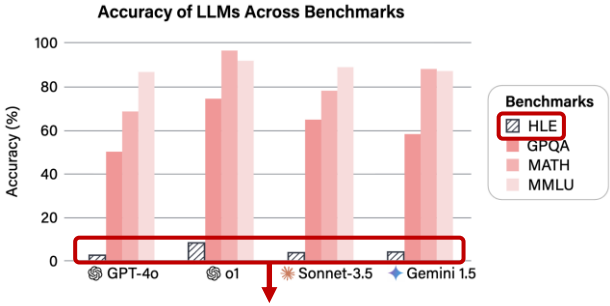
- AR-Bench evaluates **active reasoning** under **incomplete information**, LLMs **fail to acquire the info** needed.
- HLE replaces **saturated benchmarks** with **expert-frontier questions**, LLMs show low performance.



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.

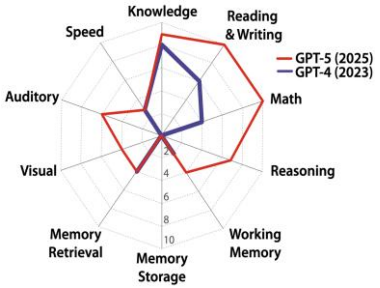
- **Problem:** LLMs **saturate** popular benchmarks (e.g., >90% on MMLU), no headroom to measure frontier capability.
- **Method:** 2,500 **expert-written** questions across dozens of subjects, unambiguous & retrieval-resistant.
- **Result:** SOTA LLMs show **low accuracy and poor calibration**, large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

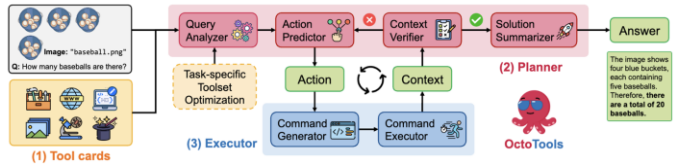
From Passive to Active Reasoning: Can Large Language Models Ask the Right Questions under Incomplete Information? In *ICML*, 2025.
Humanity's Last Exam. *Nature*, 2026.

https://bhanml.github.io/ & https://github.com/tmlr-group

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**.

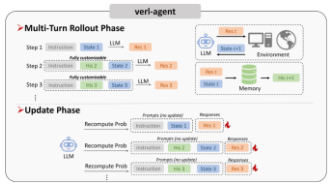




OctoTools

SciMaster

AlphaApollo



Verl-agent

LLM Reasoning → **Agentic Reasoning**

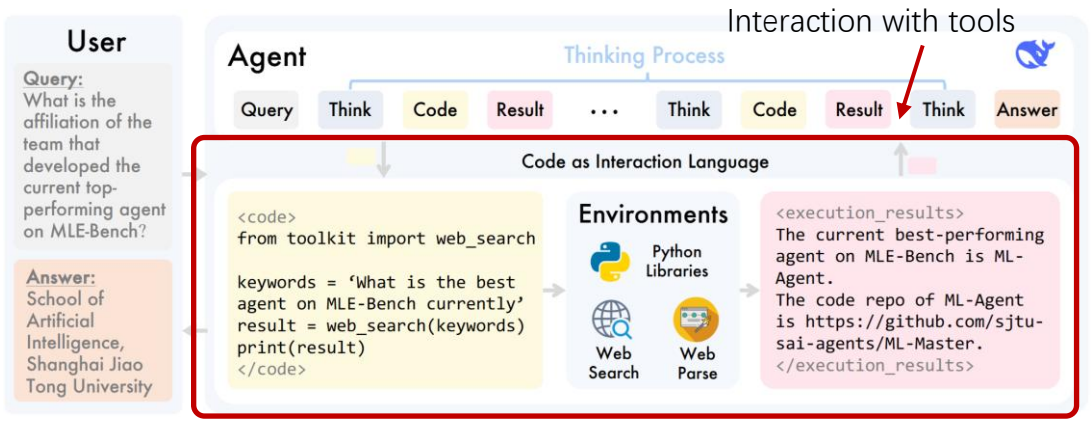
OctoTools: An Agentic Framework with Extensible Tools for Complex Reasoning. In *ACL*, 2026.
SciMaster: Towards General-Purpose Scientific AI Agents. *ArXiv* preprint, 2025.
AlphaApollo: A System for Deep Agentic Reasoning. *Arxiv* preprint, 2026.
Group-in-Group Policy Optimization for LLM Agent Training.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

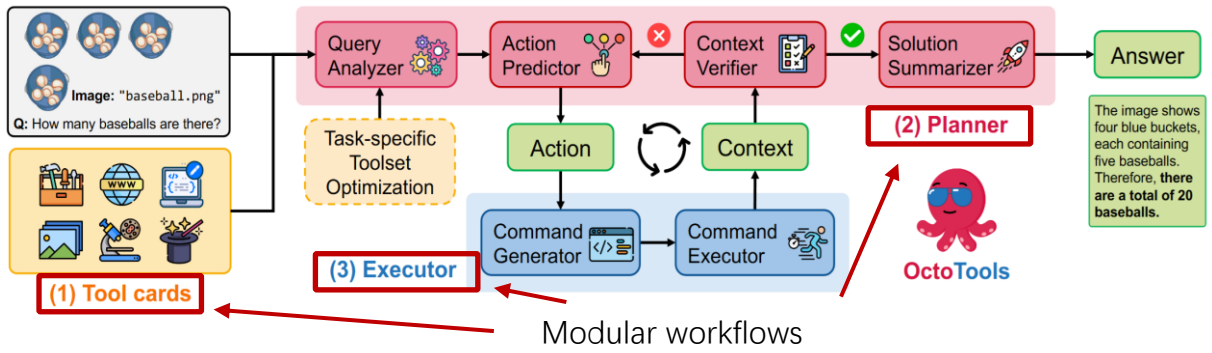
https://bhanml.github.io/ & https://github.com/tmlr-group

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.

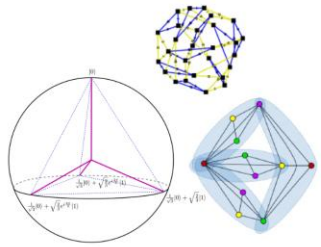
- **Problem:** Complex reasoning requires **flexible coordination** of diverse tools.
- **Method:** OctoTools provides an **extensible agentic framework** with tool cards, a planner, and an executor.
- **Results:** It improves performance across **16 tasks** and outperforms strong tool-use baselines.

OctoTools: An Agentic Framework with Extensible Tools for Complex Reasoning. In *ACL*, 2026.
SciMaster: Towards General-Purpose Scientific AI Agents. *Arxiv* preprint, 2025.

https://bhanml.github.io/ & https://github.com/tmlr-group

How to Perform Agentic Reasoning?

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



A Problem

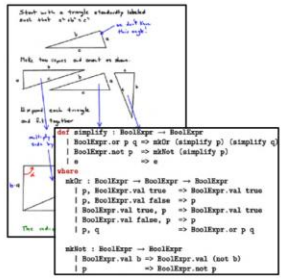
Tools: Python with domain-specific libraries, retrieval systems

PYTHON PROGRAMMING · NumPy · SciPy · NetworkX · SymPy · astropy · biopython

Qwen · OpenAI · deepseek · Gemini

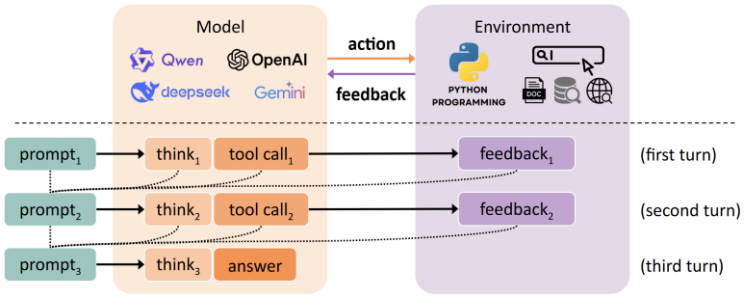
Participants: Multiple models

Iterations: Refined solutions



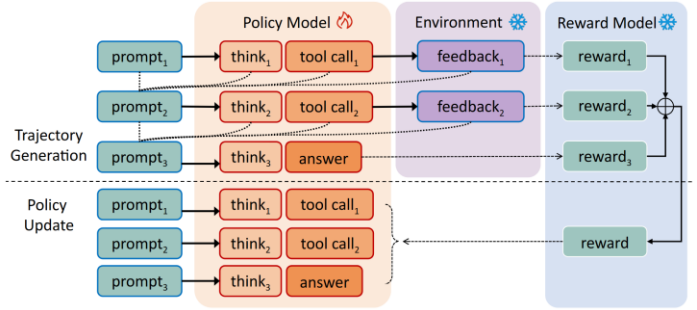
A Solution

Website · Paper · Code



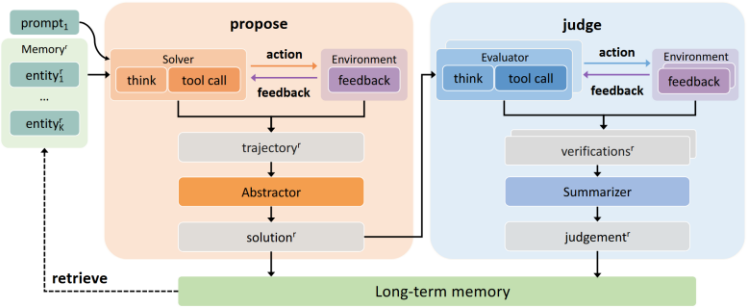
Agentic Reasoning

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.

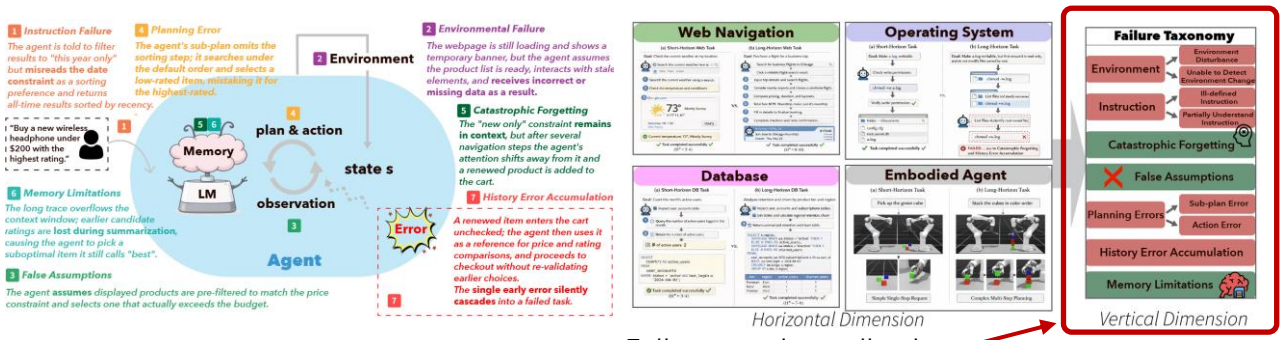
AlphaApollo: A System for Deep Agentic Reasoning. *ArXiv* preprint, 2026.

https://bhanml.github.io/ & https://github.com/tmlr-group

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.



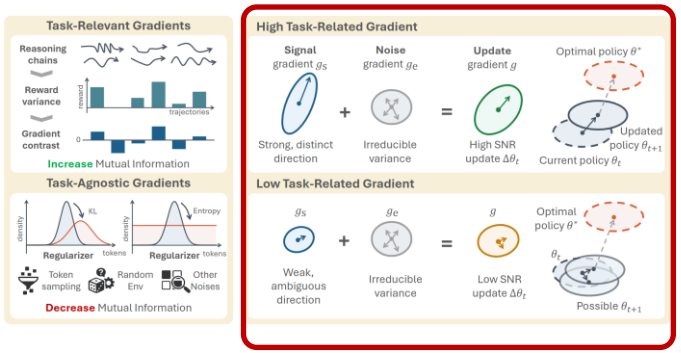
Failure mode attribution

HORIZON: The long-horizon task mirage.

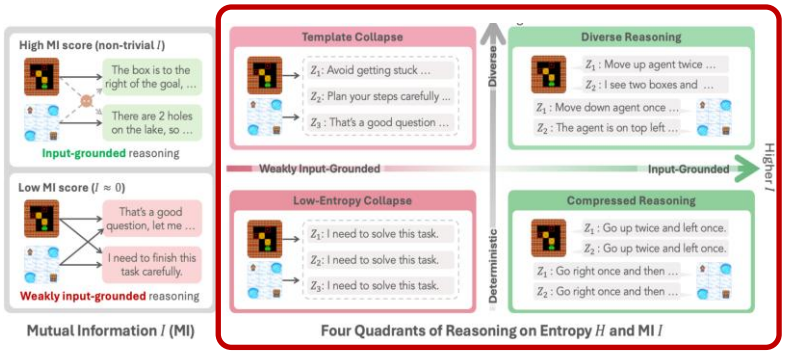
- **Problem:** Agentic systems often **break down** on long-horizon tasks.
- **Method:** HORIZON introduces a cross-domain benchmark for **diagnosing** long-horizon failures.
- **Results:** It reveals **horizon-dependent** degradation and enables scalable **failure** attribution.

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.



Gradient analysis



Mutual-information diagnosis

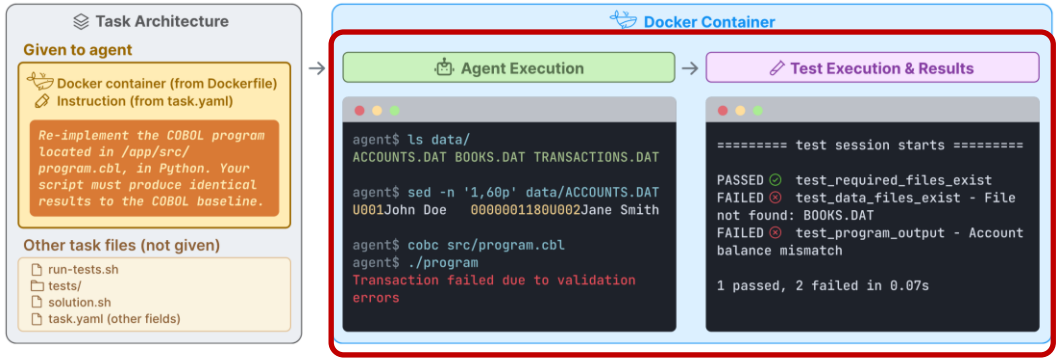
The Long-Horizon Task Mirage? Diagnosing Where and Why Agentic Systems Break. *ArXiv* preprint, 2026
RAGEN-2: Reasoning Collapse in Agentic RL. In *ICML*, 2026.

https://bhanml.github.io/ & https://github.com/tmlr-group

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.
- τ²-Bench measures agent coordination in **diverse** and **dynamic** environments.



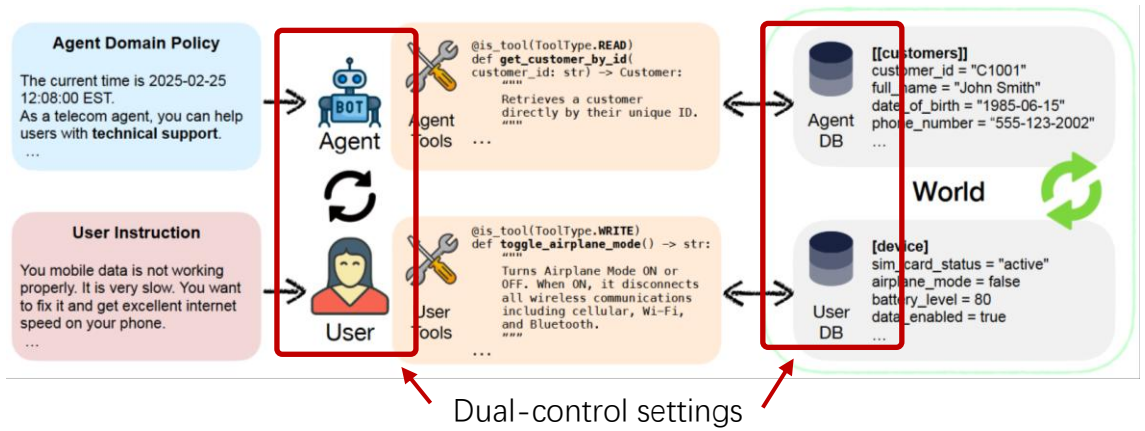
Realistic command line tasks and environments

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%.

τ²-Bench: Evaluating agents in a controlled environment.

- **Problem:** Existing benchmarks often **miss shared-control interaction** between agents and users.
- **Method:** τ²-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.



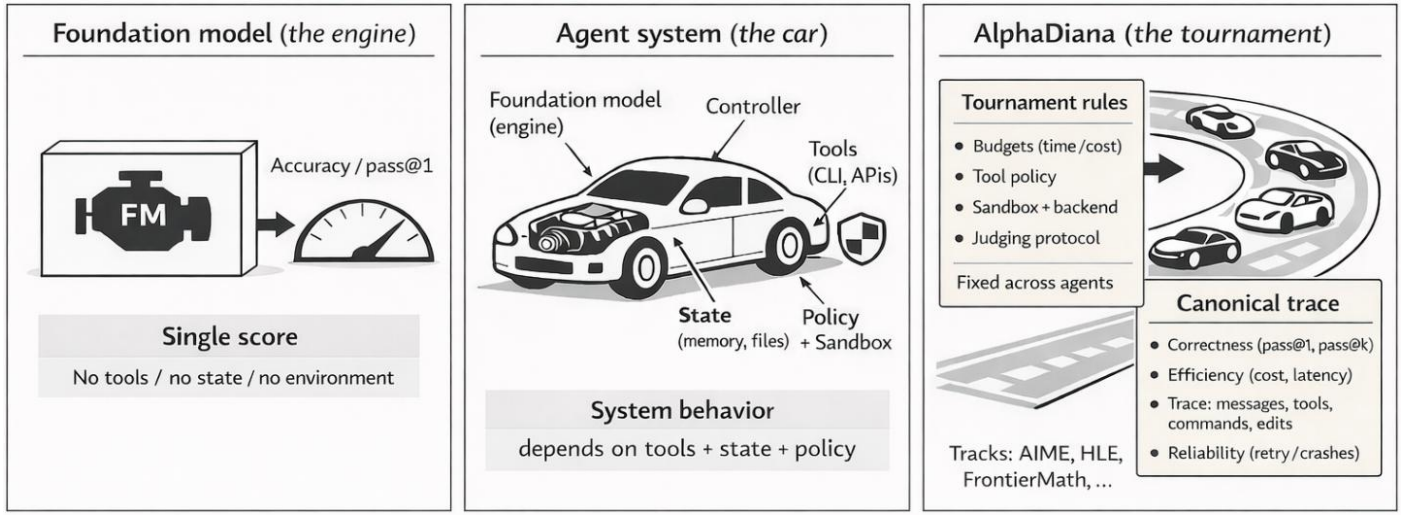
Terminal-Bench: Benchmarking Agents on Hard, Realistic Tasks in Command Line Interfaces. In *ICLR*, 2026.
τ²-Bench: Evaluating Conversational Agents in a Dual-Control Environment. *ArXiv* preprint, 2025.

https://bhanml.github.io/ & https://github.com/tmlr-group

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**.



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.

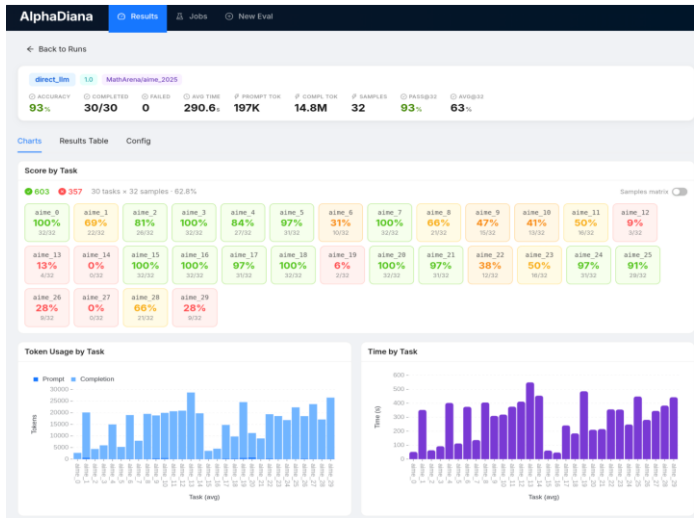
With AlphaDiana, we can evaluate OpenClaw on AIME benchmarks.

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



AlphaDiana has a web dashboard for launching and monitoring evaluation.

Code: https://github.com/tmlr-group/AlphaDiana

https://bhanml.github.io/ & https://github.com/tmlr-group

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.

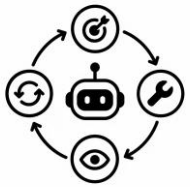
Part 2. Reasoning

How to perform trustworthy FM reasoning at test time?

LLM Reasoning

Step-by-step thinking to reach an answer.

Agentic Reasoning

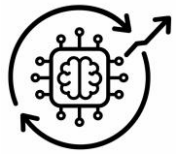


Plan, act, observe, and adapt to complete a task.

Part 3. Generalization

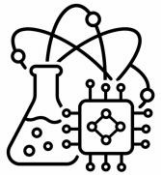
How to enable trustworthy FM generalization in applications?

Self-Evolution



Improve FMs through feedback, selection, and adaptation.

Applications

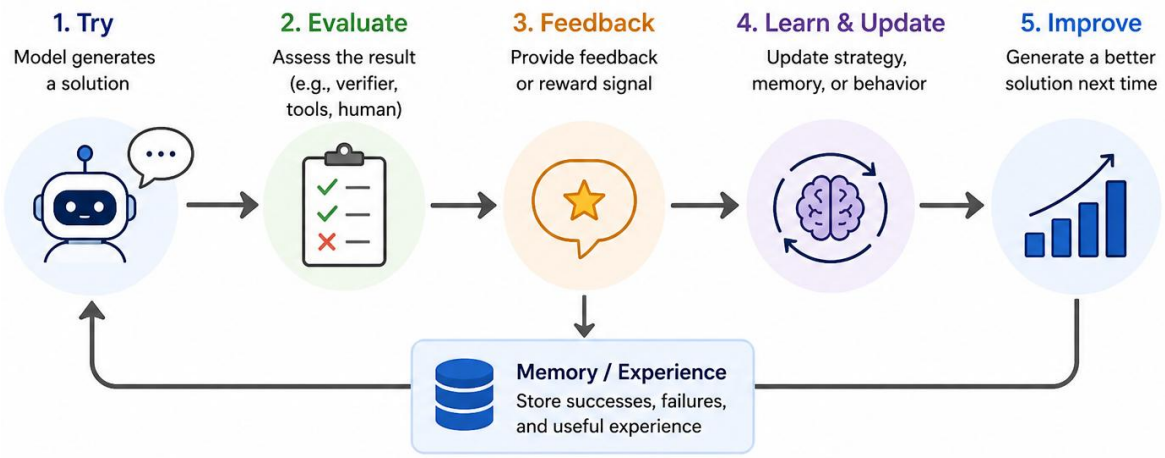


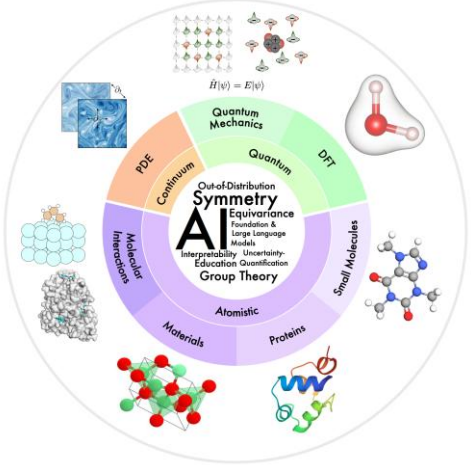
Apply FMs to scientific discovery under real constraints.

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.

Empowering biomedical discovery with AI agents. *Cell*, 2024.
Artificial intelligence for science in quantum, atomistic, and continuum systems. In *Foundations and Trends in Machine Learning*, 2025.

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.

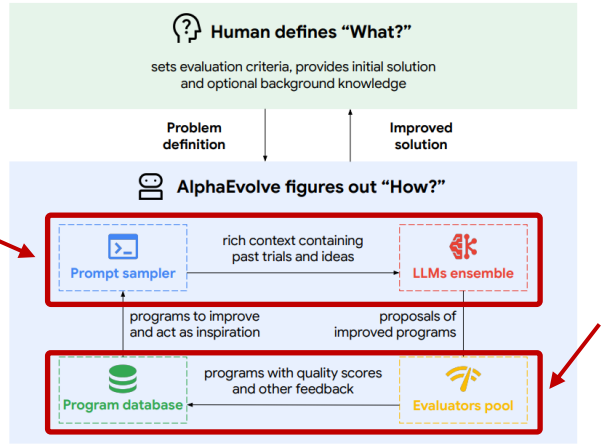
Feedback-driven evolution becomes explicit and is validated in AI4Science.

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.

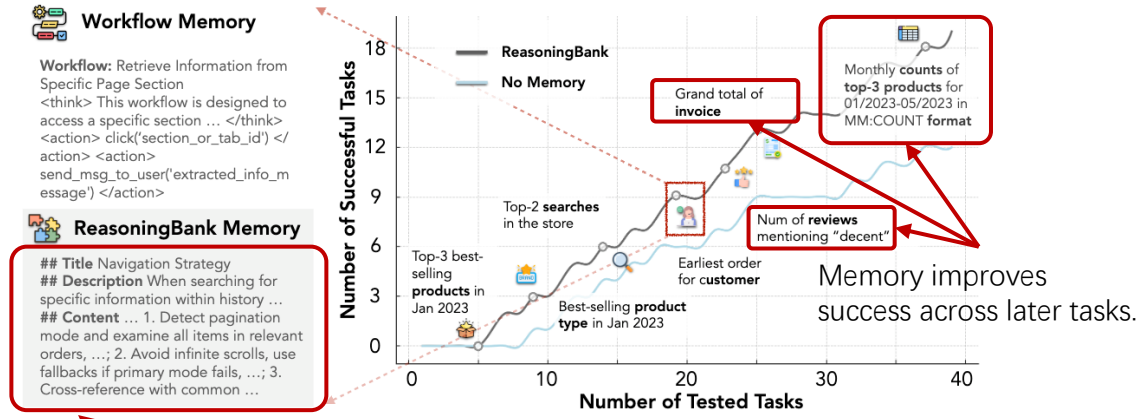
Search generates stronger program variants.



Evaluation feedback selects what to keep.

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**.



Memory improves success across later tasks.

Past trajectories become reusable memory.

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.

AlphaEvolve: A coding agent for scientific and algorithmic discovery. *ArXiv* preprint, 2025.
ReasoningBank: Scaling Agent Self-Evolving with Reasoning Memory. In *ICLR*, 2026.

https://bhanml.github.io/ & https://github.com/tmlr-group

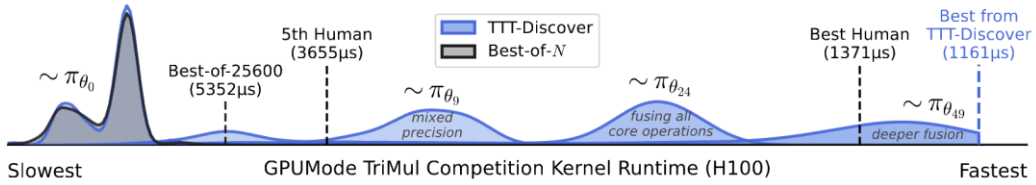
Learning-based Methods for Self-Evolution

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

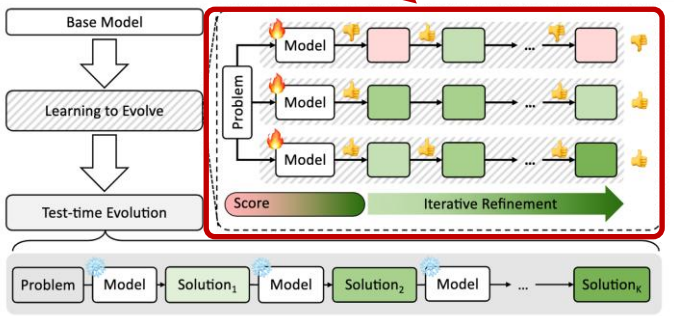
- Self-evolution turns feedback into improvement through **test-time training** or **multi-turn iterative refinement**.
- Improvement is trustworthy only when **feedback is verifiable**, **progress is measurable**, and drift is constrained.

Sets new SoTAs in almost all attempted discovery tasks.

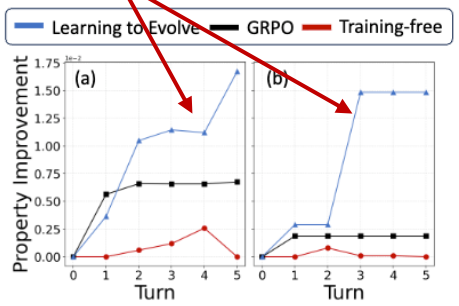
	Mathematics Erdős' Min. Overlap (↓)	Kernel Eng. (TriMul) A100 (↓)	Kernel Eng. (TriMul) H100 (↓)	Algorithms (AtCoder) Heuristic Contest 39 (↑)	Biology Denoising (↑)
Best Human	0.380927 [20]	4531 µs	1371 µs	566,997 [56]	0.64
Prev. Best AI	0.380924 [50]	N/A	N/A	558,026 [37]	N/A
TTT-Discover	0.380876	2198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



Evaluator feedback is assigned across refinement turns.



Learning improves over turns beyond training-free search.



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.

Learning to Evolve (LtE)

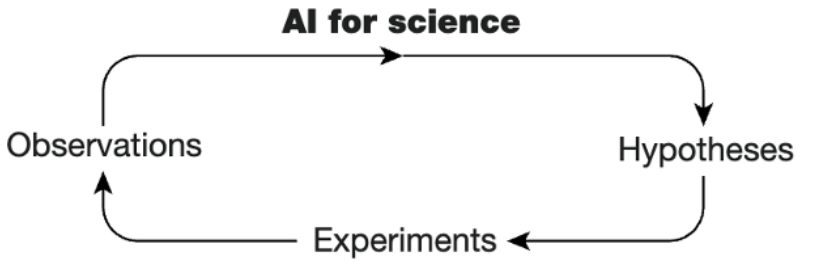
- **Problem:** Outcome-level feedback is too coarse for multi-turn self-evolution.
- **Method:** Assign evaluator feedback across refinement trajectories to learn a stronger refinement policy.
- **Result:** Improve iterative refinement more effectively than GRPO and training-free baselines.

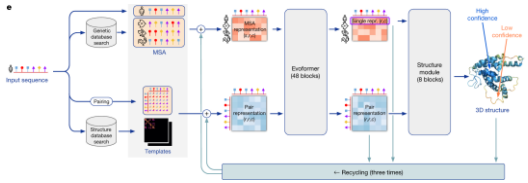
Learning to Discover at Test Time. *ArXiv* preprint, 2026.
Learning to Evolve: Scaling Open-Ended Discovery with Relative-Progress RL. In *LLA workshop of ICLR*, 2026.

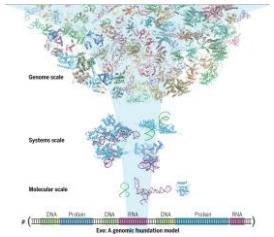
https://bhanml.github.io/ & https://github.com/tmlr-group

Applications: Trustworthy FMs for Science

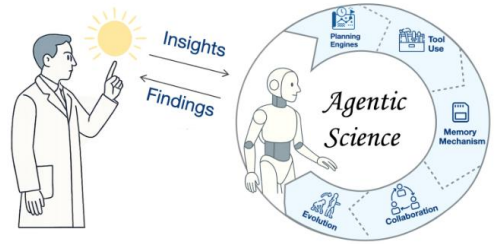
- **AI4Science** is moving from **model prediction** to **agentic discovery**, where AI systems **analyse** scientific data, **plan** actions, **use** tools or experiments, **check** feedback, and **improve** the next candidate.

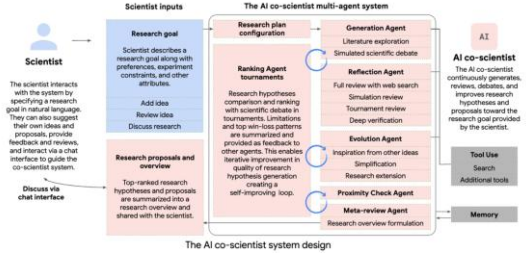


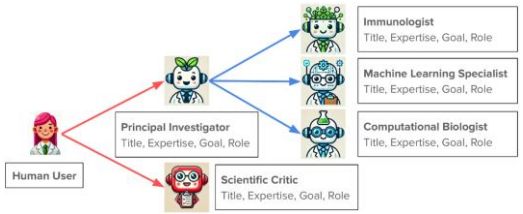




AlphaFold for protein folding. **Evo** for genome understanding.







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**

Highly accurate protein structure prediction with AlphaFold. *Nature*, 2021.
Scientific discovery in the age of artificial intelligence. *Nature*, 2023.
Towards an AI co-scientist. *ArXiv* preprint, 2025.
From AI for Science to Agentic Science: A Survey on Autonomous Scientific Discovery. *ArXiv* preprint, 2026.

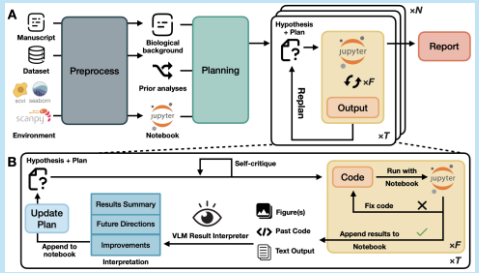
https://bhanml.github.io/ & https://github.com/tmlr-group

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.

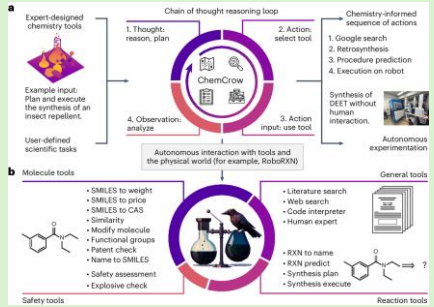
CellVoyager: AI CompBio Agent

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



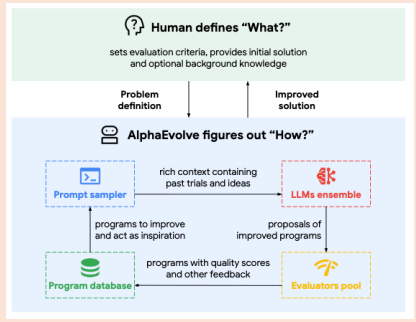
ChemCrow: Chemistry agent

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



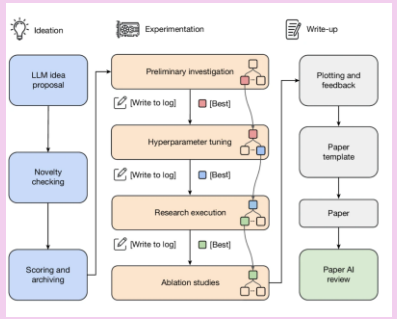
AlphaEvolve: Code-Based Scientific Discovery

- **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**.



Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 2025.
AlphaEvolve: A coding agent for scientific and algorithmic discovery. *ArXiv* preprint, 2025.
Towards end-to-end automation of AI research. *Nature*, 2026.
CellVoyager: AI CompBio agent generates new insights by autonomously analyzing biological data. *Nature Methods*, 2026.

https://bhanml.github.io/ & https://github.com/tmlr-group

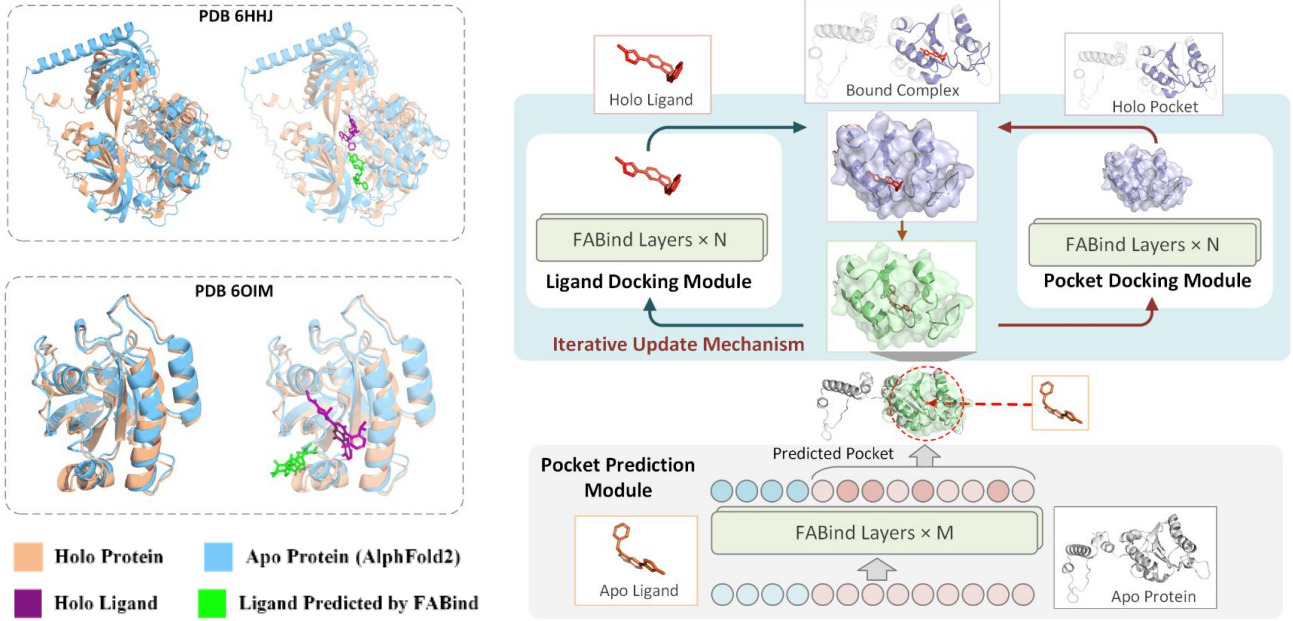
Application Case: Molecular Docking

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.

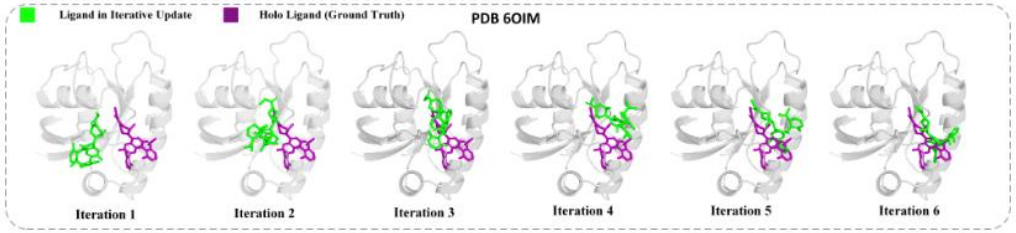


Results: FABFlex is accurate and fast.

- **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	
Traditional Docking Software													
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	7.35	5.33	24.55	50.91	146
Deep Learning-based Rigid Docking Methods													
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	1.55	3.22	6.86	5.99	36.75	62.58	1.86	3.16	9.09	7.14	29.82	**61.40**	58.72
Deep Learning-based Flexible Docking Methods													
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**	2.96	**6.16**	**5.44**	**40.59**	**68.32**	**1.81**	3.51	8.03	7.17	**32.46**	59.65	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.



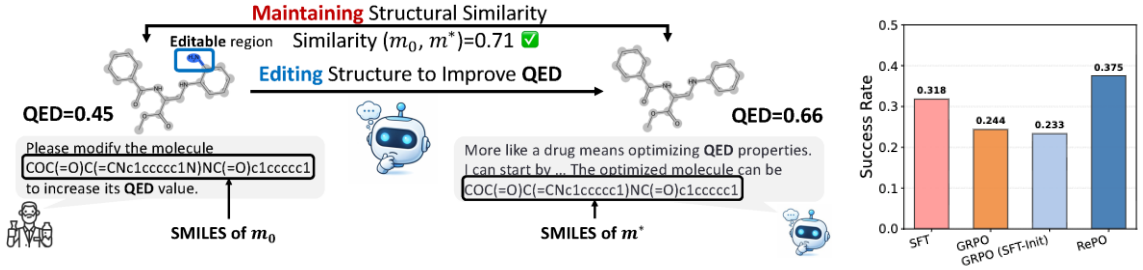
Fast and Accurate Blind Flexible Docking. In *ICLR*, 2025.

https://bhanml.github.io/ & https://github.com/tmlr-group

Application Case: Molecular Optimization

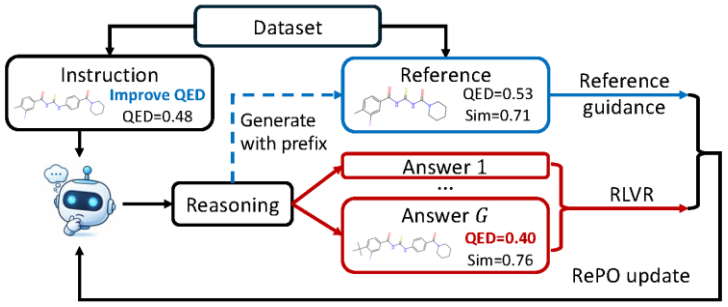
Problem: Final-answer-only supervision collapses the reasoning ability.

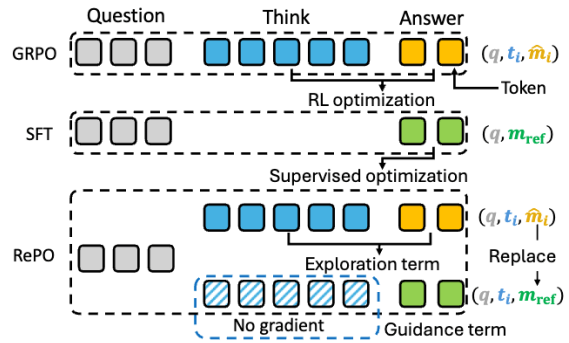
- SFT **only generates answers** without reasoning trajectories.
- RLVR rewards are **sparse and** difficult to provide optimization signals.
- SFT-initialized RL **inherits weak exploration**.



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.

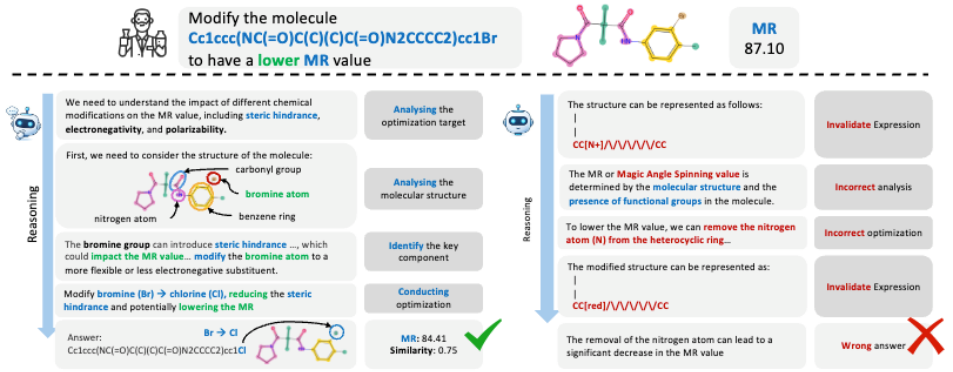




Results: RePO enhances the model in optimizing properties.

- **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
		SR×Sim	0.066	0.060	0.147	0.005	0.156	**0.239**
	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
		SR×Sim	0.092	0.128	0.153	0.008	**0.176**	0.140
	SubComponent	SR	0.057	0.078	0.366	0.053	0.420	**0.429**
		Sim	0.815	0.633	0.721	**0.972**	0.713	0.802
		SR×Sim	0.046	0.049	0.264	0.052	0.299	**0.344**
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**



Reference-guided Policy Optimization for Molecular Optimization via LLM Reasoning. In *ICLR*, 2026.

https://bhanml.github.io/ & https://github.com/tmlr-group

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.

Trustworthy Foundation Models

Benchmarking

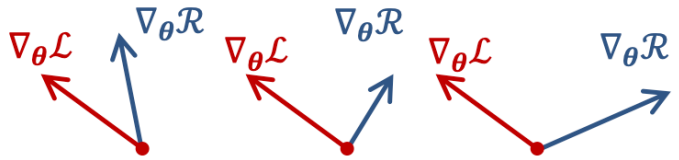
Existing datasets are NOT proper to assess if **VLMs** are robust.

CounterAnimal, a reliable benchmark for assessing VLMs.

- **Scaling backbone models** and **improving data quality** improve the robustness of VLMs.
- **Scaling raw training data** does not necessary enhance reliability.

Finetuning

Analyzing the dynamics of **LLMs** unlearning is critical yet hard.



- **Analyzing gradients** provides insights into unlearning dynamics.
- **Wrong token reweighting** within gradients leads to failures in previous methods.

Reasoning

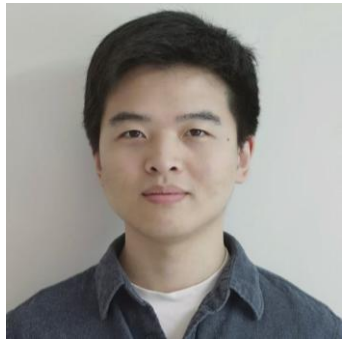
Existing **LLMs** are passive chatbots rather than active reasoners.

AR-Bench, a benchmark to evaluate LLMs' ability to ask the right questions.

- Existing LLMs and methods exhibit **a significant gap** compared to humans in active reasoning.
- LLMs struggle to consistently ask **high-quality questions**.

Part I: Benchmarking

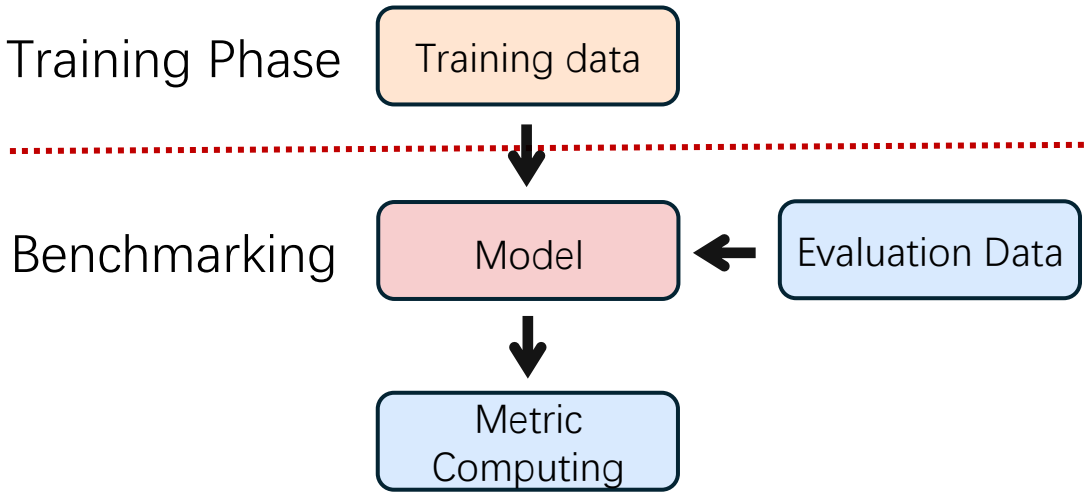
Qizhou Wang



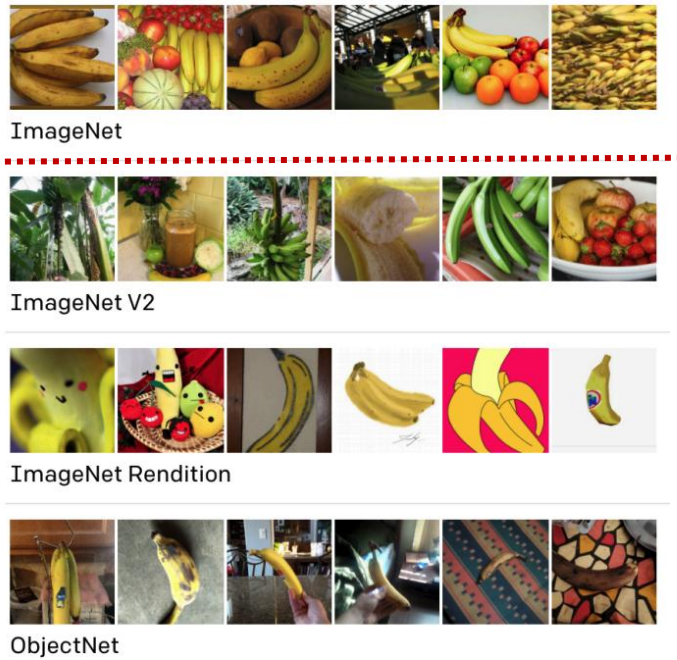
Yongqiang Chen

Benchmarking is critical to evaluate and compare model quality.

- Gathering **reliable evaluation data.**
- Conducting **proper metric evaluations.**



Training and evaluation data have **distribution shifts** to reflect **OOD Generalization**.

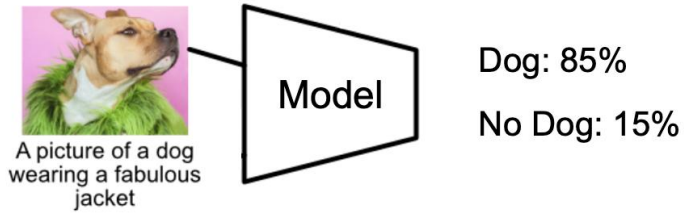


Qizhou Wang, Yong Lin, **Yongqiang Chen**, Ludwig Schmidt, **Bo Han**, and Tong Zhang. A Sober Look at the Robustness of CLIPs to Spurious Features. In *NeurIPS*, 2024.

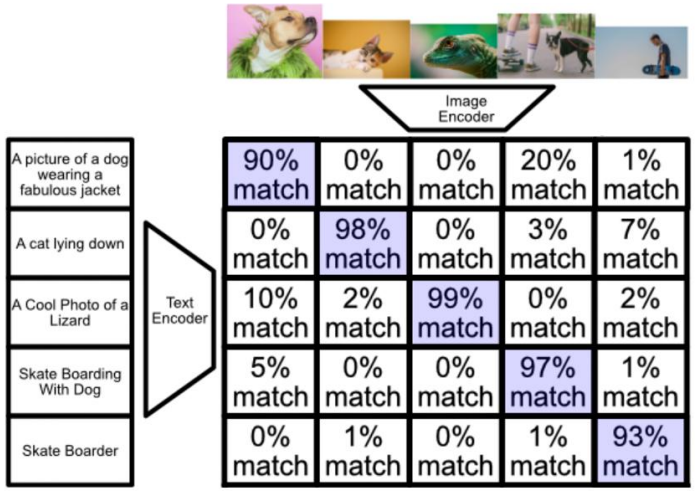
https://bhanml.github.io & https://github.com/tmlr-group

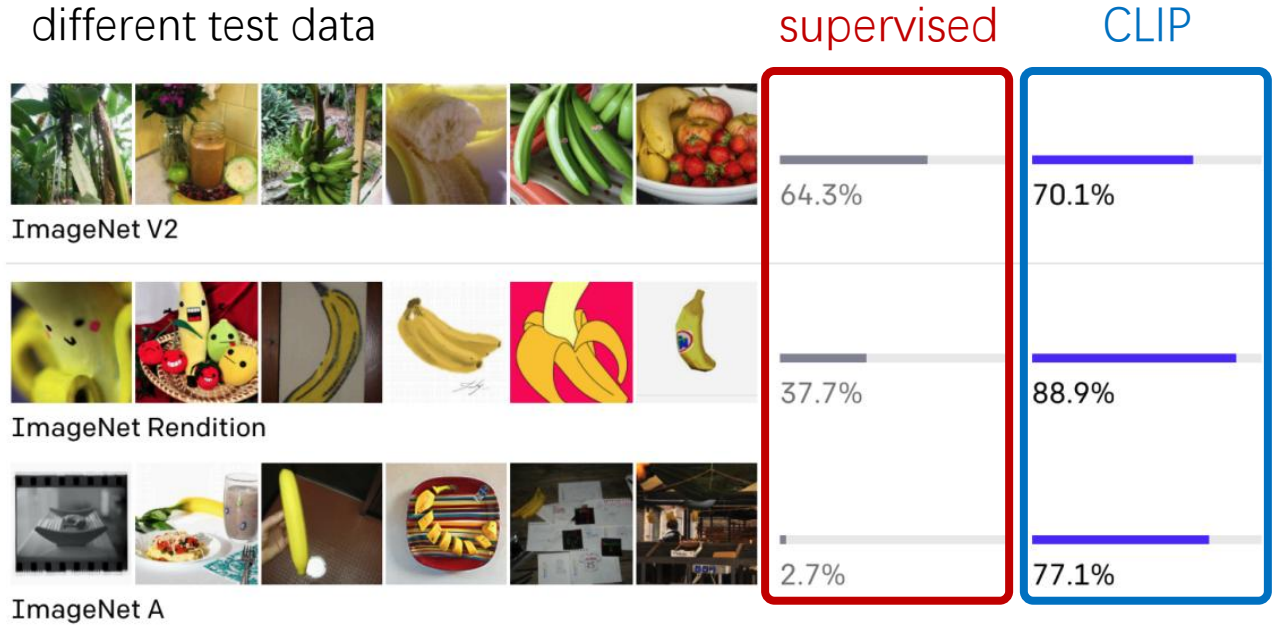
Supervised vs CLIP Training

Supervised Training *label supervision*



CLIP Training *cross-modal supervision*





*Comparison of the **OOD evaluation accuracy** between supervised and CLIP training shows that **CLIP performs better!***

Previous Belief: CLIP is **more robust to distribution shifts** than conventional supervised training.

(Radford et al., 2021)

Is the Conclusion Correct?

These OOD datasets are crafted for the distribution shifts **within ImageNet setups**, which are **NOT valid for CLIP models.**

- **Data Contamination:** Datasets considered **OOD for ImageNet-trained models** may be **ID for CLIP models.**

ImageNet V2

CLIP models may have seen ImageNet V2 during training, which is in fact ID for CLIP setups.

- **Biased Spuriousness:** Features that **mislead ImageNet-trained models** may **not mislead CLIP models** necessarily.

ImageNet A

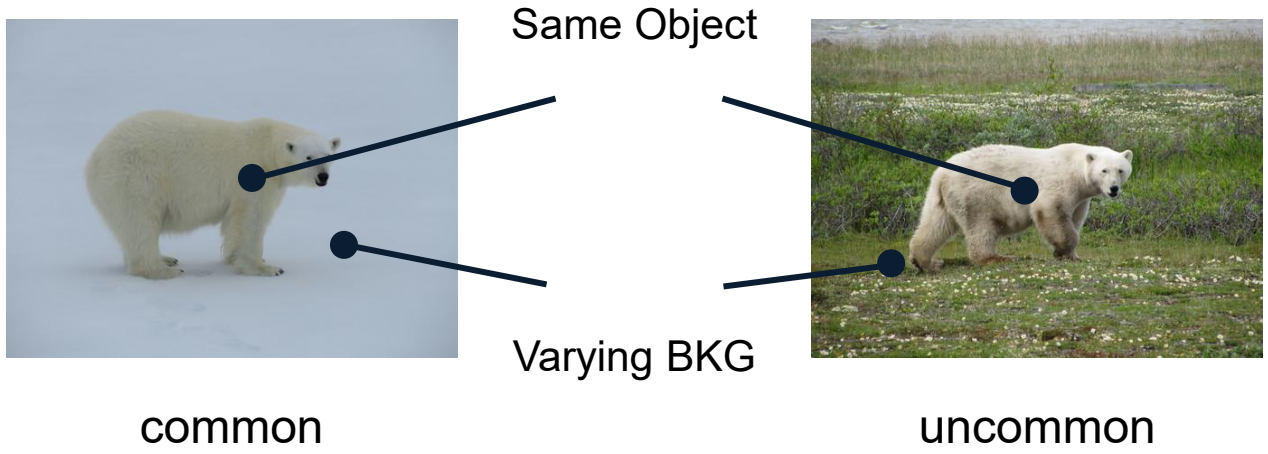
ImageNet A contains data that mislead ImageNet models, which may not make CLIP models fail.

ImageNet OOD datasets CANNOT reflect the OOD Generalization for CLIP setups!

CounterAnimal: A New Benchmark

Is there a benchmark capturing **true OOD performance** of CLIP?

- **Spuriousness:** Considering **background changes** as potential spurious features.
- **Generality:** The captured spurious features should impact **diverse CLIP configurations**.



The changes of backgrounds represent the impacts of spurious features, which is a typical distribution shift.

Basic Assumption: Since "ice bears" are more commonly appear with "ice" rather than "grass" backgrounds, CLIP may rely on ice-related spurious features.

CounterAnimal Construction

Step 1. Data Collection

Raw data from iNaturalist (https://www.inaturalist.org)

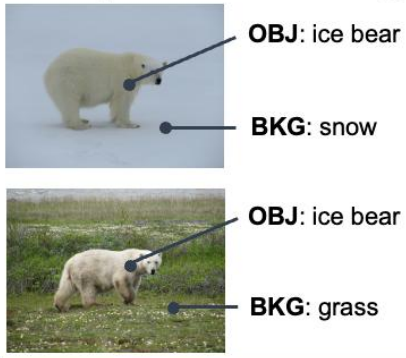
query and crawl **animal photos** given the names

Step 2. Data Curation

Raw data are susceptible to **noise** and **ambiguities**, which should be **cleansed manually**.

clean | noise | occlusion | obscurity

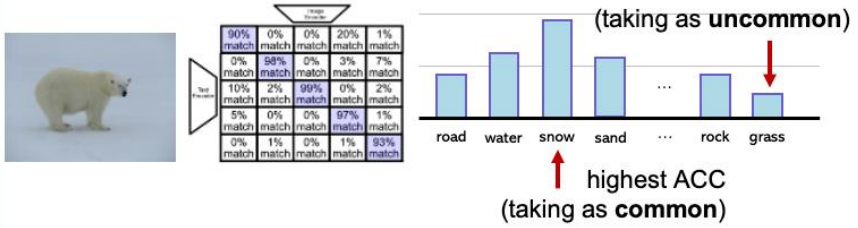
Step 3. Data Labelling



OBJ labels: *ostrich, African crocodile, water snake, ice bear*, and other totally **45 animal names**.

BKG labels: *ground, water, earth*, and other totally **16 background labels**.

Step 4. Spurious Discovery



The pair of backgrounds where the CLIP shows high-performance drops are preserved.

CounterAnimal Construction

Step 1. Data Collection

Raw data from iNaturalist (https://www.inaturalist.org)

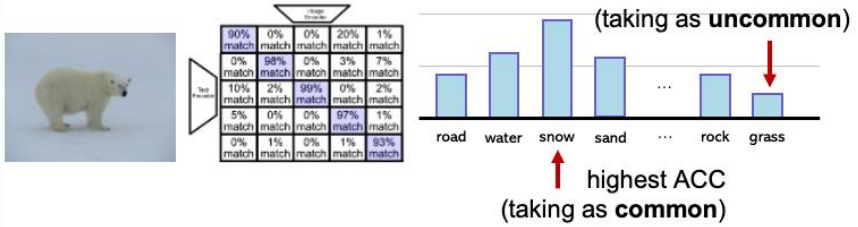
query and crawl **animal photos** given the names

Step 2. Data Curation

Raw data are susceptible to **noise** and **ambiguities**, which should be **cleansed manually**.

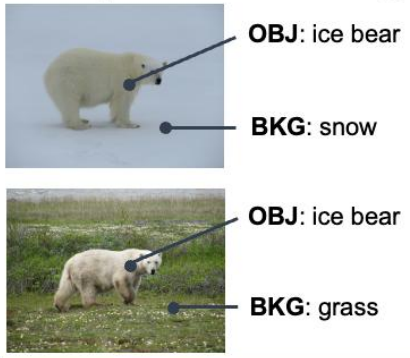
clean | noise | occlusion | obscurity

Step 4. Spurious Discovery



The pair of backgrounds where the CLIP shows high-performance drops are preserved.

Step 3. Data Labelling



OBJ labels: *ostrich, African crocodile, water snake, ice bear,* and other totally **45 animal names**.

BKG labels: *ground, water, earth*, and other totally **16 background labels**.

CounterAnimal Construction

Step 1. Data Collection

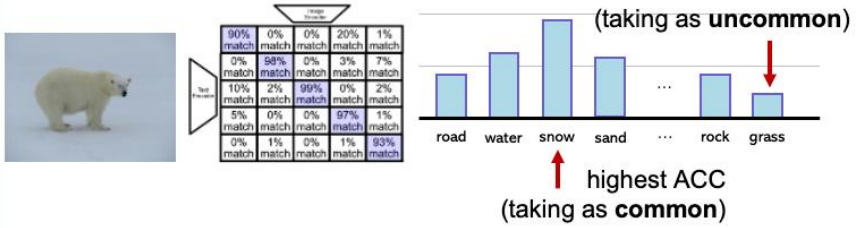
Raw data from iNaturalist (https://www.inaturalist.org)

Step 2. Data Curation

Raw data are susceptible to **noise** and **ambiguities**, which should be **cleansed manually**.

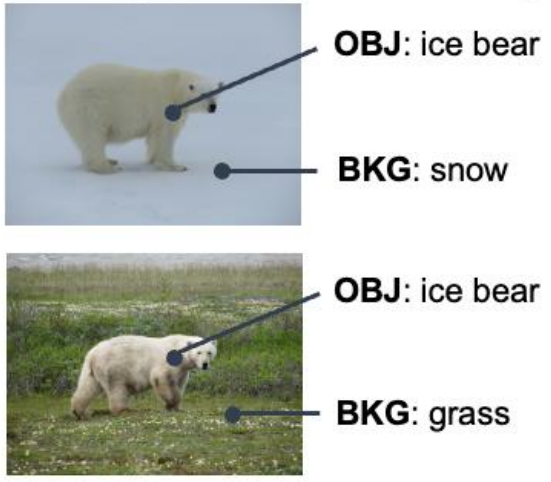
clean | noise | occlusion | obscurity

Step 4. Spurious Discovery



The pair of backgrounds where the CLIP shows high-performance drops are preserved.

Step 3. Data Labelling



OBJ labels: *ostrich, African crocodile, water snake, ice bear*, and other totally **45 animal names**.

BKG labels: *ground, water, earth*, and other totally **16 background labels**.

CounterAnimal Construction

Step 1. Data Collection

Raw data from iNaturalist (https://www.inaturalist.org)

Step 2. Data Curation

Raw data are susceptible to **noise** and **ambiguities**, which should be **cleansed manually**.

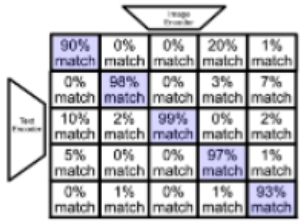
clean noise occlusion obscurity

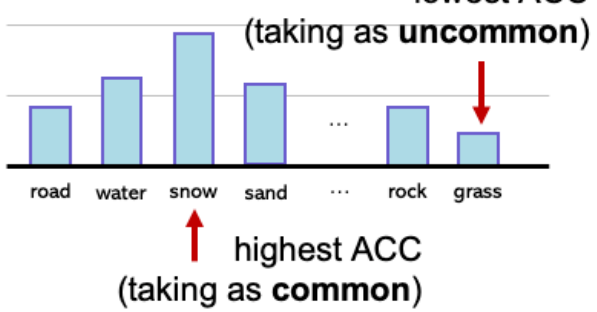
Step 3. Data Labelling

OBJ labels: *ostrich, African crocodile, water snake, ice bear,* and other totally **45 animal names**.

BKG labels: *ground, water, earth*, and other totally **16 background labels**.

Step 4. Spurious Discovery





The pair of backgrounds where the CLIP shows high-performance drops are preserved.

CounterAnimal Characteristics

```
CounterAnimal
├── ice bear
│   ├── common-ice
│   │   ├── figure1.jpeg
│   │   ├── figure2.jpeg
│   │   └── ...
│   └── uncommon-grass
│       ├── figure1.jpeg
│       ├── figure2.jpeg
│       └── ...
├── brambling
│   ├── common-green
│   │   ├── figure1.jpeg
│   │   ├── figure2.jpeg
│   │   └── ...
│   └── counter-sky
│       ├── figure1.jpeg
│       ├── figure2.jpeg
│       └── ...
```

Data Structure. *Images are organized per class and each further divided into two groups: common and uncommon.*

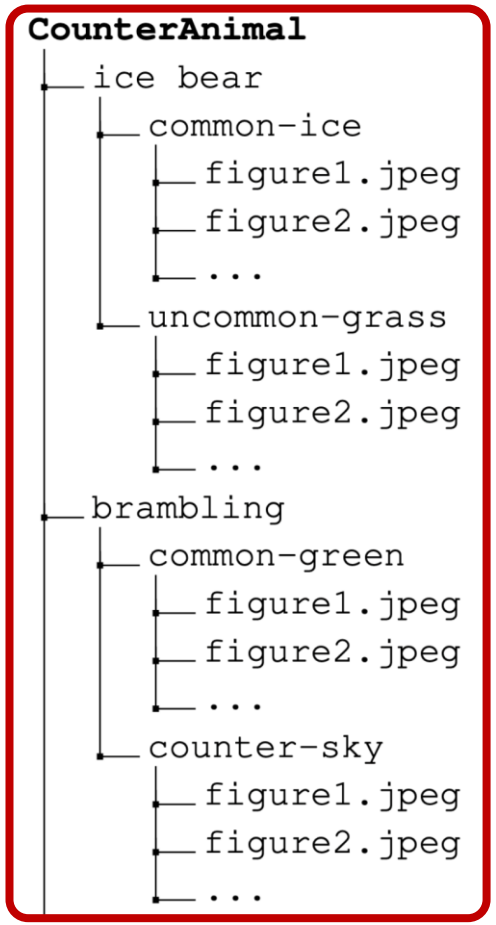
*Photos of **ice bear** in **snow** background*

*Photos of **ice bear** in **grass** background*

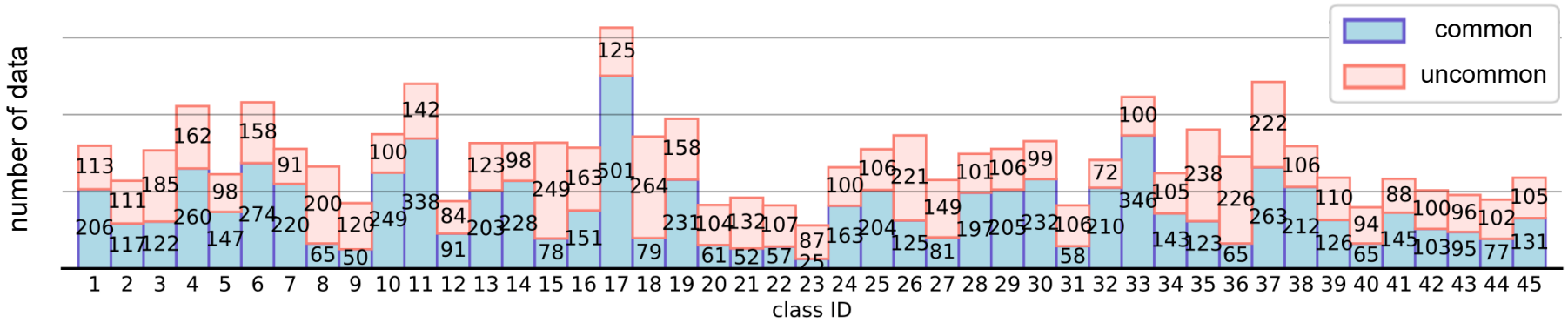
Common vs. Uncommon: Photos are grouped according to their backgrounds. For each class, we identify **group pairs** that cause **high performance drop** when evaluating with CLIP.

Assessing Robustness: The **performance drop** between common and uncommon groups indicates the robustness of evaluated models.

CounterAnimal Characteristics



Data Structure. *Images are organized per class and each further divided into two groups: common and uncommon.*



*The **data distributions** illustrate variations across different animal classes, categorized into **common** and **uncommon** groups. The horizontal axis denotes the **class IDs**, e.g., ID 1 to "ostrich", ID 2: to "brambling", ···, ID 8 to "box turtle", ID 9 to "common iguana",···, ID 18 to "scorpion", ID 19 to "tarantula", ···, ID 32 to "African hunting dog", ID 33 to "hyena", ···.*

We collect **45 classes** of animals with **7,000 common** and **6,000 uncommon** examples.

Experimental Results

common acc – uncommon acc

CLIP Training | CounterAnimal | (ImageNet) Supervised Training | Other LVLMs (large VLMs)

backbone	pre-train dataset	common	uncommon	drop
RN-101	OpenAI	64.27	45.15	19.12
RN-50×4	OpenAI	70.02	49.07	20.95
ViT-B/16	LAION400M	73.11	52.17	20.94
ViT-B/16	OpenAI	73.08	56.56	16.52
ViT-B/16	DataComp1B*	80.36	64.24	16.12
ViT-B/16	LAION2B	73.18	53.18	20.00
ViT-B/16	DFN2B*	85.03	70.61	14.42
ViT-B/32	LAION400M	67.13	36.95	30.18
ViT-B/32	OpenAI	69.13	45.62	23.51
ViT-B/32	DataComp1B*	75.96	53.74	22.22
ViT-B/32	LAION2B	72.94	48.74	24.20
ViT-L/14	LAION400M	80.90	63.31	17.59
ViT-L/14	OpenAI	85.38	70.28	15.10
ViT-L/14	DataComp1B*	89.29	79.90	9.39
ViT-L/14	LAION2B	82.23	66.27	15.96
ViT-L/14	DFN2B*	90.77	80.55	10.22
ViT-L/14-336	OpenAI	86.36	73.14	13.21
ViT-H/14	LAION2B	85.74	73.13	12.61
ViT-H/14	DFN5B*	88.55	79.13	9.42
ViT-G/14	LAION2B	86.81	73.32	13.49
ViT-bigG/14	LAION2B	87.57	76.96	10.61

increasing model scale

diverse data source

backbone	common	uncommon	drop
AlexNet	59.56	39.24	20.31
VGG-11	73.37	56.12	17.25
VGG-13	75.33	58.43	16.90
VGG-19	77.84	61.74	16.10
RN-18	74.36	56.07	18.29
RN-34	78.31	61.01	17.30
RN-50	81.44	66.07	15.37
RN-101	81.76	68.18	13.57
ViT-B/16	84.97	74.98	9.99
ViT-B/32	79.84	64.36	15.48
ViT-L/16	83.74	72.69	11.05
ViT-L/32	81.23	67.54	13.69
ConvNext-S	88.27	79.97	8.30
ConvNext-B	88.60	80.53	8.07
ConvNext-L	89.12	81.47	7.65

increasing model scale

LVLMs	common	uncommon	drop
MiniGPT4-Viccuna7B	47.99	39.73	8.26
LLaVA1.5-7B	40.06	30.09	9.97
CLIP-LAION400M-ViT-L/14	80.90	63.31	17.59
CLIP-OpenAI-ViT-L/14	85.38	70.28	15.10
CLIP-DataComp1B-ViT-L/14	89.29	79.90	9.39
CLIP-LAION2B-ViT-L/14	82.23	66.27	15.96
CLIP-DFN2B-ViT-L/14	90.77	80.55	10.22

different LVLM paradigms

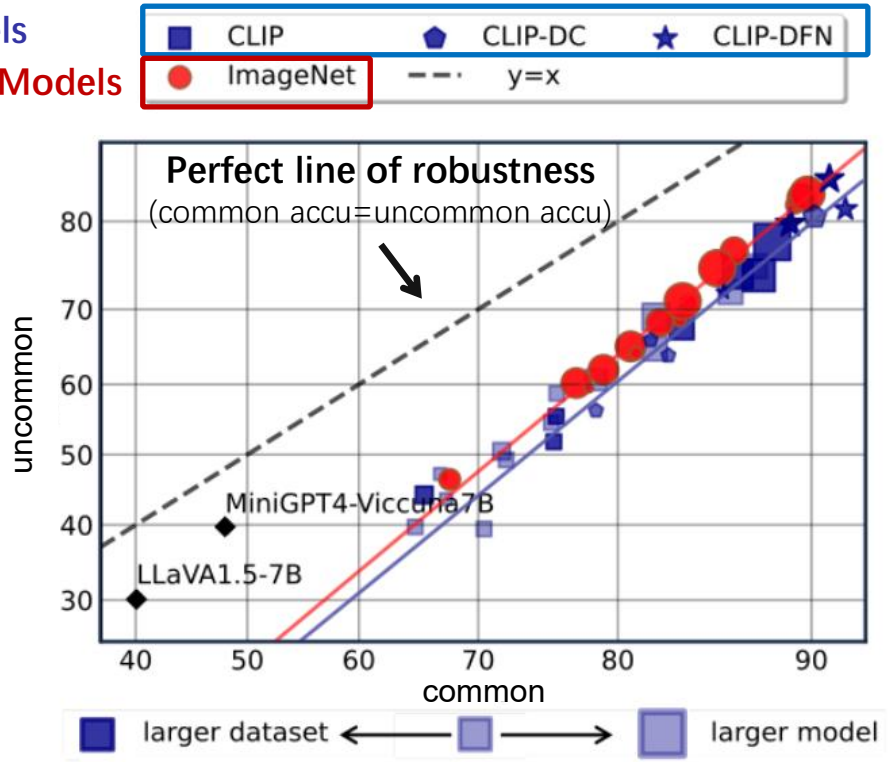
What observations can we draw from these results?

Observations

DataComp (DC) and Data Filtering Networks (DFN) are two **high-quality CLIP data sources.**

CLIP Models

ImageNet Models



*The **marker size** indicates the **backbone scale**, and the **color shade** indicates **pre-train data scale**.*

Observation 1 (ImageNet Models vs. CLIPs).

ImageNet models perform better than CLIPs against spuriousness within CounterAnimal.

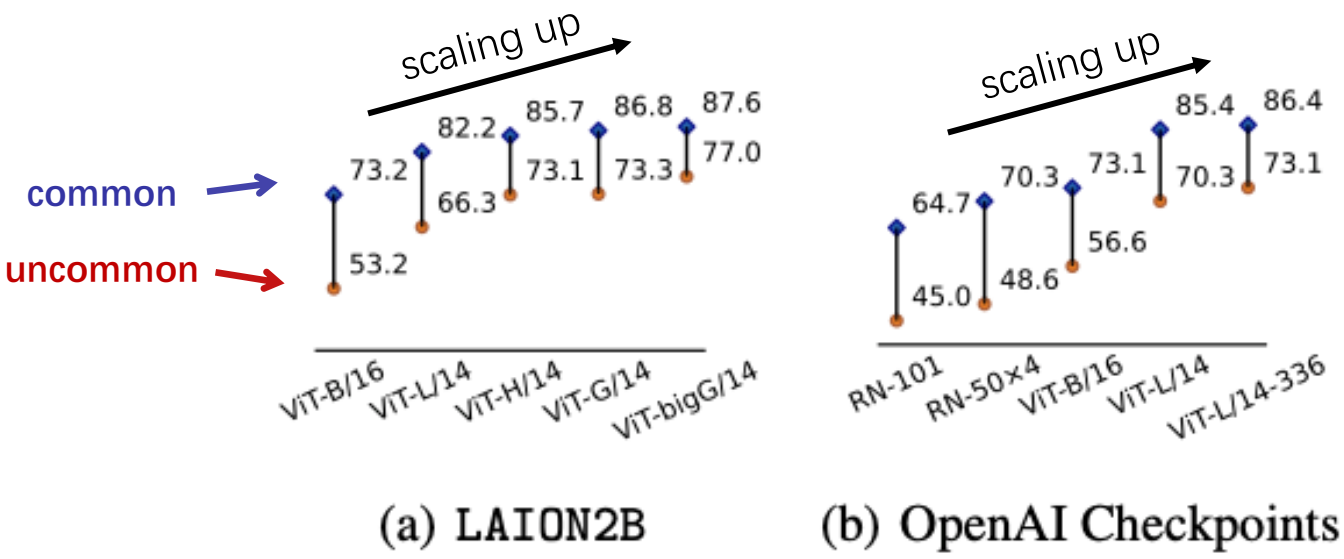
Note. CounterAnimal characterizes the spuriousness within CLIPs, thus proper for assessing CLIPs.

Observation 2 (CLIPs vs. More Advanced LVLMs).

LLaVA and MinGPT4 show **stronger robustness** (closer to $y = x$) yet with **lower performance** than CLIPs.

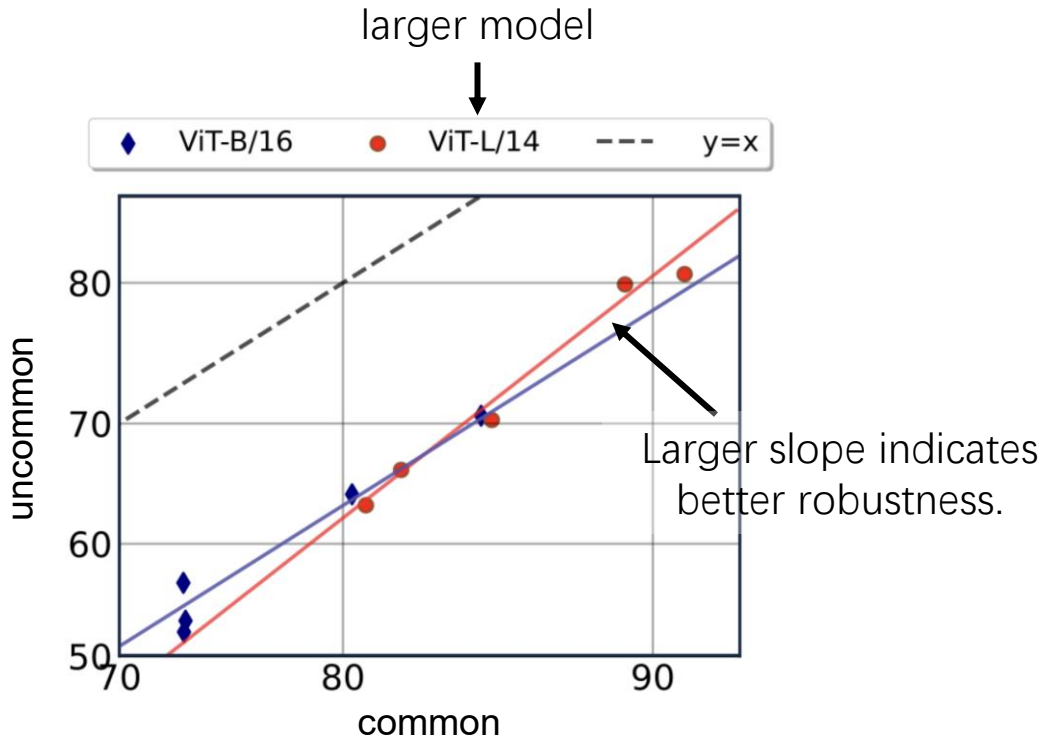
Note. More advanced VLMs built upon CLIPs are still affected by spuriousness within CounterAnimal.

Observations



(a) LAION2B (b) OpenAI Checkpoints

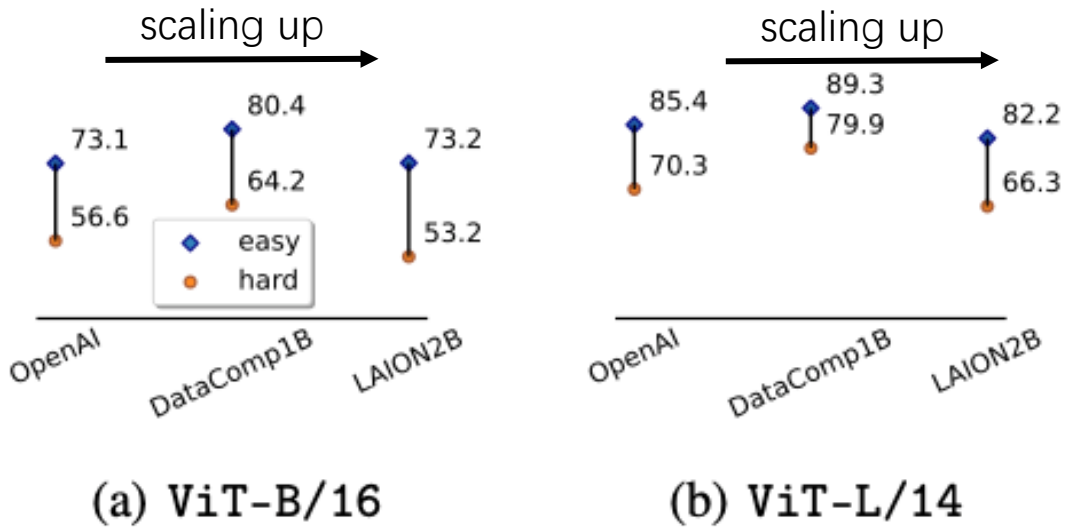
Accuracy and Performance Drop.



Effective Robustness.

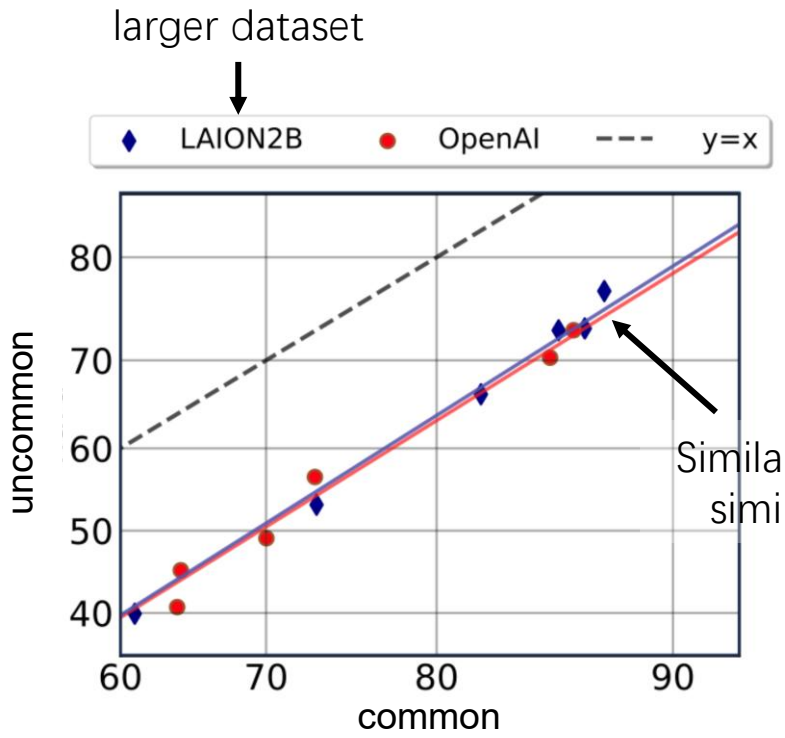
Observation 3 (Model Size). Scaling up model size CAN enhance CLIP robustness.

Observations



(a) ViT-B/16 (b) ViT-L/14

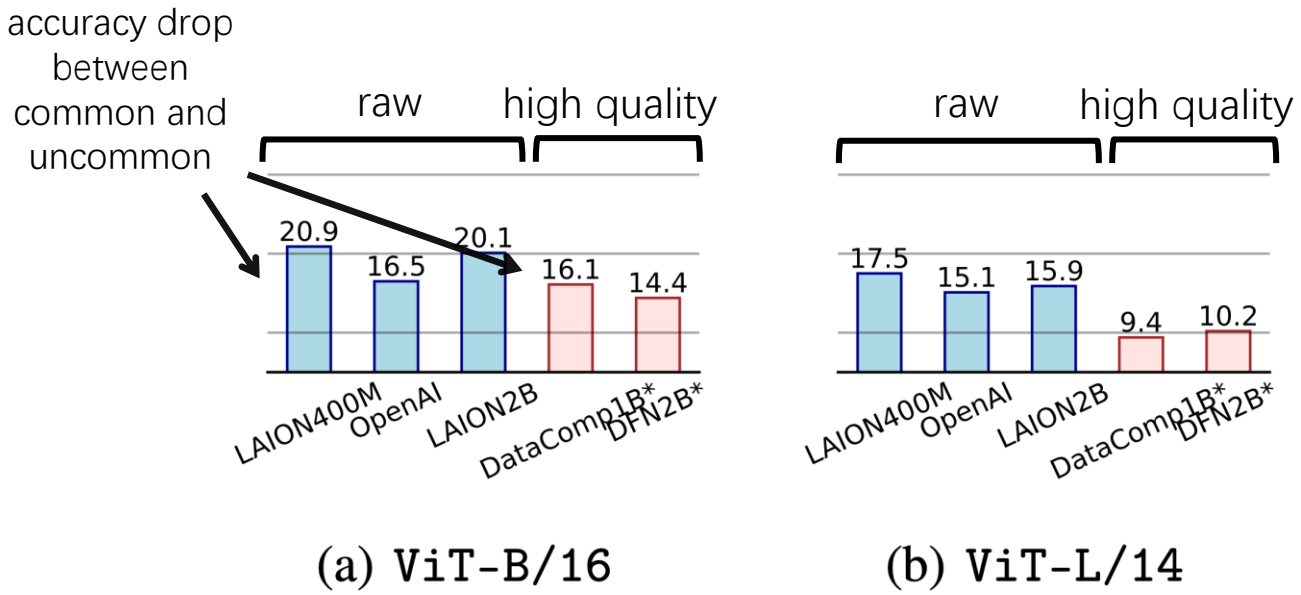
Accuracy and Performance Drop.



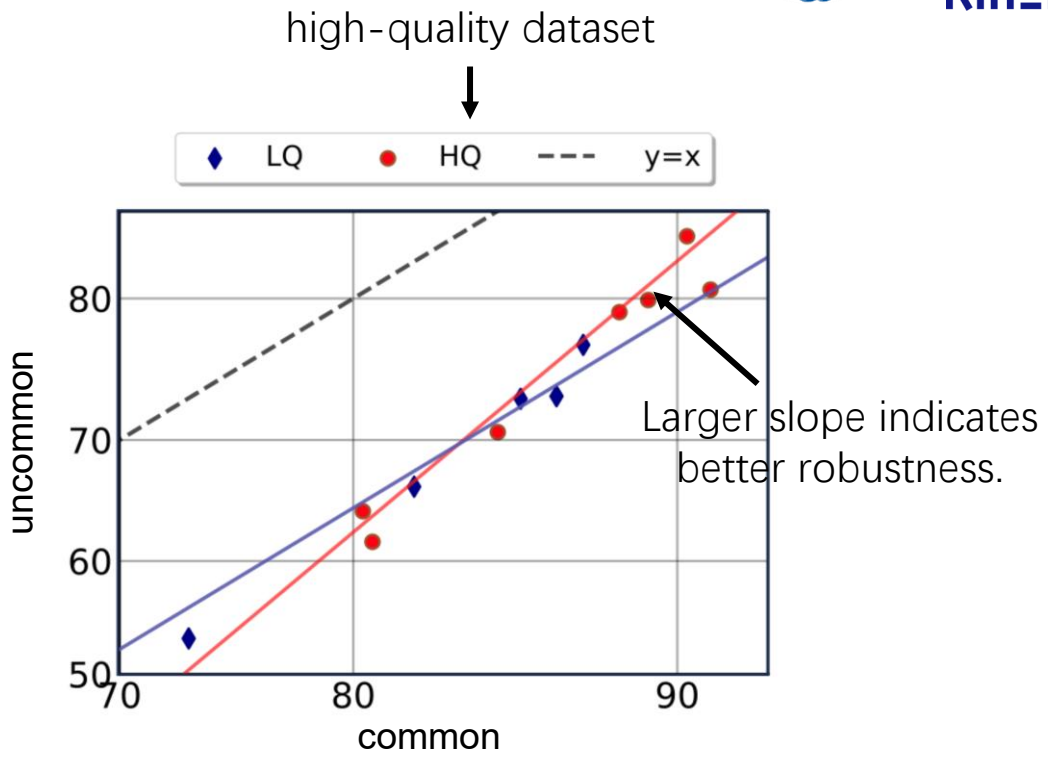
Effective Robustness.

Observation 4 (Data Size). Scaling up data size CANNOT enhance CLIP robustness.

Observations



Performance Drop.



Effective Robustness.

Observation 5 (Data Quality). Improving data quality CAN enhance CLIP robustness.

Theoretical Understanding

Assumption (Multi-modal Dataset). Considering n image-text pairs $\{(\boldsymbol{x}_{\mathrm{I}}^i, \boldsymbol{x}_{\mathrm{T}}^i)\}_{i=1}^n$, both $\boldsymbol{x}_{\mathrm{I}}^i$ and $\boldsymbol{x}_{\mathrm{T}}^i$ are generated from the latent factor $\boldsymbol{z}_i$, where $\boldsymbol{z} = [z_{inv}, z_{spu}] \in \mathbb{R}^2$ is composed of

- **invariant feature** $z_{inv} \sim \mathcal{N}(\mu_{inv} y, \sigma_{inv}^2)$
- **spurious feature** $z_{spu} \sim \mathcal{N}(\mu_{spu} a, \sigma_{spu}^2)$

with $\Pr(a = y) = p_{spr}$ otherwise $a = -y$. y is the label uniformly drawn from $\{-1,1\}$. The training data $\mathcal{D}^{tr}$ is drawn with $\frac{1}{2} \leq p_{spr} \leq 1$ and test data $\mathcal{D}^*$ is drawn with $p_{spu} = \frac{1}{2}$.

Note. The dataset is **biased** to spurious feature z_{spu} due to **different** p_{spr} between training and test.

Theorem 1. Given the multi-modal dataset with a large spurious correlation $p_{spu} = 1 - o(1)$. Then, under reasonable assumptions, w.p. at least $1 - O(1)$, the CLIP model achieves

- a **small zero-shot error** on test data where $a = y$: $\mathrm{Acc}(g_{\mathrm{I}}, g_{\mathrm{T}}) \geq 1 - \Phi(\kappa_2) - o(1)$,
- a **large zero-shot error** on test data where $a \neq y$: $\mathrm{Err}(g_{\mathrm{I}}, g_{\mathrm{T}}) \geq 1 - \Phi(\kappa_1) - o(1)$.

Therein, κ_1, κ_2 are constants that depend on $\mu_{inv}, \sigma_{inv}, \mu_{inv}$, and σ_{inv}.

Note. The model relies on whether $a = y$ (whether biased) to make right predictions.

Take Home Messages

We should be cautious about **test setups** when assessing new **training setups**.

CounterAnimal (https://counteranimal.github.io/) is a proper benchmark for assessing the robustness of CLIPs to spurious features.

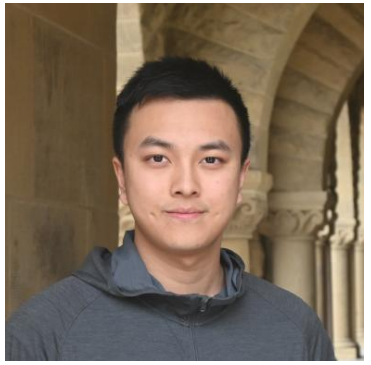
Distribution shifts remain an open question for CLIP and other VLMs.

Scaling up model size can enhance robustness, while **scaling up pre-train data** is not that effective.

Improving data quality is effective to enhance robustness.

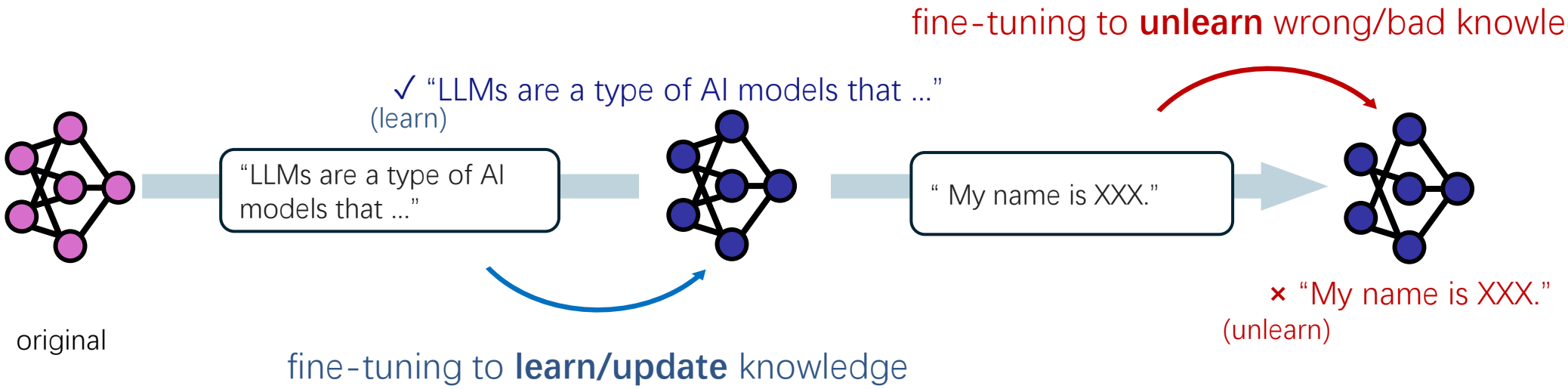
Part II: Finetuning

Qizhou Wang



Zhanke Zhou

Finetuning aims to adapts the model parameters to fit tasks or knowledge, of which the specific goals can be attributed to **learning** and **unlearning**.



Qizhou Wang, Jin Peng Zhou, **Zhanke Zhou**, Saebyeol Shin, **Bo Han**, Kilian Q. Weinberger. Rethinking LLM Unlearning Objectives: A Gradient Perspective and Go Beyond. In *ICLR*, 2025.

https://bhanml.github.io & https://github.com/tmlr-group

Right to be Forgotten

"The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data …"

"A consumer shall have the right to request that a business delete any personal information about the consumer which the business has collected from the consumer …"

LLM Unlearning

to be unlearned

Bi-objective Goal

- **Unlearn:** removing model capability to generate targeted data $\mathcal{D}_{\mathrm{u}} = \{s_{\mathrm{u}}\}_{n_{\mathrm{u}}}$
- **Retain:** maintain performance on other non-targeted data $\mathcal{D}_{\mathrm{r}} = \{s_{\mathrm{r}}\}_{n_{\mathrm{r}}}$

not to be unlearned

Gradient Ascent (GA)-based Method

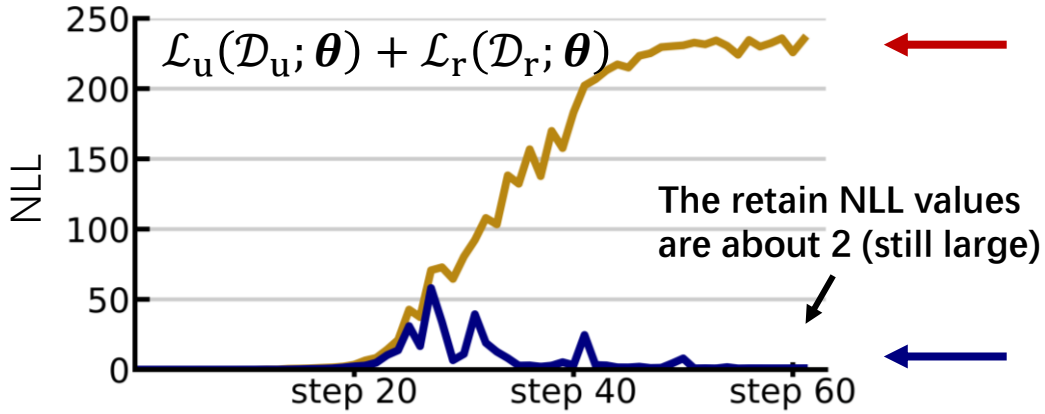
$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{\mathcal{D}_{\mathrm{u}}} \log P(s_{\mathrm{u}}; \boldsymbol{\theta})}_{\mathcal{L}_{\mathrm{u}}(\mathcal{D}_{\mathrm{u}}; \boldsymbol{\theta})} + \underbrace{\mathbb{E}_{\mathcal{D}_{\mathrm{r}}} -\log P(s_{\mathrm{r}}; \boldsymbol{\theta})}_{\mathcal{L}_{\mathrm{r}}(\mathcal{D}_{\mathrm{r}}; \boldsymbol{\theta})}$$

Unlearn Objective **Retain Objective**

Basic Assumption: If the negative log-likelihood is a proper objective for learning, then the log-likelihood should be appropriate for unlearning.

Impacts of GA

Negative log-likelihood (NLL) as the **metric** $\mathcal{R}$ to assess performance.



Performance regarding unlearning and retention.

$\mathcal{R}(\mathcal{D}_{\mathrm{u}};\boldsymbol{\theta})$: **large unlearn NLL** indicates **strong** unlearning.

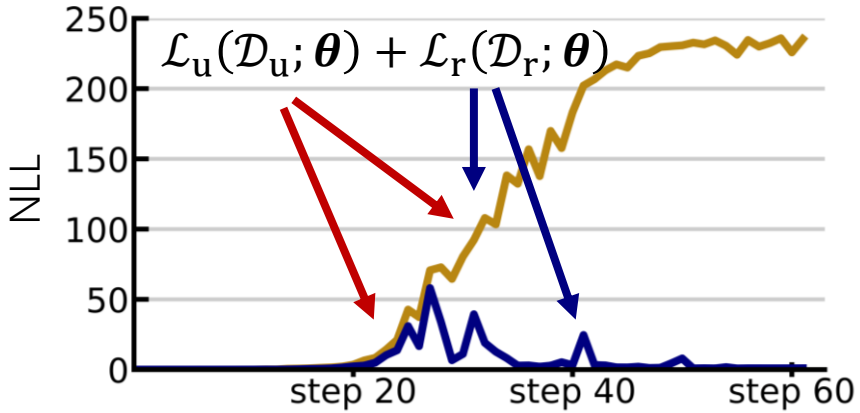
$\mathcal{R}(\mathcal{D}_{\mathrm{r}};\boldsymbol{\theta})$: **large retain NLL** indicates **damage** in retention.

Observation 1. GA-based methods CAN achieve strong unlearning but CANNOT ensure reliable retention, thus **NOT meeting the dual-objective goal.**

Delve Deeper?

Performance metrics offer **limited** insights towards deeper understandings.

Limitation 1. We CANNOT **disentangle** the impacts of $\mathcal{L}_{\mathrm{u}}(\mathcal{D}_{\mathrm{u}};\boldsymbol{\theta})$ and $\mathcal{L}_{\mathrm{r}}(\mathcal{D}_{\mathrm{r}};\boldsymbol{\theta})$ on model performance.



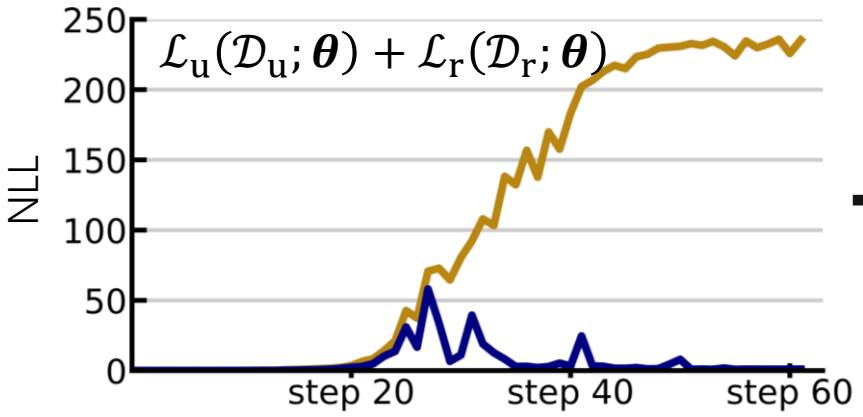
Using NLL to assess performance changes regarding unlearning and retention.

Both $\mathcal{L}_{\mathrm{u}}(\mathcal{D}_{\mathrm{u}};\boldsymbol{\theta})$ and $\mathcal{L}_{\mathrm{r}}(\mathcal{D}_{\mathrm{r}};\boldsymbol{\theta})$ have impacts on $\mathcal{R}(\mathcal{D}_{\mathrm{u}};\boldsymbol{\theta})$ and $\mathcal{R}(\mathcal{D}_{\mathrm{r}};\boldsymbol{\theta})$ in an **intertwined** manner.

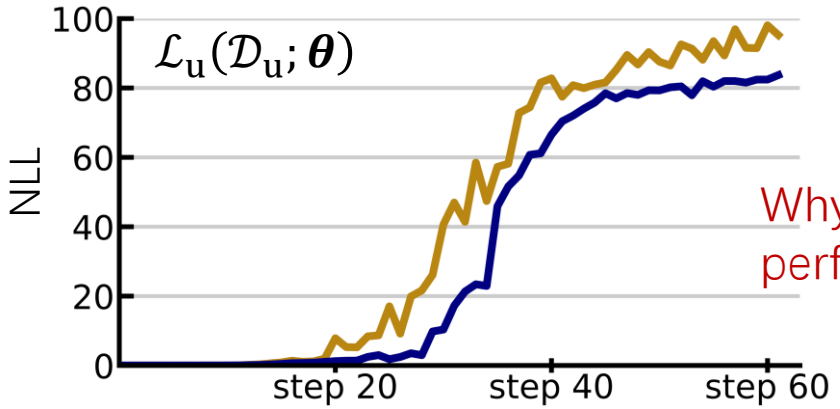
Delve Deeper?

Performance metrics offer **limited** insights towards deeper understandings.

Limitation 2. Even disentangled, we CANNOT fully **understand the factors** that lead to the observed behaviors.



Unlearning with $\mathcal{L}_u(\mathcal{D}_u;\boldsymbol{\theta}) + \mathcal{L}_r(\mathcal{D}_r;\boldsymbol{\theta})$.



For illustration, we approximate the disentanglement by unlearning only with $\mathcal{L}_u(\mathcal{D}_u;\boldsymbol{\theta})$.

Why does the retention performance drop so quick? 🤔

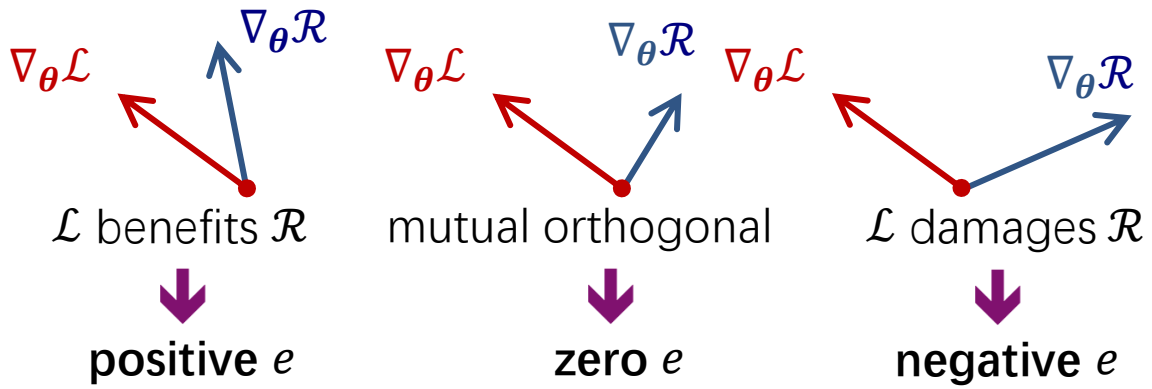
G-effect: A Gradient View

Studying the impacts of **unlearning methods** (e.g., GA) on **performance metrics** (e.g., NLL) from a gradient view.

gradients of **objective** (unlearning method)

$$e = \nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathcal{D};\boldsymbol{\theta})^{\top}\nabla_{\boldsymbol{\theta}}\mathcal{R}(\mathcal{D};\boldsymbol{\theta})$$

gradients of **metric**

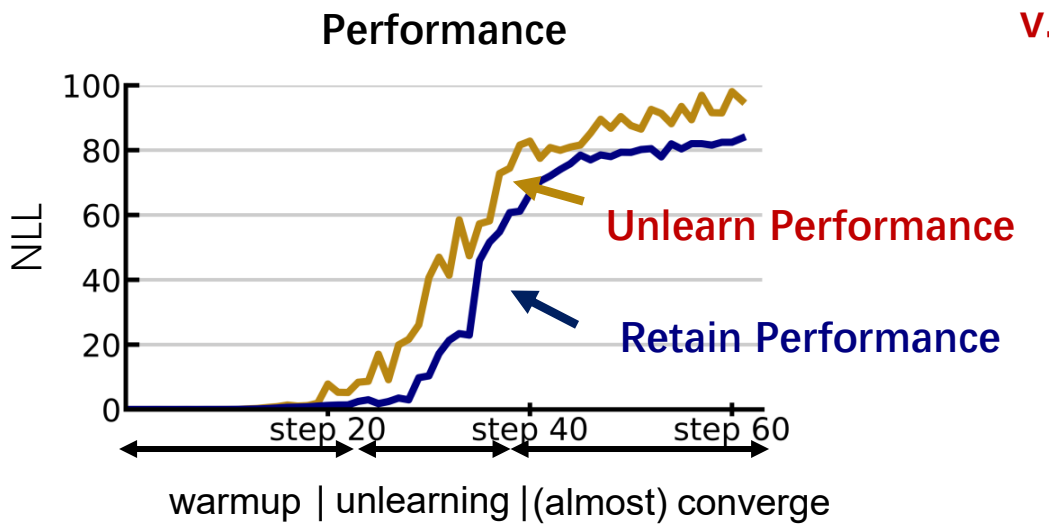


- **Fulfill Goal 1** as the G-effect can be computed for $\mathcal{L}_{\mathrm{u}}(\mathcal{D}_{\mathrm{u}};\boldsymbol{\theta})$ and $\mathcal{L}_{\mathrm{r}}(\mathcal{D}_{\mathrm{r}};\boldsymbol{\theta})$ separately.
- **Fulfill Goal 2** as gradients provide more messages than merely CE performance.

G-effect: An Example

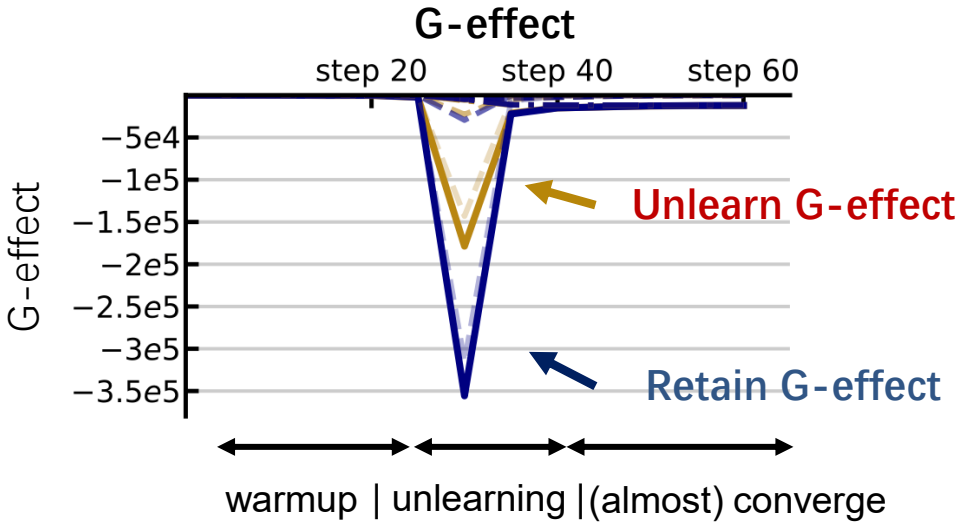
Retain G-effect: $e_{\mathrm{r}} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}_{\mathrm{u}}; \boldsymbol{\theta})^{\top} \nabla_{\boldsymbol{\theta}} \mathcal{R}(\mathcal{D}_{\mathrm{r}}; \boldsymbol{\theta})$. A **positive** e_{r} is preferred to enhance retention.

Unlearn G-effect: $e_{\mathrm{u}} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}_{\mathrm{u}}; \boldsymbol{\theta})^{\top} \nabla_{\boldsymbol{\theta}} \mathcal{R}(\mathcal{D}_{\mathrm{u}}; \boldsymbol{\theta})$. A **negative** e_{u} is preferred for strong unlearning.



Using NLL to assess performance.

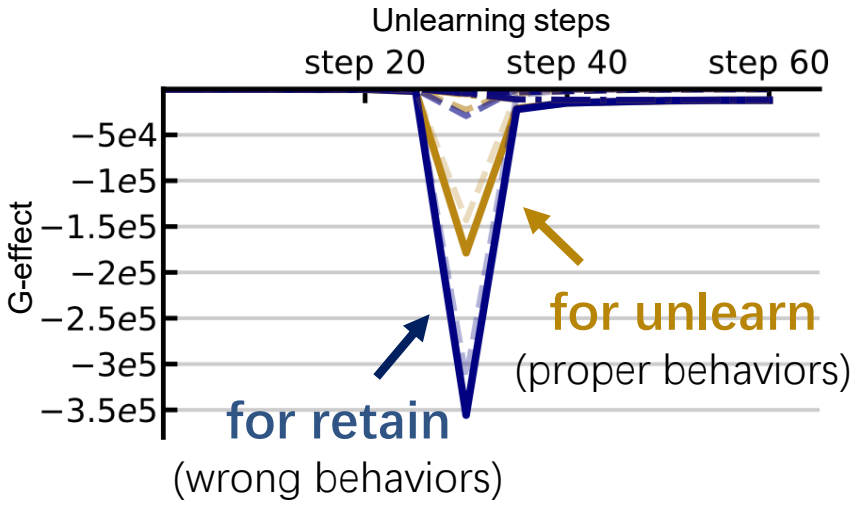
v.s.



Using G-effect to assess performance change.

Note. The G-effect quantifies the **rate of change** (increase/decrease) in performance, which can be calculated **separately** for retention and unlearning.

GA: Objective 1



The G-effects of GA.

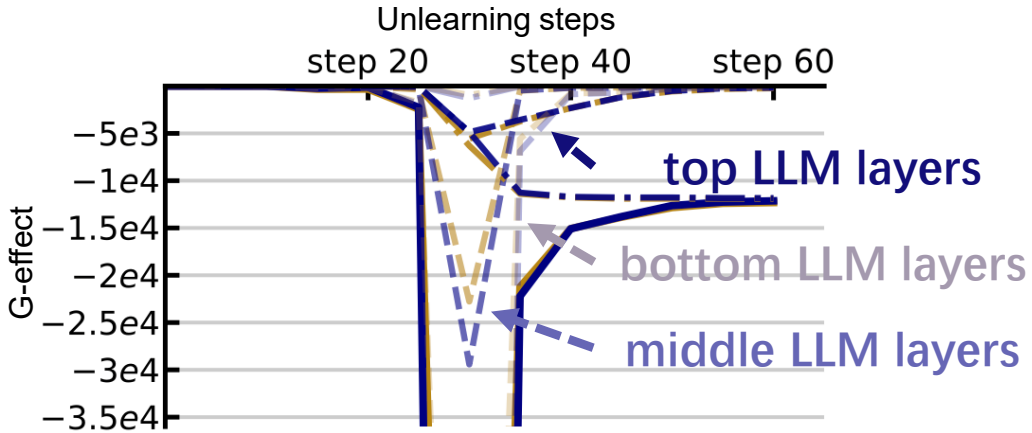
Objective: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \sum_i \log P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)$

Gradient: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \sum_i \underbrace{\frac{1}{P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)}}_{\text{inverse likelihood}} \nabla_{\boldsymbol{\theta}} P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)$

Observation 2. **Excessive extent of removal** incurs negative costs to retention.

Reason. The inverse likelihood wrongly focuses more on sufficiently unlearned tokens, leading to **over-unlearning** that negatively impacts model utility.

GA: Objective 1



The G-effects of GA (closer look).

Objective: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \sum_i \log P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)$

Gradient: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \sum_i \underbrace{\frac{1}{P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)}}_{\text{inverse likelihood}} \nabla_{\boldsymbol{\theta}} P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)$

Observation 3. Unlearning **affects on bottom layers** of LLMs more than others.

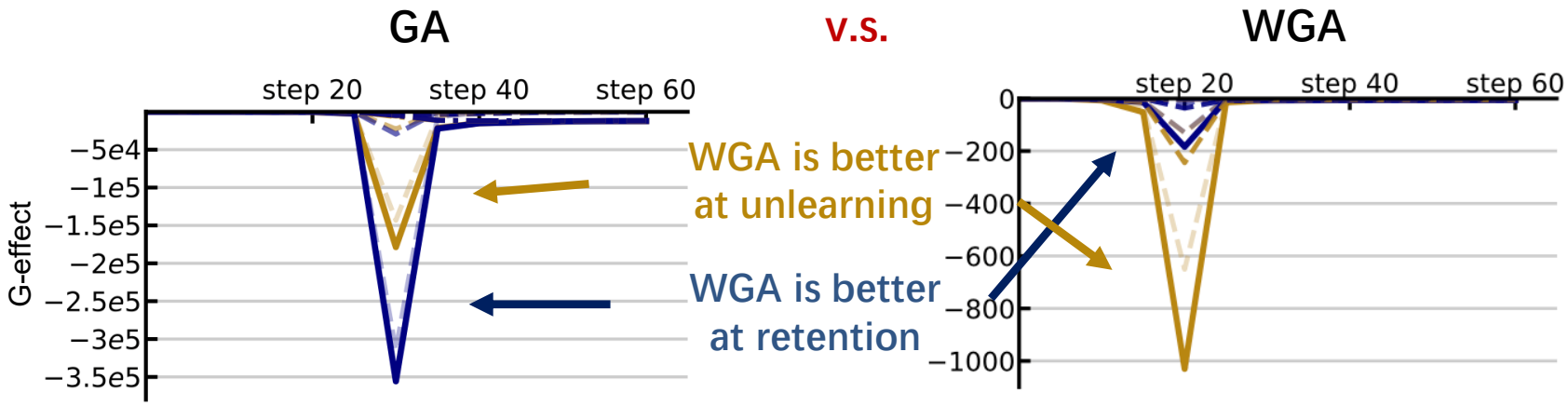
Reason. Large gradients will **accumulate** due to the chain rule, a general scenario holds for many other unlearning objectives.

WGA: Improvement 1

Motivation: Combating the inverse likelihood term via **loss reweighting.**

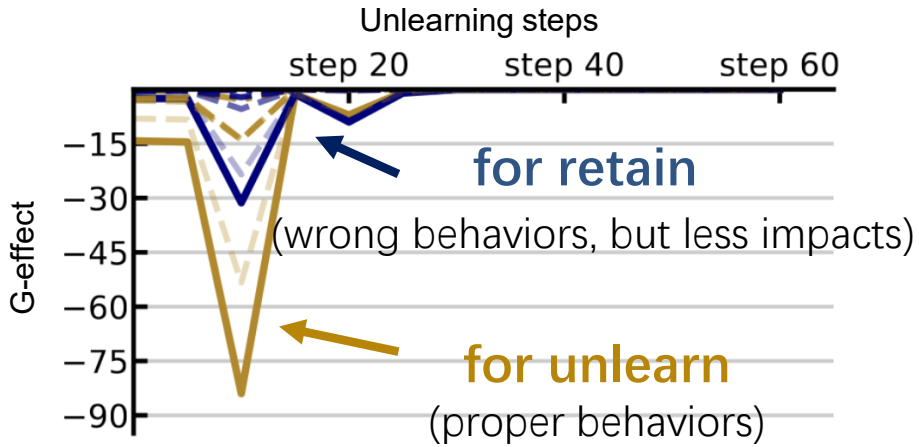
Original GA: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \sum_i \log P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)$ ➔ **Weighted GA:** $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \sum_i P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)^{\alpha} \log P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)$

Gradients: $\mathbb{E}_{s_\mathrm{u} \sim \mathcal{D}_\mathrm{u}} \sum_i \underbrace{P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)^{\alpha-1}}_{\text{counteract the inverse likelihood}} \nabla_{\boldsymbol{\theta}} P\left(s_\mathrm{u}^i | s_\mathrm{u}^{<i}; \boldsymbol{\theta}\right)$



Comparison of the G-effects between GA and WGA.

NPO: Objective 2



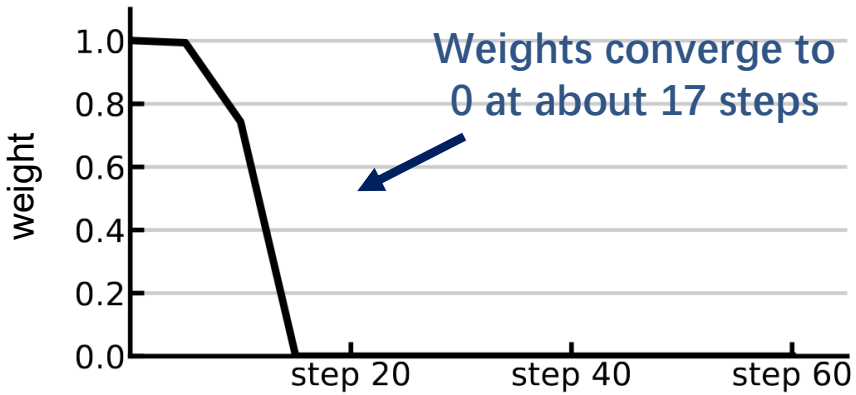
The G-effects of NPO.

Objective: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \frac{1}{\beta}\log\left(1+\left(\frac{p(s_\mathrm{u};\boldsymbol{\theta})}{p(s_\mathrm{u};\boldsymbol{\theta}_\mathrm{o})}\right)^{\beta}\right)$

Gradient: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \sum_i \underbrace{\frac{2P(s_\mathrm{u};\boldsymbol{\theta})^{\beta}}{P(s_\mathrm{u};\boldsymbol{\theta})^{\beta}+P(s_\mathrm{u};\boldsymbol{\theta}_\mathrm{o})^{\beta}}}_{w_\mathrm{npo}\text{ reweighting}} \nabla_{\boldsymbol{\theta}} \log P(s_\mathrm{u};\boldsymbol{\theta})$

Observation 4. NPO (Negative Preference Optimization) has **fewer negative impacts** on retention compared to GA.
Reason. The gradients of NPO are very similar to GA, yet further **reweighting** by w_npo, which mainly contributes to its improvements over GA.

NPO: Objective 2



The curve of w_{npo} during unlearning.

Objective: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \frac{1}{\beta} \log\left(1 + \left(\frac{p(s_\mathrm{u};\boldsymbol{\theta})}{p(s_\mathrm{u};\boldsymbol{\theta}_\mathrm{o})}\right)^{\beta}\right)$

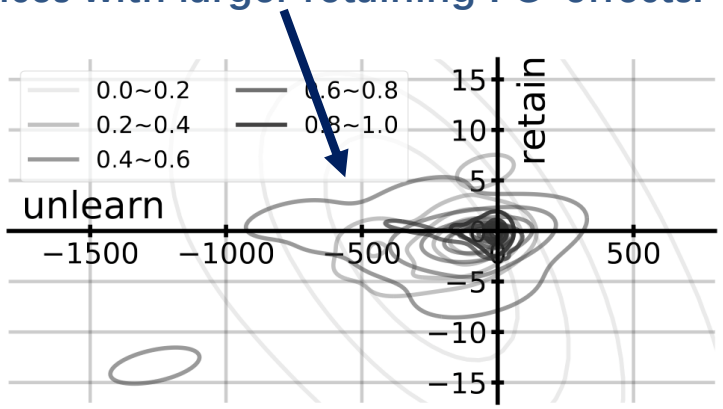
Gradient: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \sum_i \underbrace{\frac{2P(s_\mathrm{u};\boldsymbol{\theta})^{\beta}}{P(s_\mathrm{u};\boldsymbol{\theta})^{\beta}+P(s_\mathrm{u};\boldsymbol{\theta}_\mathrm{o})^{\beta}}}_{w_\mathrm{npo}\text{ reweighting}} \nabla_{\boldsymbol{\theta}} \log P(s_\mathrm{u};\boldsymbol{\theta})$

Observation 5. The NPO weight w_npo serves a role like **early stopping.**

Reason. w_npo approaches 0 when $P(s_\mathrm{u};\boldsymbol{\theta}) \to 0$.

NPO: Objective 2

Larger weights are assigned to those instances with larger retaining PG-effects.



The distributions of the point-wise G-effects across different range of w_{npo}.

Gradient: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \sum_i \frac{2P(s_\mathrm{u};\boldsymbol{\theta})^\beta}{P(s_\mathrm{u};\boldsymbol{\theta})^\beta + P(s_\mathrm{u};\boldsymbol{\theta}_\mathrm{o})^\beta} \nabla_{\boldsymbol{\theta}} \log P(s_\mathrm{u};\boldsymbol{\theta})$

G-effect: $\mathbb{E}_{\mathcal{D}_\mathrm{u}} \underbrace{w_\mathrm{npo}}_{\text{weights}} \underbrace{\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{s}_\mathrm{u};\boldsymbol{\theta})^\top \nabla_{\boldsymbol{\theta}} \mathcal{R}(\mathcal{D};\boldsymbol{\theta})}_{\text{point-wise G-effect (PG-effect)}}$

(The impacts of a particular data point on model performance.)

Observation 6. The NPO reweighting mechanism w_npo **prioritizes instances** that less damages retention.

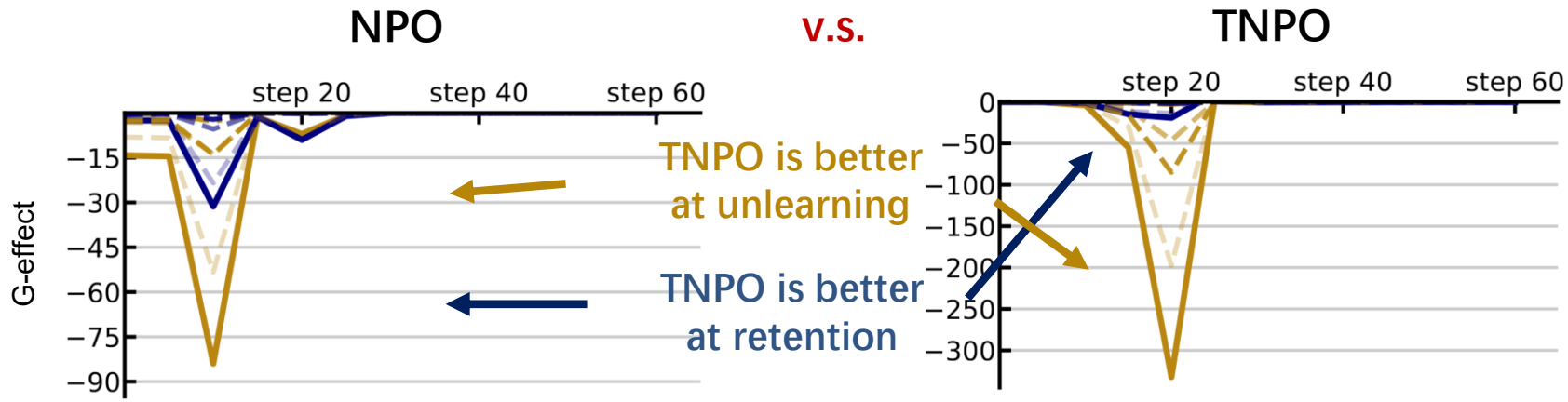
Reason. Data that have small impacts on **retention** also have small impacts on **unlearning.**

TNPO: Improvement 2

Motivation: Generalized the reweighting mechanism of NPO for tokens.

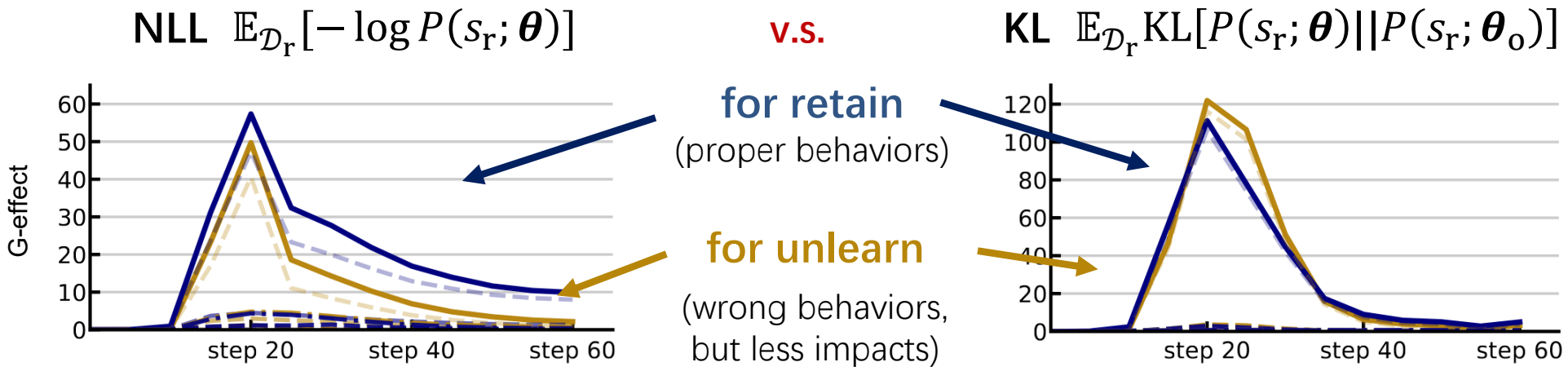
Token-wise NPO $\sum_i w^i_{\text{tnpo}} \log P\left(s^i_{\text{u}} | s^{<i}_{\text{u}}; \boldsymbol{\theta}\right)$ with $w^i_{\text{tnpo}} = \underbrace{\frac{2P\left(s^i_{\text{u}} | s^{<i}_{\text{u}}; \boldsymbol{\theta}\right)^{\alpha}}{P\left(s^i_{\text{u}} | s^{<i}_{\text{u}}; \boldsymbol{\theta}\right)^{\alpha} + P\left(s^i_{\text{u}} | s^{<i}_{\text{u}}; \boldsymbol{\theta}_{\text{o}}\right)^{\alpha}}}$

same reweighting scheme yet applied point-wise.



Comparison of the G-effects between NPO and TNPO.

Retain Objectives



Comparison between two representative retain objectives.

Observation 7. **NLL** and **KL** are both effective for retention, while KL can lead to overall larger retain G-effect, thus preferred.

Note. The unlearn G-effect for the unlearning objective is much larger than for the retain objectives. Thus, we do not need to worry about the side effect on unlearning.

Empirical Evaluations

LLM setup	method	Phi-1.5 ES-exact retain ↑	Phi-1.5 ES-exact unlearn ↓	Phi-1.5 ES-perturb retain ↑	Phi-1.5 ES-perturb unlearn ↓	Phi-1.5 MU ↑	Phi-1.5 FQ ↑	Llama-2-7B ES-exact retain ↑	Llama-2-7B ES-exact unlearn ↓	Llama-2-7B ES-perturb retain ↑	Llama-2-7B ES-perturb unlearn ↓	Llama-2-7B MU ↑	Llama-2-7B FQ ↑
before unlearning		0.44	0.59	0.21	0.16	0.52	-5.80	0.82	0.80	0.53	0.40	0.63	-7.59
1%	GA	0.11	0.05	0.08	0.08	0.37	-0.54	0.42	**0.05**	0.26	**0.04**	0.53	-0.54
	PO	**0.36**	0.84	**0.16**	0.36	**0.51**	-4.24	**0.75**	0.83	**0.47**	0.52	0.62	-5.80
	WGA	**0.36**	**0.03**	**0.18**	**0.02**	**0.51**	**-0.54**	**0.67**	0.08	0.38	0.06	**0.65**	**-0.08**
	NPO	0.27	0.09	0.11	0.07	0.48	-2.91	0.47	0.12	0.38	0.09	0.62	-1.32
	TNPO	0.33	**0.03**	0.12	**0.04**	0.49	**-0.08**	0.51	**0.03**	**0.43**	**0.03**	**0.64**	**-0.08**
	RMU	0.23	0.08	0.15	0.05	0.43	-0.54	0.23	0.08	0.15	0.05	0.52	-1.32
before unlearning		0.44	0.56	0.21	0.23	0.52	-29.65	0.82	0.77	0.53	0.41	0.63	-32.13
5%	GA	0.00	**0.00**	0.00	**0.00**	0.00	-11.40	0.03	**0.00**	0.02	**0.00**	0.00	**-12.42**
	PO	**0.26**	0.79	**0.16**	0.49	**0.51**	-26.50	**0.55**	0.84	**0.36**	0.49	**0.64**	-28.84
	WGA	**0.29**	0.01	**0.16**	0.01	**0.51**	**-1.30**	0.47	0.00	**0.39**	0.00	**0.64**	-16.32
	NPO	0.08	0.12	0.08	0.06	0.38	-7.75	0.17	0.07	0.12	0.08	0.52	**-9.95**
	TNPO	0.16	0.01	0.08	0.00	0.46	-2.18	**0.50**	0.01	0.34	0.00	0.63	-32.13
	RMU	0.21	**0.00**	0.12	**0.00**	0.27	**-1.95**	0.12	**0.00**	0.12	**0.00**	0.58	-21.44
before unlearning		0.44	0.47	0.21	0.18	0.52	-39.00	0.82	0.83	0.53	0.30	0.63	-44.45
10%	GA	0.00	**0.00**	0.00	**0.00**	0.00	-45.26	0.00	**0.00**	0.00	**0.00**	0.00	-20.86
	PO	**0.32**	0.73	0.14	0.26	0.50	-38.25	**0.55**	0.84	**0.37**	0.43	**0.62**	-39.76
	WGA	**0.34**	**0.00**	**0.16**	**0.00**	**0.51**	-9.06	**0.66**	0.02	**0.42**	**0.01**	0.62	-24.85
	NPO	0.08	0.09	0.07	0.07	0.38	-10.57	0.12	0.13	0.10	0.14	0.50	**-12.19**
	TNPO	0.20	0.01	0.09	0.01	**0.50**	**-7.66**	0.45	**0.01**	0.26	0.01	**0.63**	**-13.47**
	RMU	0.03	0.05	0.03	0.06	0.31	**-7.00**	0.25	0.01	0.20	0.01	0.59	-16.72

Comparison between unlearning objective on TOFU with KL regularization.

Observation 8. Larger unlearning datasets and smaller model sizes make it more challenging to unlearn.

Observation 9. GA-based works (GA & TNPO) are superior to other lines of works like PO or RMU.

Observation 10. Instance-wise reweighting is promising for unlearning efficacy.

Take Home Messages

General knowledge within **shallow layers undergoes substantial alterations** over deeper layers during unlearning.

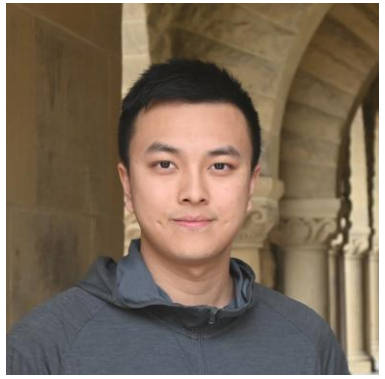
Although conceptually existing, **current objectives all fail** to retain the overall performance when conducting unlearning.

Prioritizing some tokens is effective for unlearning. However, there still exists a large space to further refine weighting mechanisms.

With **excessive unlearning**, the deterioration in common model responses can outweigh improvements in unlearning.

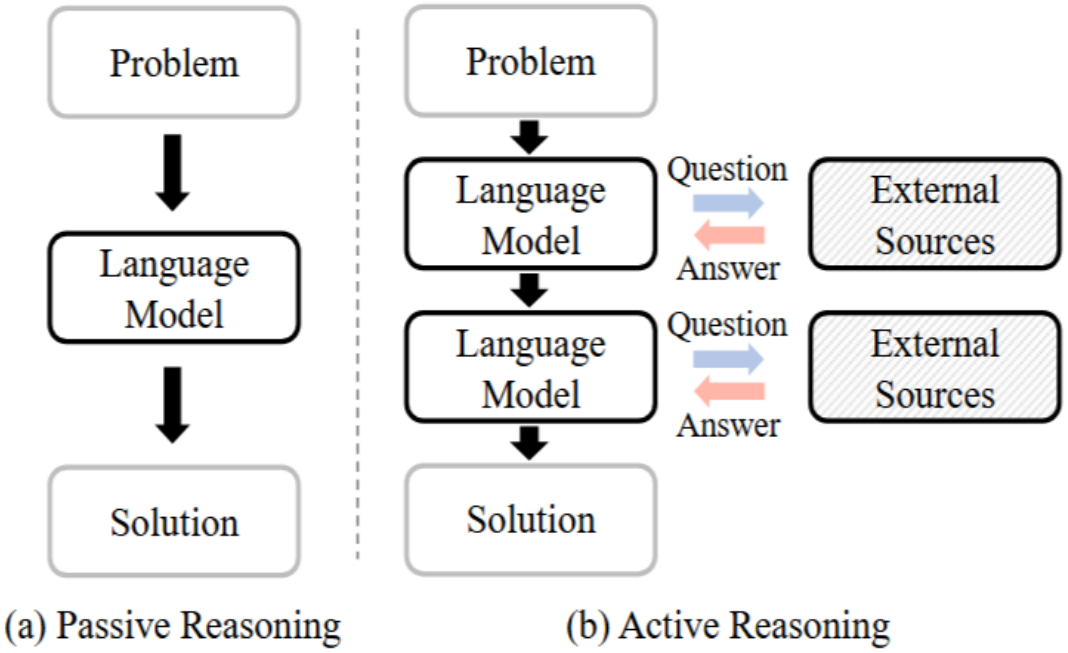
Part III: Reasoning

Can LLMs Ask the Right Questions under Incomplete Information?



Zhanke Zhou

Xiao Feng



(a) Passive Reasoning (b) Active Reasoning

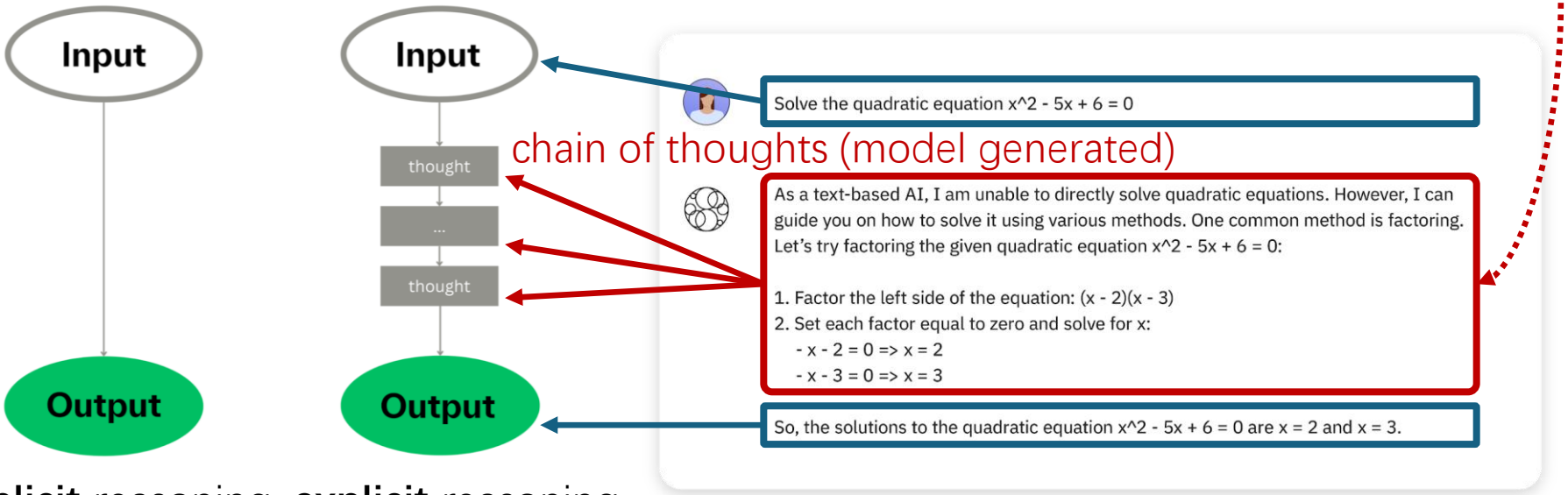
Zhanke Zhou, **Xiao Feng**, Zhaocheng Zhu, Jiangchao Yao, Sanmi Koyejo, **Bo Han**.
From Passive to Active Reasoning: Can Large Language Models Ask the Right Questions under Incomplete Information? In *ICML*, 2025

Background

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

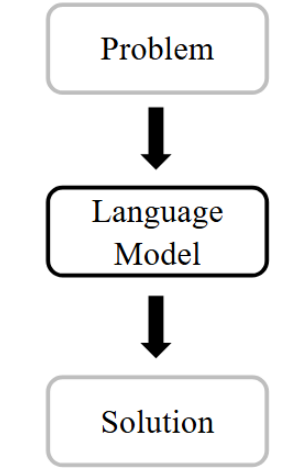
deepseek R1

OpenAI o3

Passive Reasoning

Existing works primarily focus on passive reasoning.

- A model is provided **necessary information** (e.g., conditions in a 24-Game problem).
- The model conducts **step-by-step reasoning** to derive the solution.



(a) Passive Reasoning

Passive Reasoning

Problem: Numbers: 3, 7, 8, 9
Goal: Use addition (+), subtraction (-), multiplication (×), and division (÷) to make 24. Each number must be used exactly once.

Solution:

1. 9 - 8 = 1
2. 7 + 1 = 8
3. 3 x 8 = 24

Done.

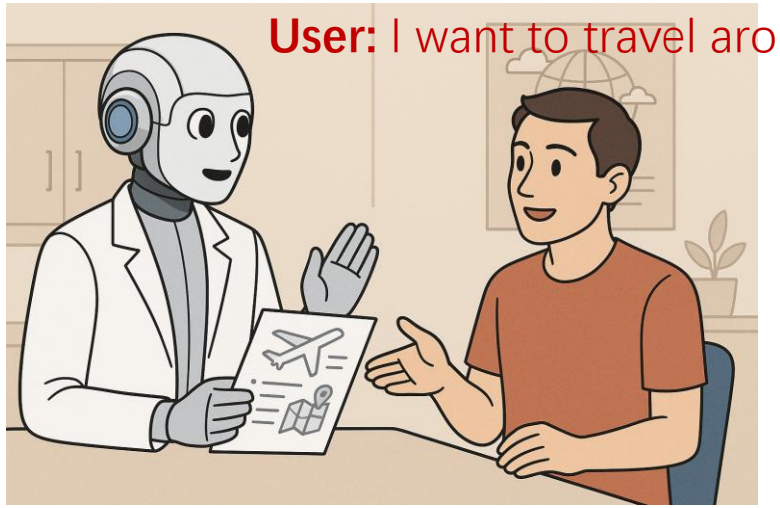
Beyond the Common Assumption

Prior works all assume that the given information is **complete.**

But what if the initially given information is **incomplete?** 🤔

Scenario 1: Travel Planner

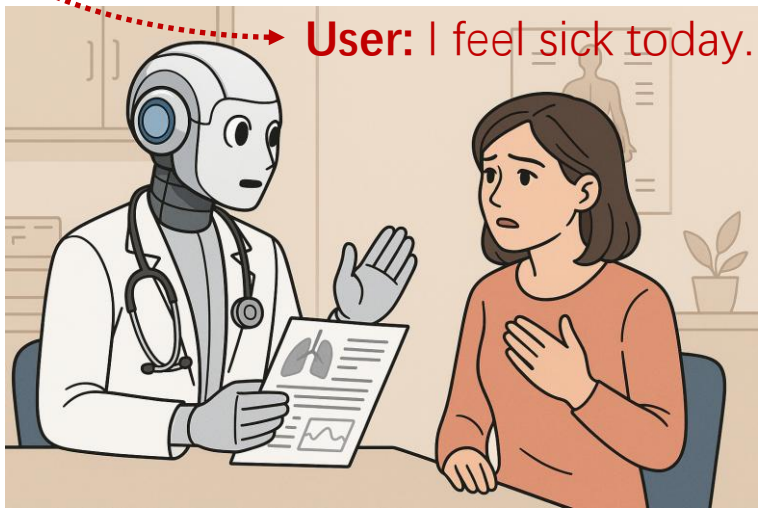
User: I want to travel around HK.



Unknown:
budget?
preferences?
time slots?
...

Scenario 2: Healthcare

User: I feel sick today.



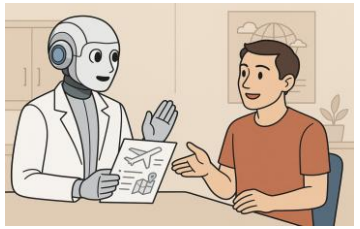
Unknown:
symptoms?
history?
examination?
...

Pictures are generated by GPT-4o.

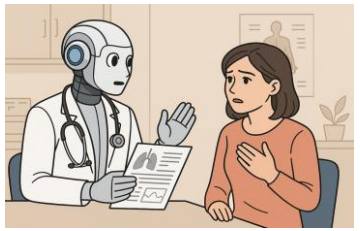
New Research Problem: Active Reasoning

Active Reasoning: The model has to actively interact with **external information sources** to **seek more essential information** and then draw the conclusion.

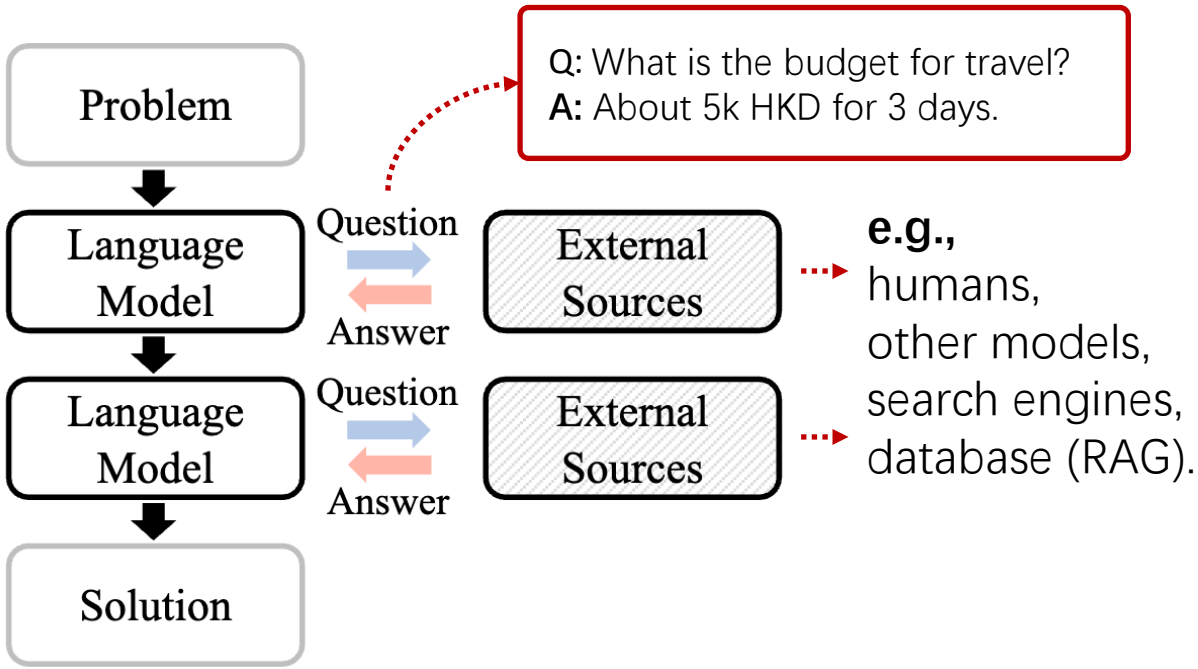
Travel Planner



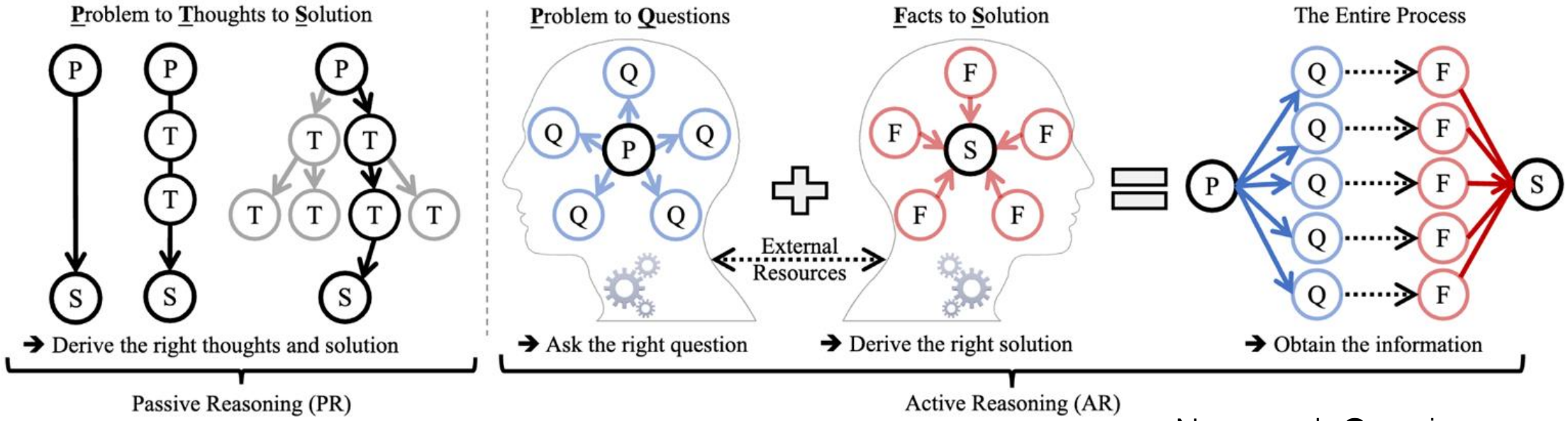
Healthcare

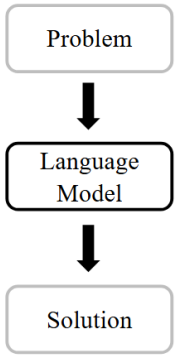


abstract to

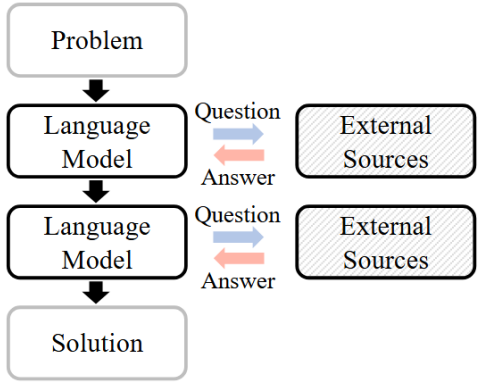


Passive Reasoning v.s. Active Reasoning





(a) Passive Reasoning



(b) Active Reasoning

Note: each **Q**uestion corresponds to an Answer (**F**act), all facts lead to **S**olution.

The Research Gap

So far, the AR capabilities of LLMs remain **largely under-explored**.

Existing AR datasets are relatively **simple** for the advanced models.

"LLMs are much more intelligent than most humans on most exams, but much less useful than most humans on most real-world tasks."

—The second half (by Shunyu Yao, Researcher at Tencent).

Therefore, it is necessary to conduct **a systematic evaluation with a new benchmarking dataset** that is tailored to active reasoning.

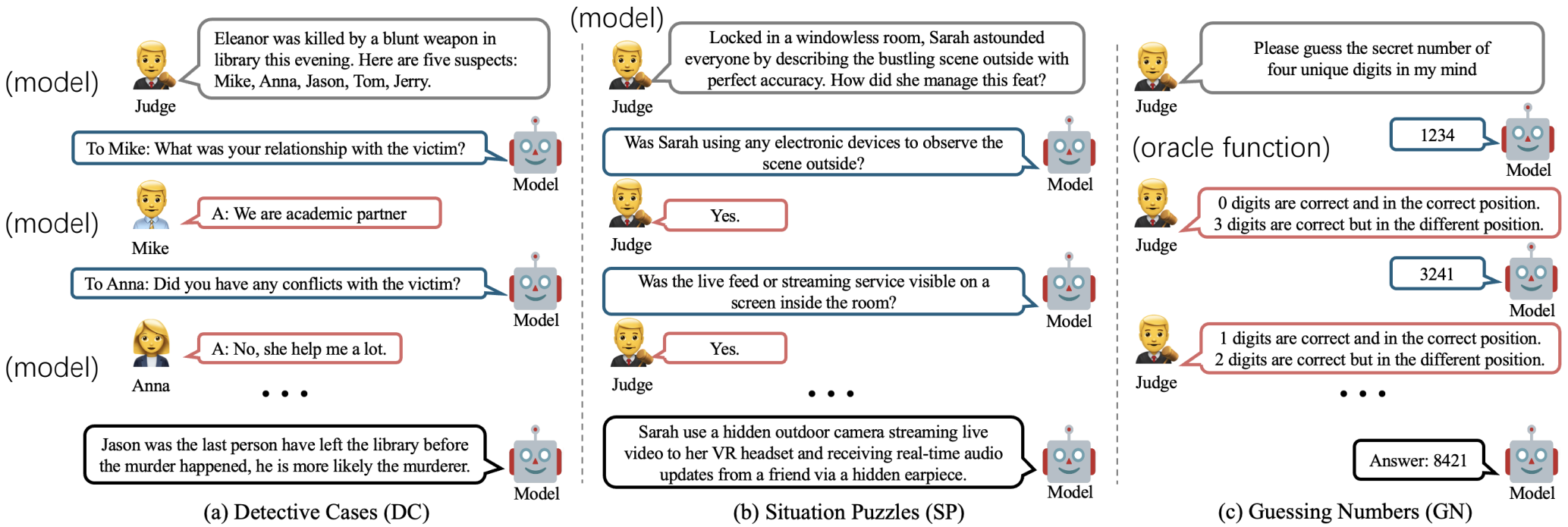
AR-Bench

Task	DC	SP	GN
Size (train/test)	400/100	400/100	4940/100
Avg. problem tokens	564.06	178.53	176.00
Interaction feedback	Narrative	Yes/No	Info. about correct digits
Answer space	5	-	5040
Metric	Accuracy	F1 score	Exact match

Table 2: Dataset statistics for the three tasks in AR-Bench.

AR-Bench (Active Reasoning Benchmark) contains **6040** problems, covering **3** AR tasks:

- **Detective Cases (DC):** The interrogation between a detective and 5 suspects.
- **Situation Puzzles (SP):** The game to reveal the truth from a puzzling mystery.
- **Guessing Numbers (GN):** The game to uncover a 4-unique-digits number.



The interaction rounds are set to 25.

Position

The uniqueness of AR-Bench

- Evaluate LLMs reasoning under **different types of feedback.**
- Require **active information-seeking** and **deliberate reasoning.**

Paradigm	Dataset	Incomplete problem	External interaction	Hypothesis verification	Language feedback	Symbolic feedback	Complex reasoning
Passive Reasoning	CommonsenseQA (Talmor et al., 2019)	✗	✗	✗	✗	✗	✗
	SocialIQA (Sap et al., 2019)	✗	✗	✗	✗	✗	✗
	GSM8K (Cobbe et al., 2021)	✗	✗	✗	✗	✗	✓
	MMLU (Hendrycks et al., 2021a)	✗	✗	✗	✗	✗	✓
	Game24 (Yao et al., 2023)	✗	✗	✗	✗	✗	✓
	Crosswords (Yao et al., 2023)	✗	✗	✗	✗	✗	✓
	Blocksworld (Valmeekam et al., 2023)	✗	✗	✗	✗	✗	✓
Active Reasoning	Qulac (Aliannejadi et al., 2019)	✓	✓	✗	✓	✗	✗
	Abg-CoQA (Guo et al., 2021)	✓	✓	✗	✓	✗	✗
	20 Questions (Abdulhai et al., 2023; Hu et al., 2024)	✓	✓	✓	✓	✗	✗
	Guess My City (Abdulhai et al., 2023)	✓	✓	✓	✓	✗	✗
	Trouble Shooting (Hu et al., 2024)	✓	✓	✓	✓	✗	✗
	MediQ (Li et al., 2024c)	✓	✓	✓	✓	✗	✗
	AR-Bench (ours)	✓	✓	✓	✓	✓	✓

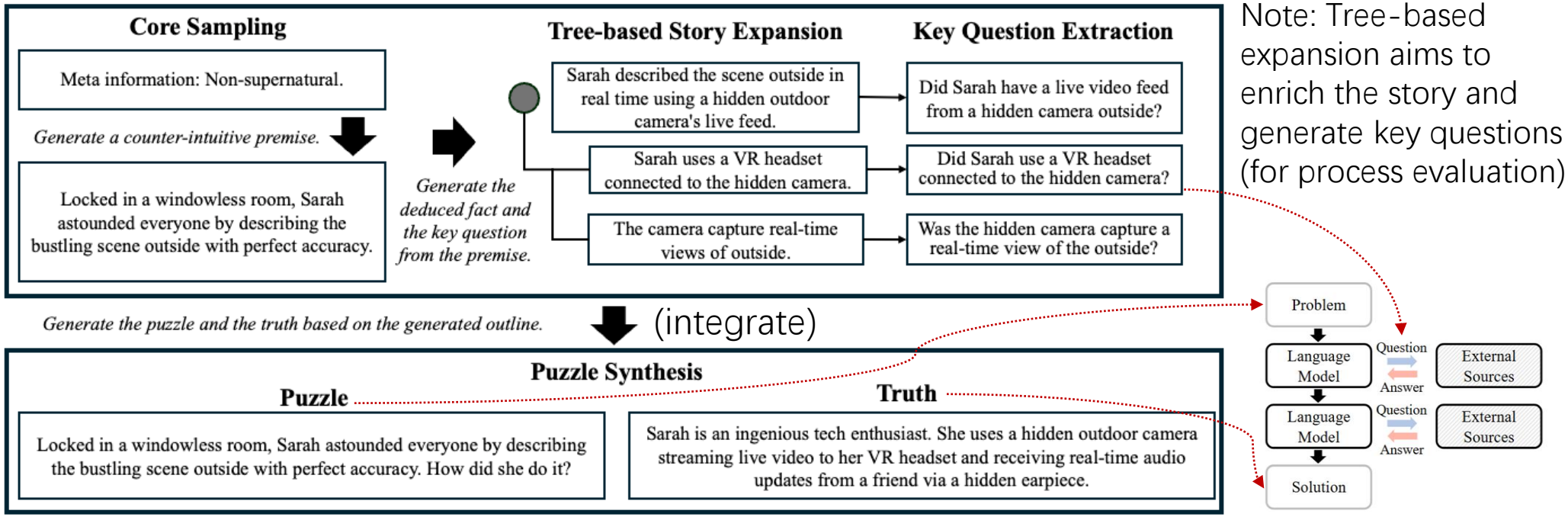
- **Hypothesis verification:** Generates initial hypothesis and verifies via asking questions.
- **Language feedback:** Receives natural-language feedback from external sources.
- **Symbolic feedback:** Receives symbolic feedback from external sources.
- **Complex reasoning:** Derives the solution through multi-step reasoning.

Dataset Construction

4-step LLM-driven construction (with a follow-up manual check)

1. **Core Sampling** to collect outlines/topics.
2. **Tree-based Story Expansion** to enrich the details of outlines/topics.
3. **Key Question Extraction** for fine-grained process analysis.
4. **Puzzle Synthesis** to construct the narrative of puzzles and truths.

(LLM-driven data generation)



Note: Tree-based expansion aims to enrich the story and generate key questions (for process evaluation).

Evaluation Metrics

Outcome score: The quality of the conclusion after the conversation.

- Detective Cases (DC): Accuracy.
- Situation Puzzles (SP): Character-level F1 score.
- Guessing Numbers (GN): Exact-match rate.

Process score: The quality of the question-answering record.

- DC & SP: How many key questions can be answered with the record.
- GN: How many digits of the guessing number match the ground truth.

Outcome Score. Process Score.

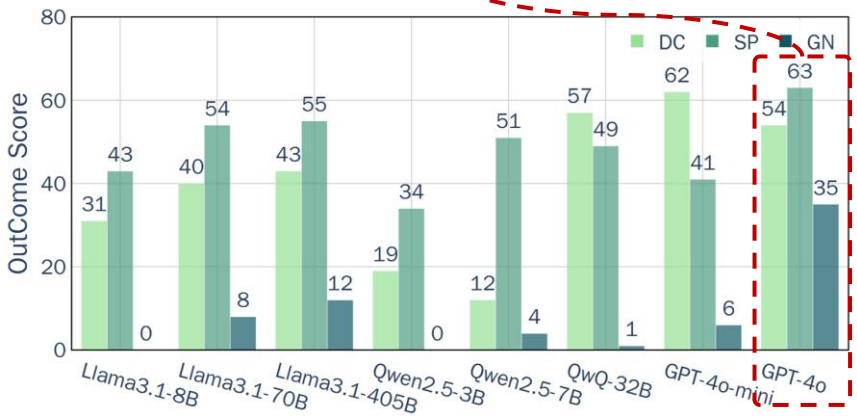
Process score of GN: (**Number of correct digits in correct positions**)/4 + 0.5 × (**Number of correct digits in wrong positions**)/4.
An example of outcome score and process score at GN (Ground truth: 2048).
Proposed number: **2048**, outcome score: 1, process score: 1.
Proposed number: **2084**, outcome score: 0, process score: 0.75.

Empirical Findings (Outcome Score)

Observations 1 & 2: AR-Bench demonstrates **challenges** across existing models and methods.

(compare different models)

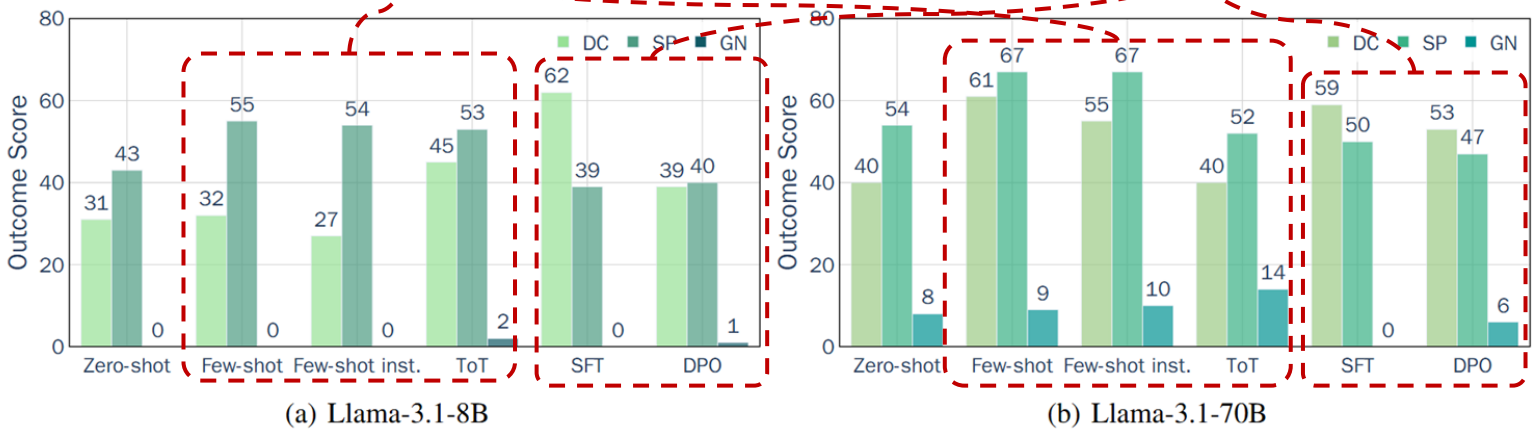
GPT-4o only has **35%** match rate in GN; other models even worse.



(compare different methods; zero-shot as the baseline)

Prompting-based methods exhibit **marginal** improvement on average across 3 tasks.

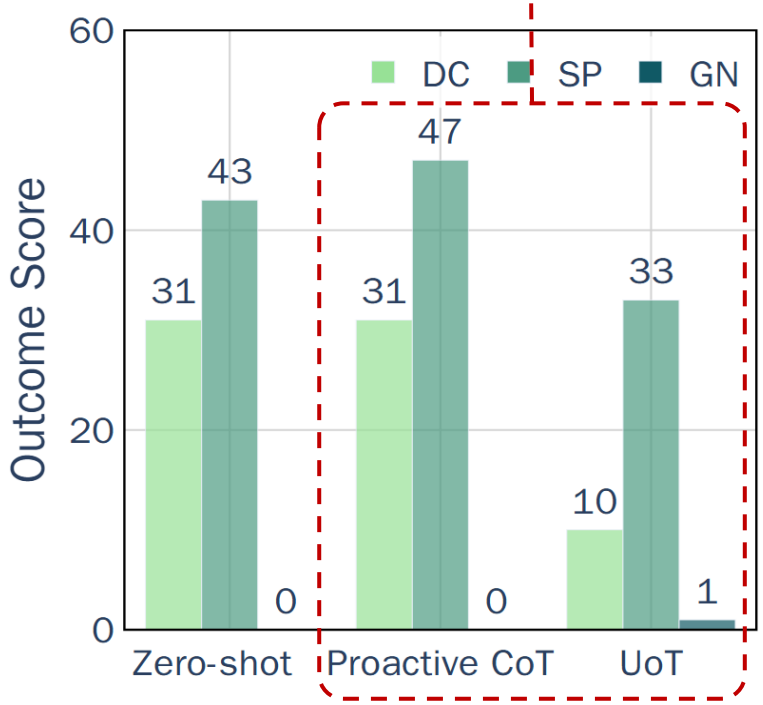
Post-training methods can even **degenerate** AR performance (in SP and GN).



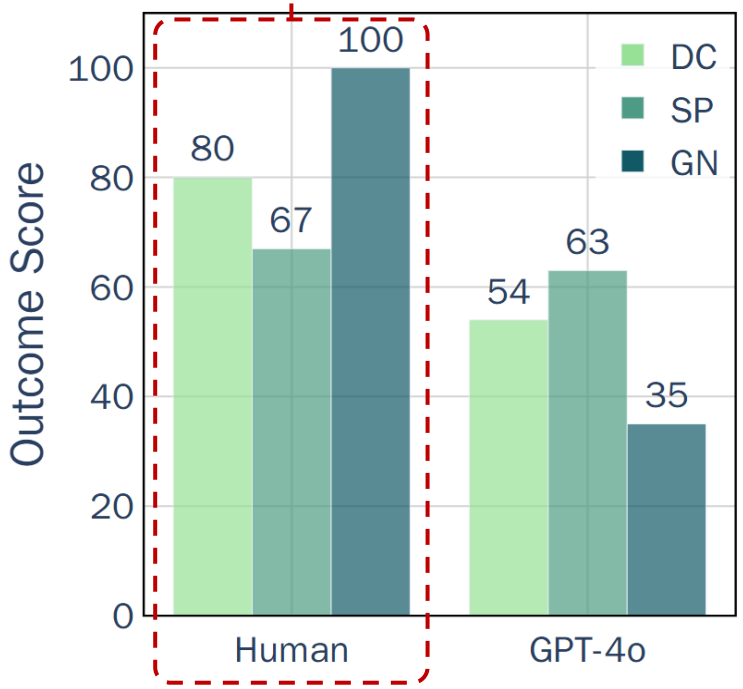
(a) Llama-3.1-8B (b) Llama-3.1-70B

Empirical Findings (Outcome Score)

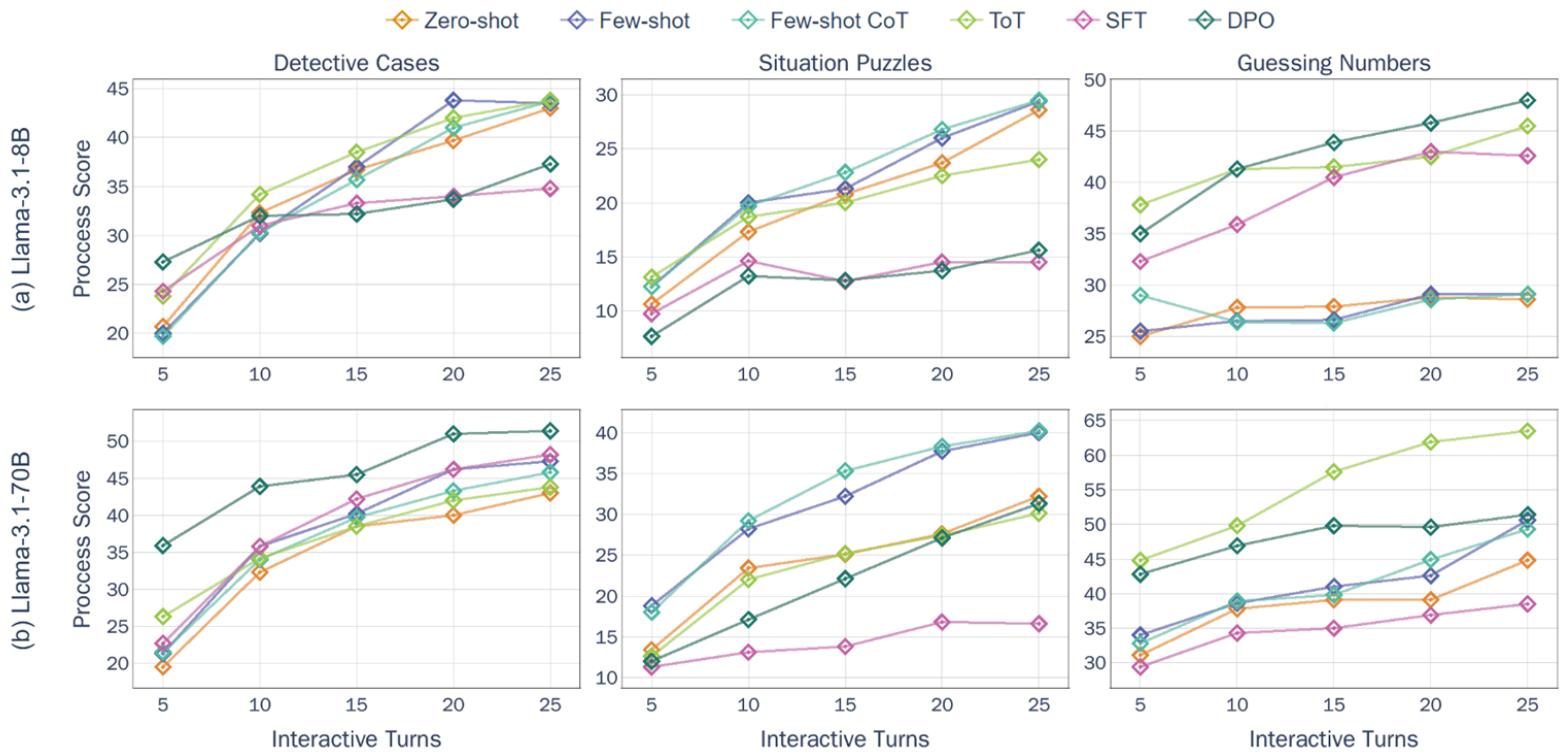
Observation 3:
Existing active reasoning methods **fail** in AR-Bench.



Observation 4:
Human baselines significantly surpass cutting-edge language models.



Empirical Findings (Process Score)



Observations 5, 6, 7:
LLMs struggle to consistently propose **high-quality questions.**
Unreliable verifier (using LLM-as-a-judge in ToT) limits the performance of search methods on tasks with language feedback (DC and SP).

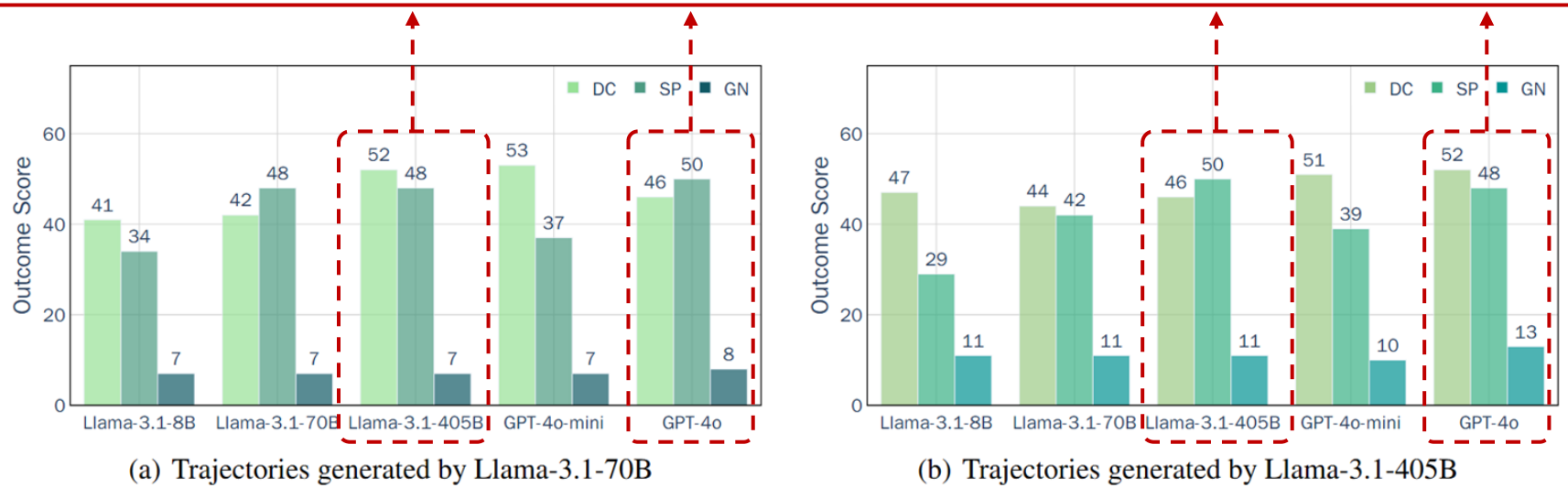
Empirical Findings (Process Score)

Observation 8: Underperforming LLMs ask **low-quality** questions.

Observation 9: **Larger models** can retrieve **more useful information** by proposing questions in interactions.

Empirical Findings (Process Score)

Observation 10: Given process conversations (trajectories) and directly ask for the solution, **larger models** demonstrate **higher robustness** to **insufficient information** (on average).



(a) Trajectories generated by Llama-3.1-70B

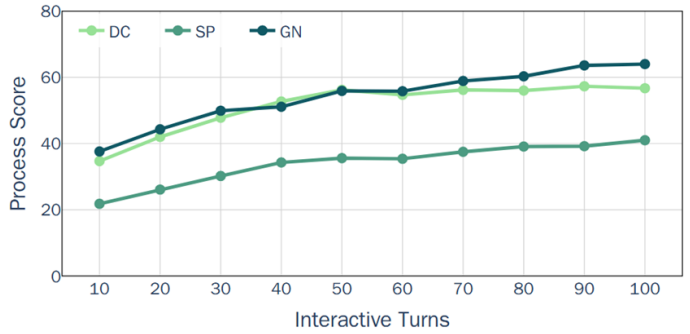
(b) Trajectories generated by Llama-3.1-405B

Note: We collect the QA record of 70B/405B model,
and feed the QA record to different models to generate solutions.

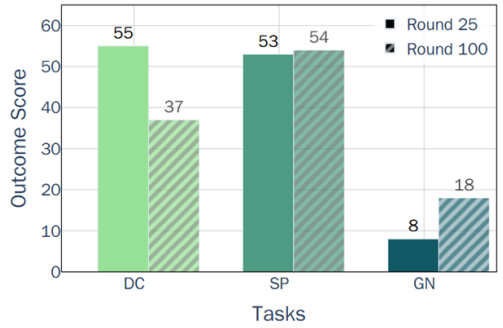
Empirical Findings (Ablation Study)

The question-asking scaling effect in AR

Observation 11: Question-asking **scaling cannot fully solve** the active reasoning tasks.



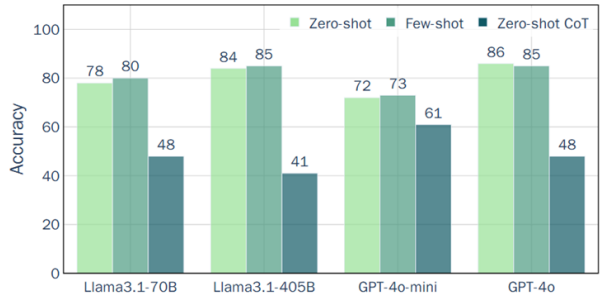
(a) Process Score



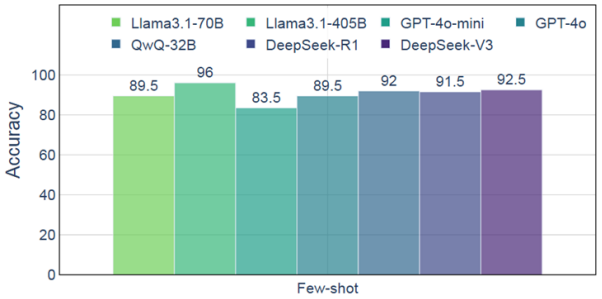
(b) Outcome Score

Verifying the reliability of the LLM judge

Observation 12: Larger models, as the judge, tend to provide more **reliable responses.**



(a) TurtleBenchmark



(b) AR-Bench

Empirical Findings (Case Study)

Incorrect case in DC

A Distinguished historian Dr. Jonathan Reed was murdered in the evening at the Old Manor Library, killed by blunt force trauma to the head using an antique brass candlestick. ...

Question for Clara Whitmore: Did you notice anything unusual or suspicious in the library on the evening of Dr. Reed's murder?
Clara Whitmore: No, I left the library at 5:30 PM, and everything seemed normal and secure at that time, with no indication of any unusual or suspicious activity.

Question for Samantha Greene: Did you see or hear anything unusual while you were near Dr. Reed's office in the library?
Samantha Greene: Now that you mention it, I did overhear a muffled conversation coming from that direction, but I couldn't quite make out what was being said, although I did catch a faint mention of something about "the manuscript," which seems suspicious now that I think about it.

Observation 13:
Models ask broad and non-specific questions.

Incorrect case in SP

Puzzle: Sarah found herself locked out of her own house without ever leaving the room she was in. How could this happen?

Q: Was Sarah in a bathroom when she found herself locked out?
A: No

Q: Was Sarah in a basement or attic when she found herself locked out?
A: Unknown

Observation 14:
Models ask repeated or unhelpful questions.

Incorrect case in GN

Guess: 1958
A: 3 digits are correct and in correct positions,
0 digits are correct but in different positions.

Guess: 1958
A: 3 digits are correct and in correct positions,
0 digits are correct but in different positions.

Task	Error Pattern	Model	
		Llama-3.1-8B	GPT-4o
DC	Timeline Misinterpretation	10%	31%
	Evidence Overlooked	61%	15%
SP	Evidence Overlooked	36%	44%
	Unsupported Assumptions	90%	72%
GN	Feedback Misunderstanding	78%	61%
	Incomplete Testing	81%	55%

Table 3: Error pattern analysis for AR-Bench. Proportions indicate the frequency of specific error types in error cases.

Observation 15 (Error Patterns):
Timeline Misinterpretation: models request information about irrelevant periods.
Evidence Overlooked: models fail to identify critical evidence needed to uncover the truth.
Unsupported Assumptions: models introduce fabricated details in their conclusions.
Feedback Misunderstanding: models fail to track the correct and eliminate the incorrect with feedback.
Incomplete Testing: models fail to identify the position of a correct digit.

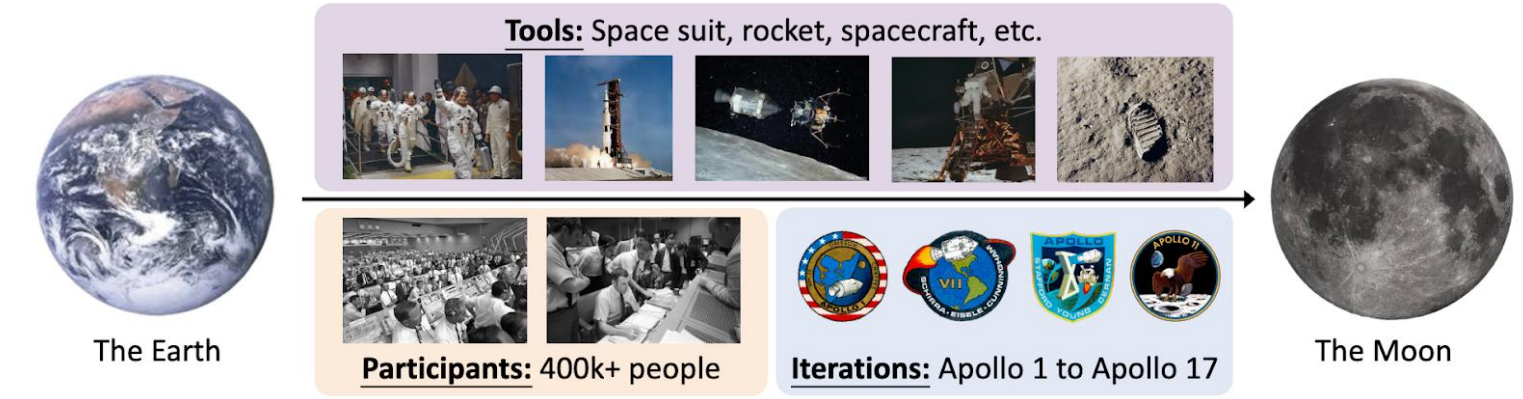
Take Home Messages

We construct **AR-Bench** to systematically evaluate the active reasoning capability of prevailing LLMs, and reveal **a general vulnerability** of LLMs in solving AR tasks.

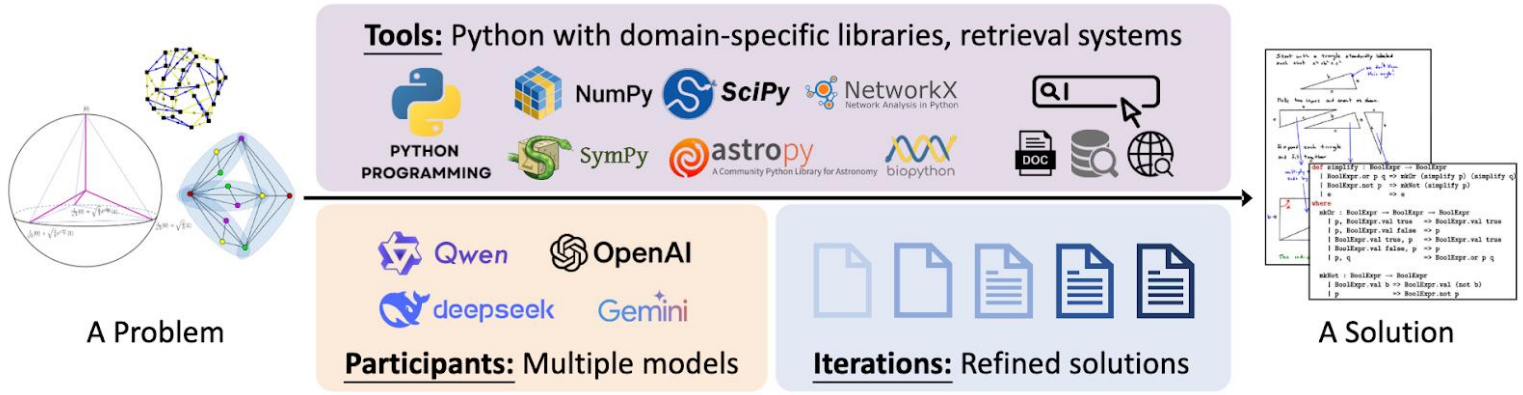
- Active reasoning demonstrates challenges across **all models and methods.**
- Human reasoners significantly **surpass** cutting-edge LLMs.
- LLMs struggle to consistently ask **high-quality questions** to retrieve critical information across long-horizon conversations.

AlphaApollo: A System for Deep Agentic Reasoning

- Orchestrating Foundation Models and Professional Tools into a Self-Evolving System.



(a) The Apollo Program (in 1960s) for moon landing with humans

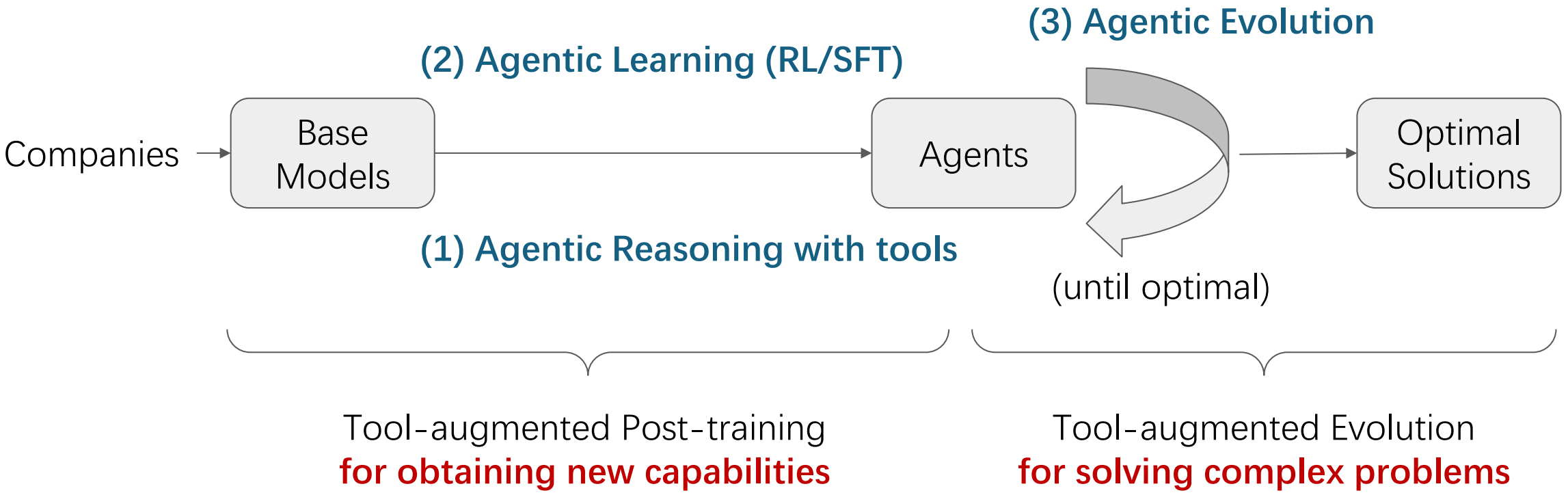


(b) The AlphaApollo System (ours) for problem solving with foundation models

Project homepage: https://alphaapollo.org/.

An Overview of AlphaApollo

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



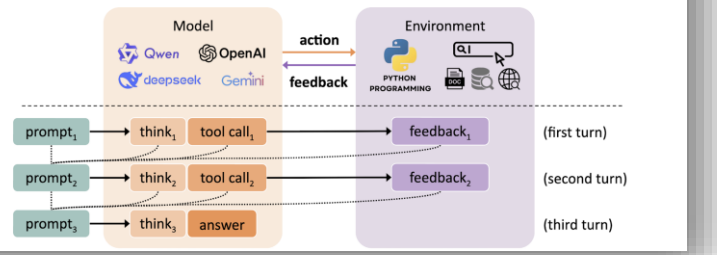
Project homepage: https://alphaapollo.org/.

An Overview of AlphaApollo

Feature 1: Agentic Reasoning

Agentic Reasoning (multi-turn interaction between model and environment).

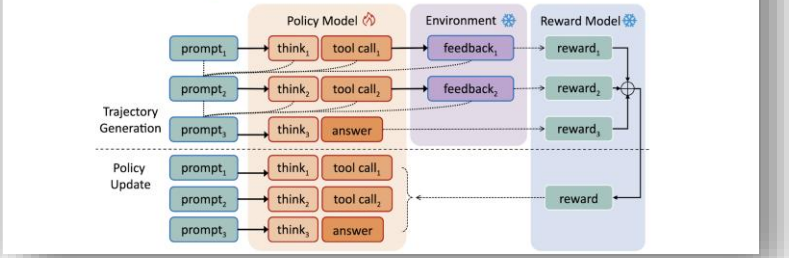
- Given the prompt, the model generate output (think/tool call/answer tokens).
- The environment parses output, executes tool, and give feedback to the model.



Feature 2: Agentic Learning

Agentic Learning (multi-turn optimization on the output of model).

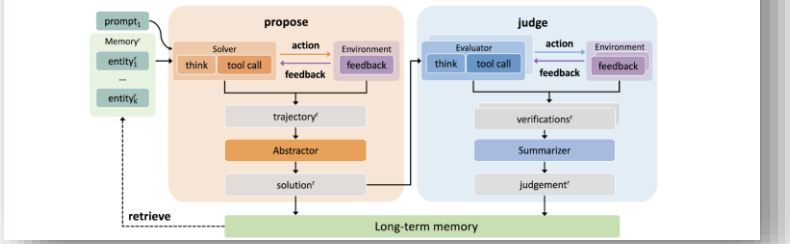
- Incorporates VeRL into a stable, turn-level agentic learning.
- Supports multiple algorithms (e.g., GRPO/SFT) and models (e.g., Qwen).



Feature 3: Agentic Evolution

Agentic Evolution (a test-time mechanism to evolve solutions).

- Operates through a propose-judge-update loop of multi-round evolution.
- Long-term memory to enable long-horizon evolution.
- Parallel (distributed) evolution to support scalable evolution with multiple mode



Quick Start Guidance

Installation

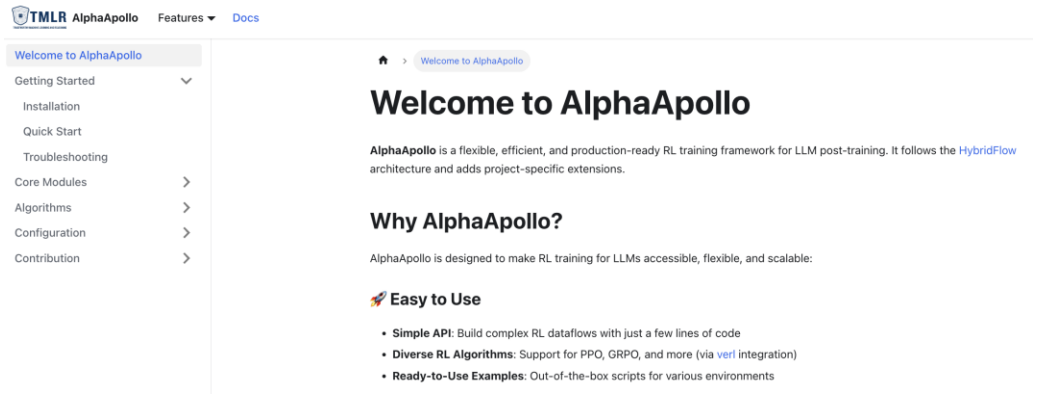
BASH Copy

```
conda create -n alphaapollo python==3.12 -y
conda activate alphaapollo

git clone https://github.com/tmlr-group/AlphaApollo.git
cd AlphaApollo

bash installation.sh
```

Document & Tutorial



Project homepage: https://alphaapollo.org/.

Demonstrations of AlphaApollo

Code: https://github.com/tmlr-group/AlphaApollo

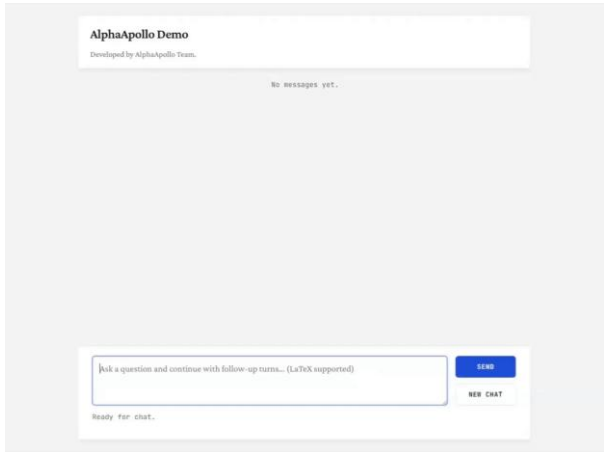
Host a model:

```
CUDA_VISIBLE_DEVICES=0,1,2,3
python -m
vllm.entrypoints.openai.api_server
\
--model
/data1/models/hub/Qwen2.5-
72B-Instruct \
--tensor-parallel-size 4 \
--port 8000
```

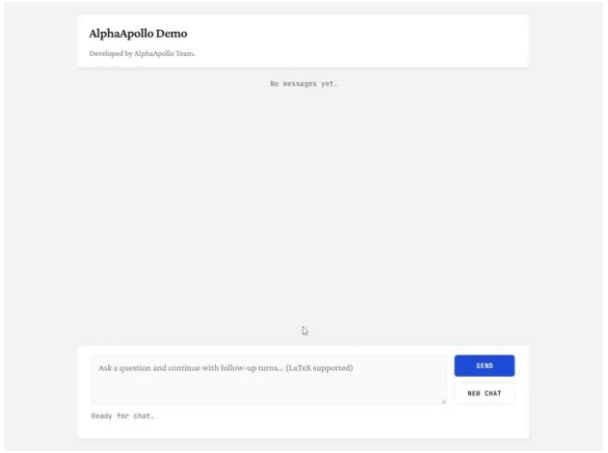
Run local web UI:

```
MODE=web
VLLM_MODEL="/data1/models/h
ub/Qwen2.5-72B-Instruct" \
bash
examples/demo/run_terminal_de
mo_vllm.sh
```

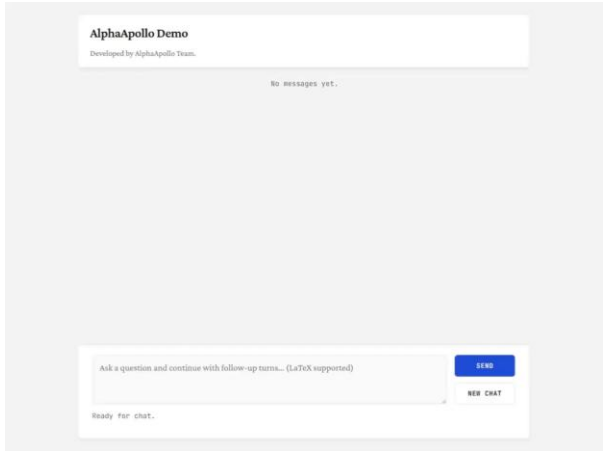
QUESTION 1



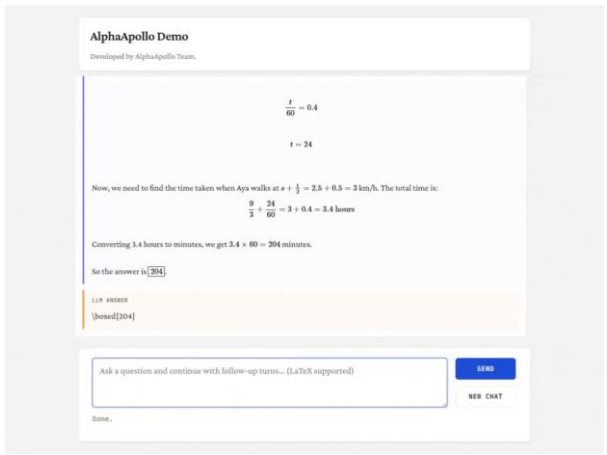
QUESTION 2



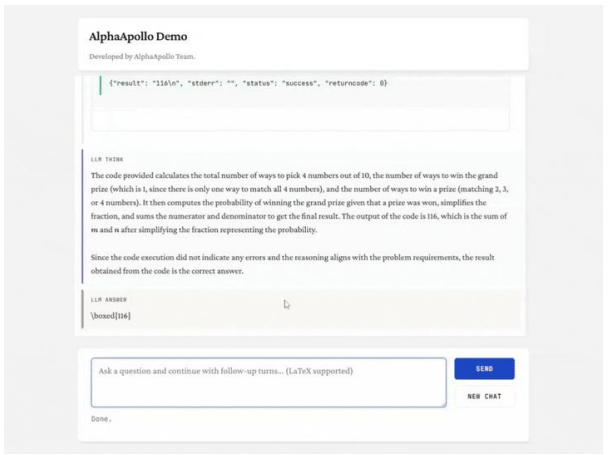
QUESTION 3



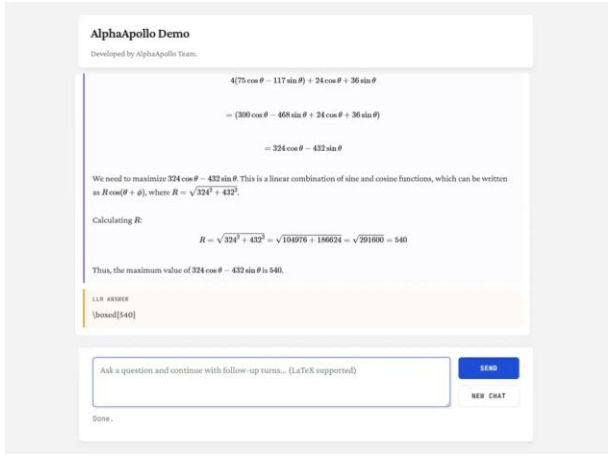
FOLLOW UP 1



FOLLOW UP 2



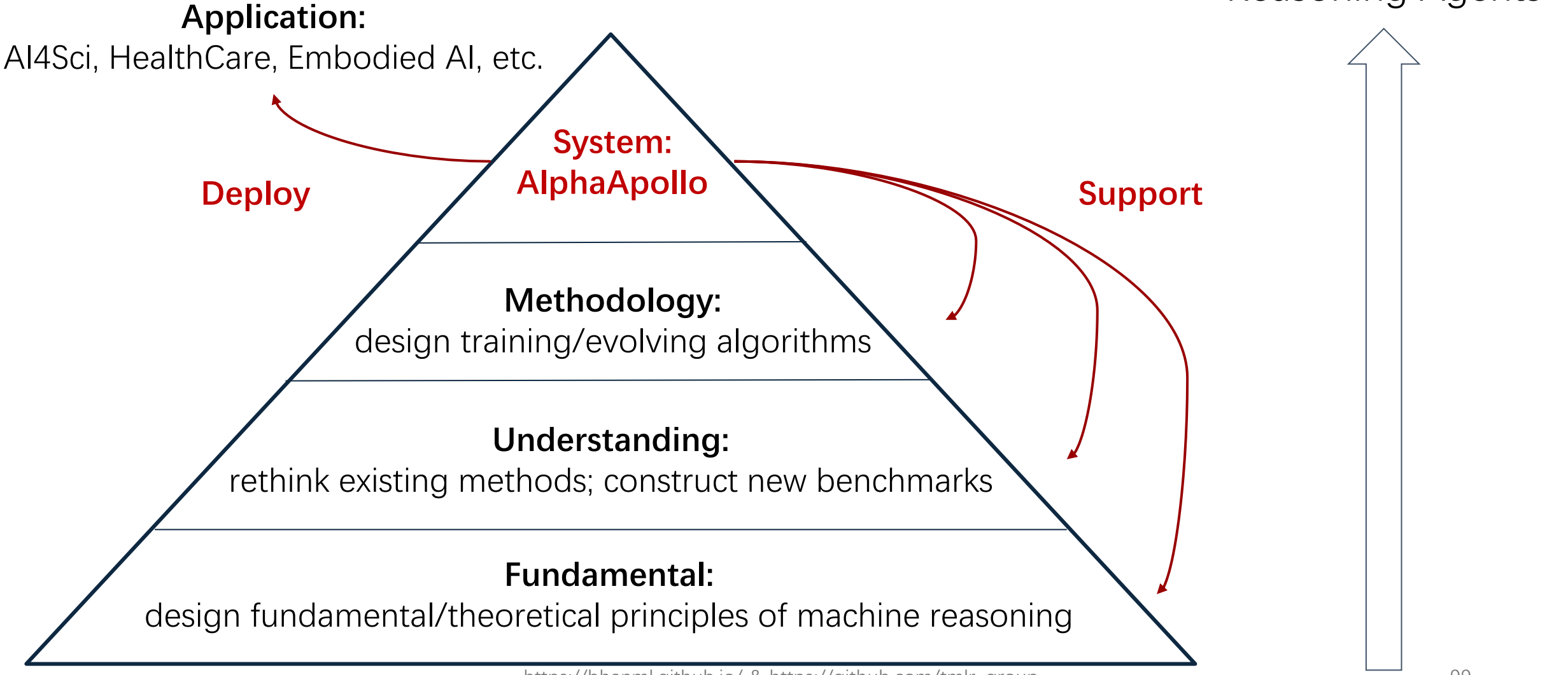
FOLLOW UP 3



Project homepage: https://alphaapollo.org/.

The Research Scope of AlphaApollo

Towards Trustworthy Reasoning Agents



Project homepage: https://alphaapollo.org/.

Future Directions

Robust pre-training/fine-tuning methods are required for VLMs.

- VLMs can still be mislead by spurious features.
- Larger models and high-quality data lead to better robustness.

The trade-off between unlearning and retention remains a critical issue.

- Current unlearning objectives all have negative impacts on retention.
- Data and optimization aspects of unlearning are not well explored.

The broader tasks of active reasoning and corresponding methods are necessary.

- Broader tasks of active reasoning (especially agentic scenarios) can be further investigated.
- Advanced post-training methods to enhance active reasoning are under-explored.

Open Questions

Robust pre-training/fine-tuning methods are required for VLMs.

- How to design robust pre-training/fine-tuning algorithms for VLMs?

The trade-off between unlearning and retention remains a critical issue.

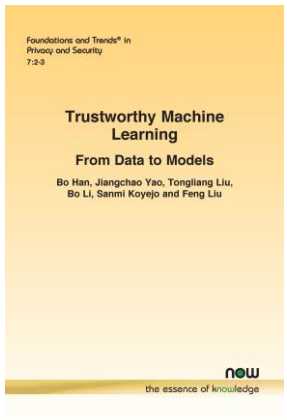
- How to explore data and optimization aspects of unlearning?

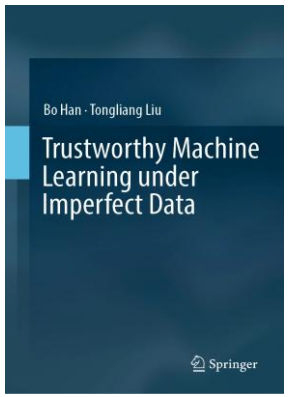
Active reasoning can be further investigated.

- How to design active reasoning algorithms for LLMs?

Appendix

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





- Tutorial and Lecture:
  - AAAI 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
  - AAAI 2026 Tutorial on The Science and Practice of Machine Unlearning for AI Safety
  - AAAI 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

https://bhanml.github.io & https://github.com/tmlr-group