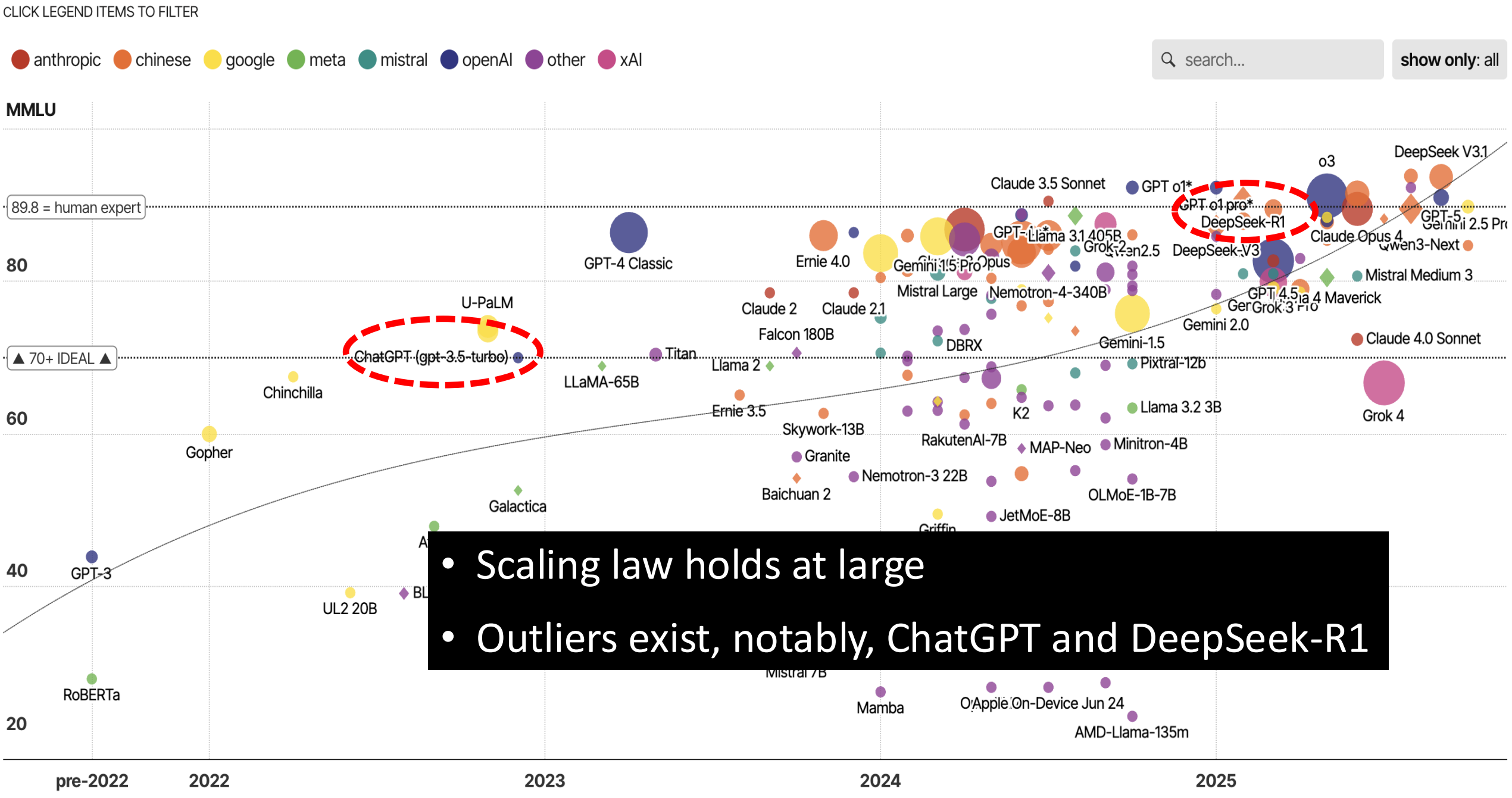
Reinforcement Learning for Large Language Models: From Fine-Tuning to Finer-Tuning to Auxiliary Policy Models

Haipeng Chen

Data Driven Decision Intelligence (D3i) Lab

Assistant professor, William & Mary

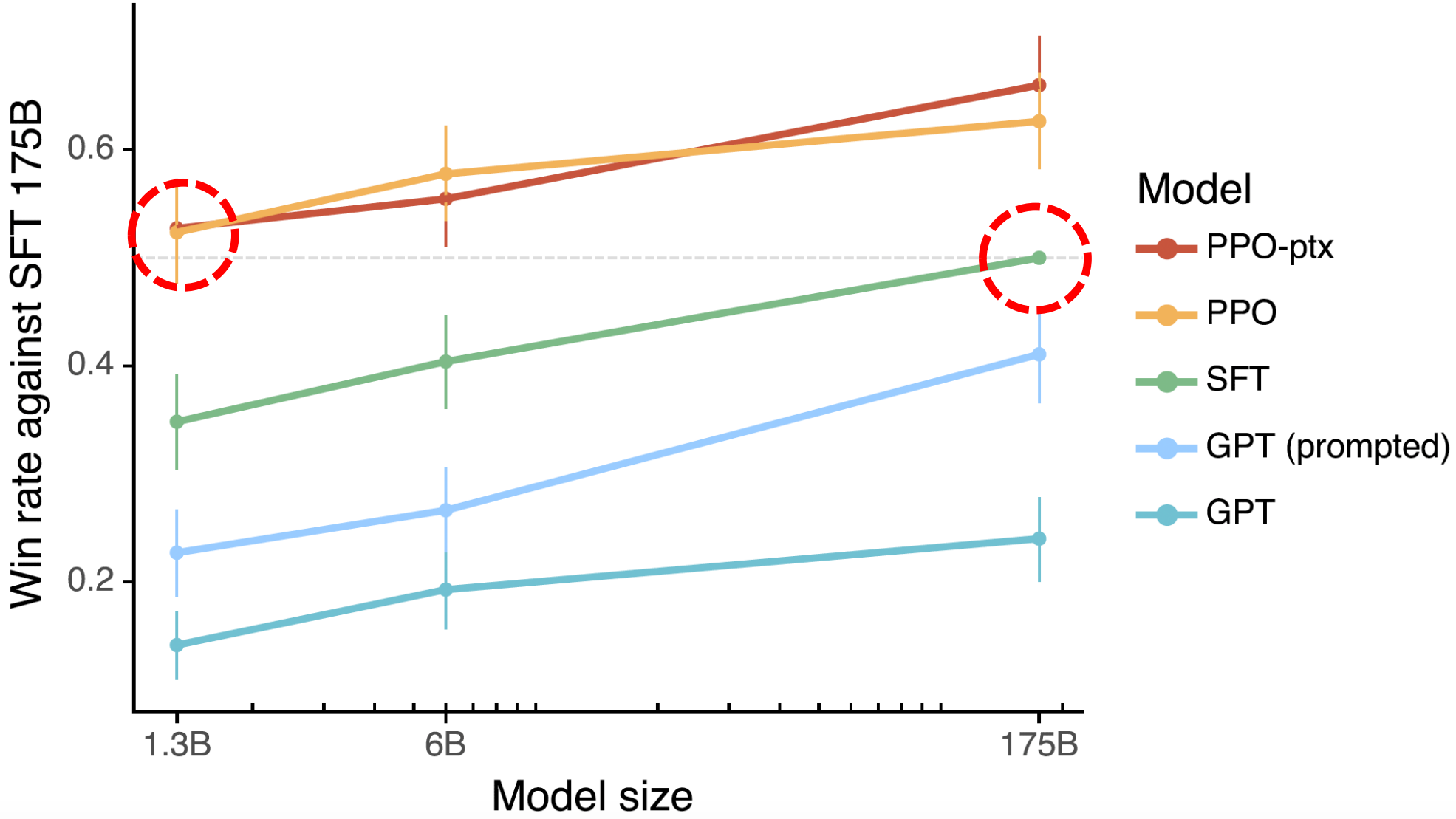
WILLIAM & MARY | DATA SCIENCE



David McCandless, Tom Evans, Paul Barton
Informationisbeautiful // Sep 2025

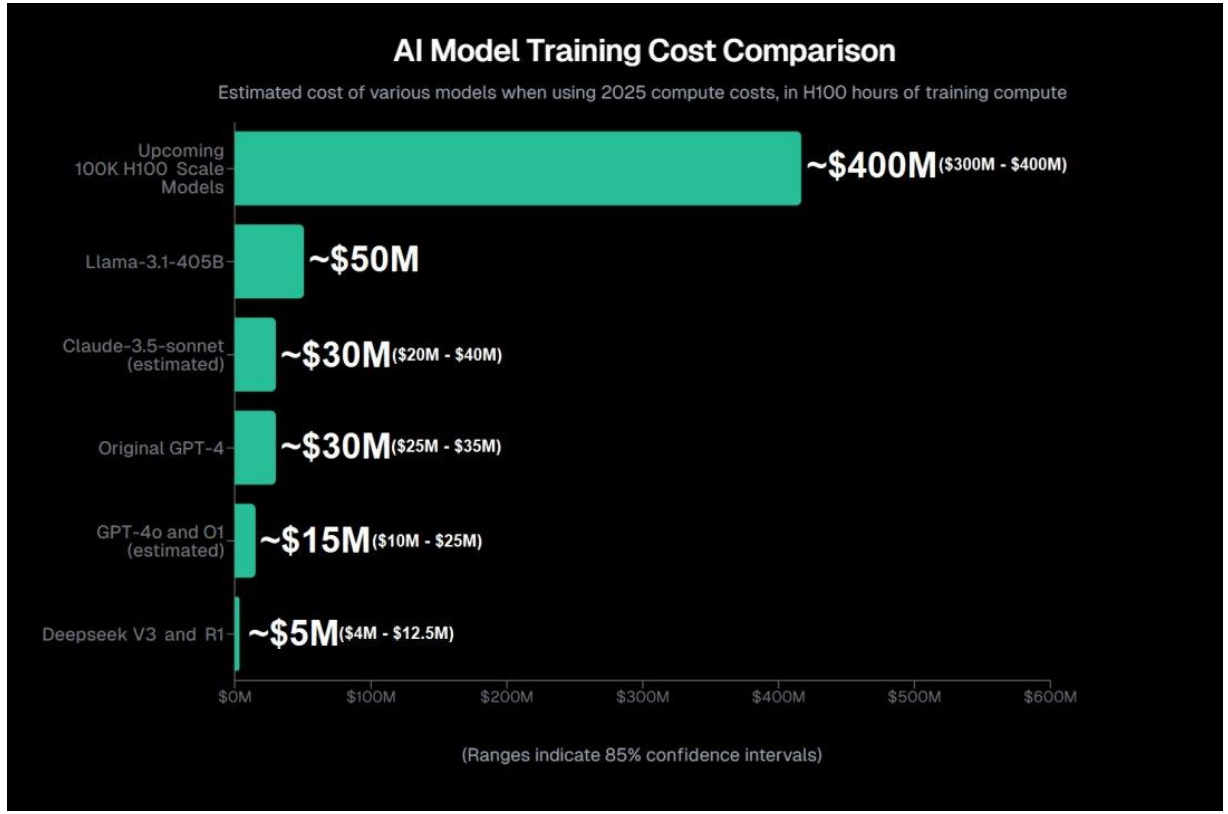
MMLU = benchmark for measuring LLM capabilities
* = parameters undisclosed // source: LifeArchitect // data

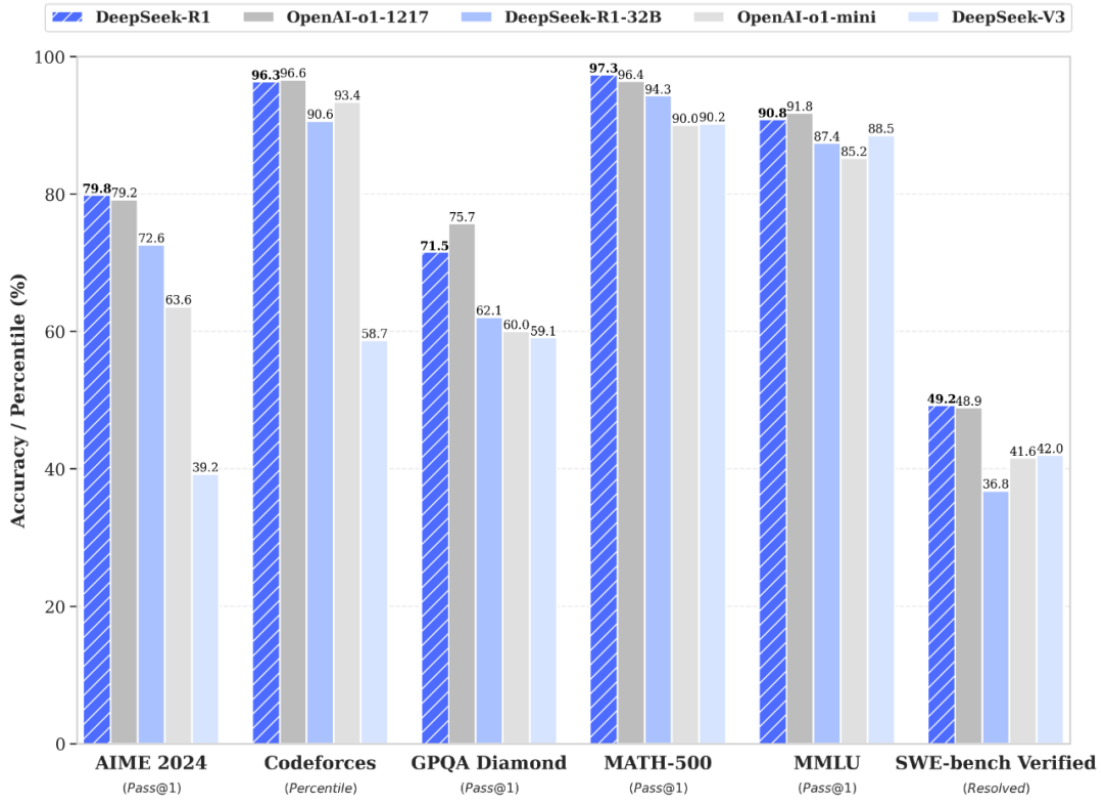
OpenAI InstructGPT (A sibling of ChatGPT)



Ouyang et al. "Training language models to follow instructions with human feedback." *NeurIPS, 2022*.

DeepSeek R1





@arankomatsuzaki

https://api-docs.deepseek.com/news/news250120

Reinforcement learning-based fine-tuning

InstructGPT – Reinforcement Learning from Human Feedback (**RLHF**)

DeepSeek R1 – Reinforcement Learning from Verifiable Reward (**RLVR**)

Ilya Sutskever – We're moving from the age of scaling to the age of research

Dwarkesh Patel
1.16M subscribers

28K | Share | Ask | Save

1,114,742 views Nov 25, 2025 Dwarkesh Podcast

Andrej Karpathy
@karpathy

2025 LLM Year in Review

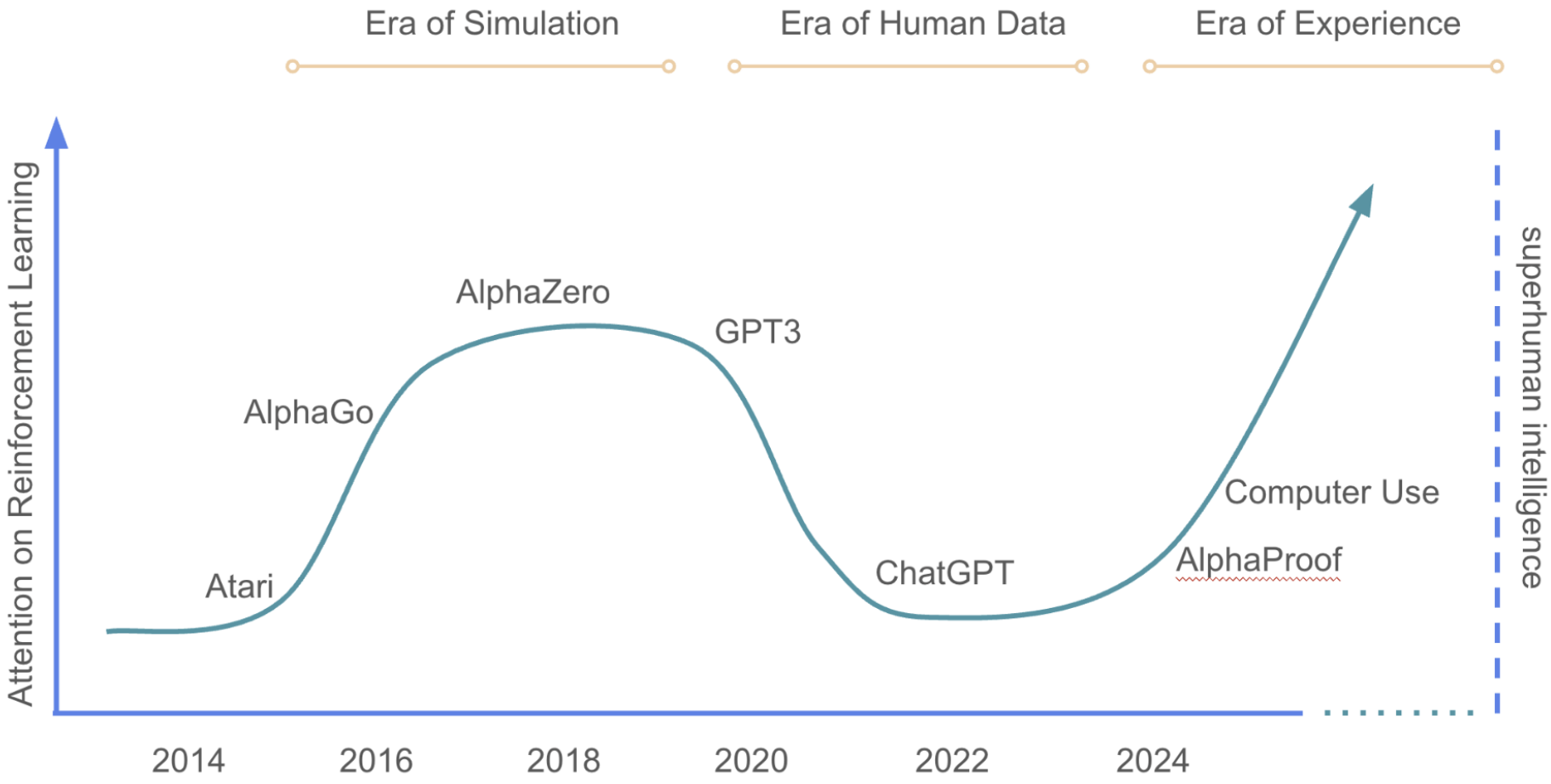
359 3.4K 15K 2.8M

2025 has been a strong and eventful year of progress in LLMs. The following is a list of personally notable and mildly surprising "paradigm changes" - things that altered the landscape and stood out to me conceptually.

1. Reinforcement Learning from Verifiable Rewards (RLVR)

3:45 PM · Dec 19, 2025 · **2.8M** Views

Welcome to the Era of Experience



http://incompleteideas.net/papers/TheEraOfExperience.pdf

W&M DATA SCIENCE

(What I think) RL x LLMs research as of today?

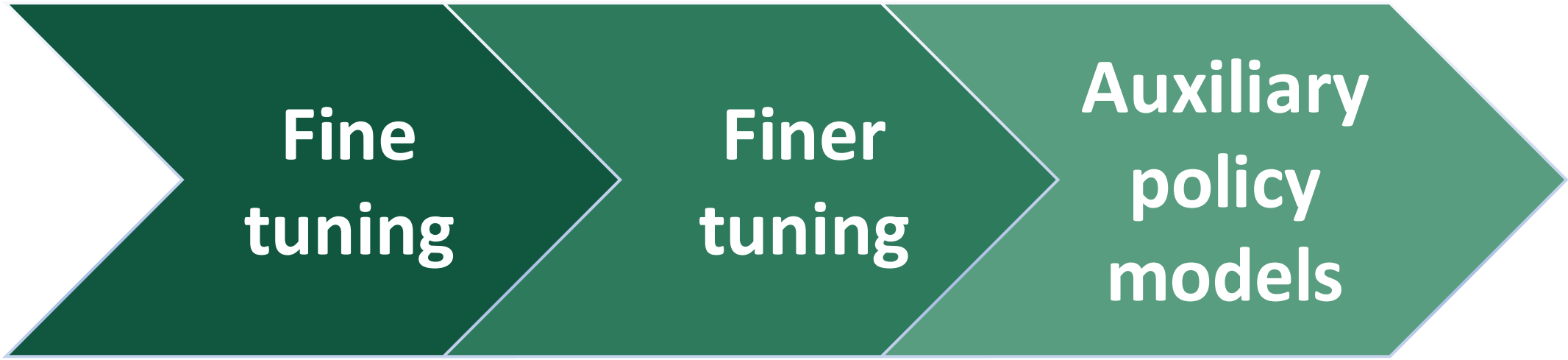
- **RLHF is mature & standard**
  Used mainly for *alignment and helpfulness*, not for discovering new core capabilities
- **RLVR works in limited domains**
  Uses *verifiable / automatic rewards* (math, code, logic) to train reasoning without humans
- **New policy optimization variants help**
  PPO alternatives (e.g., GRPO) improve stability and efficiency
- **Reward function remains a challenge at large**
  Reward design, hacking, and generalization remain the central research challenges
- **Agents & "finer-tuning"**
  RL increasingly used for *tool use, planning, web-agents, …*

(What I think, among others)
A missing piece RL x LLMs research?

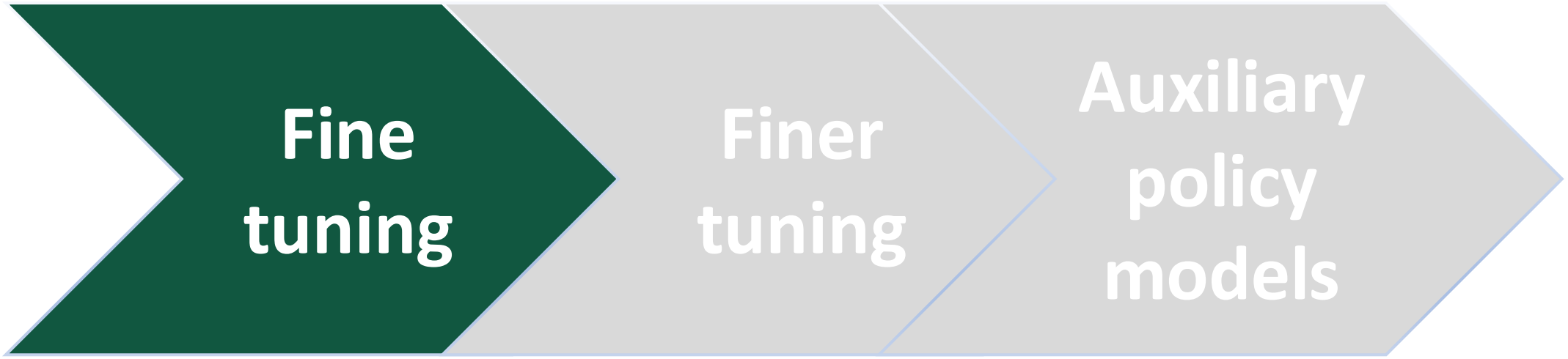
- **Agents & “finer-tuning”**
  RL increasingly used for more LLM down-stream tasks

- **Auxiliary policy models (APMs)**
  Light-weight, separately trained policy models act as steering models for downstream LLM tasks

Outline

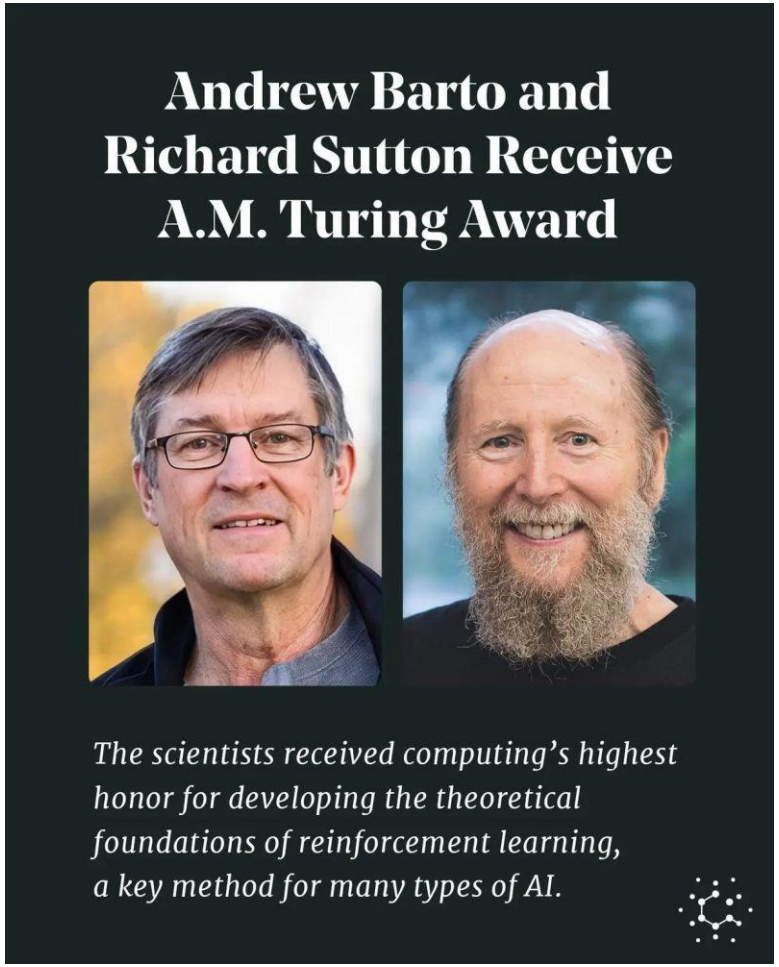


Outline



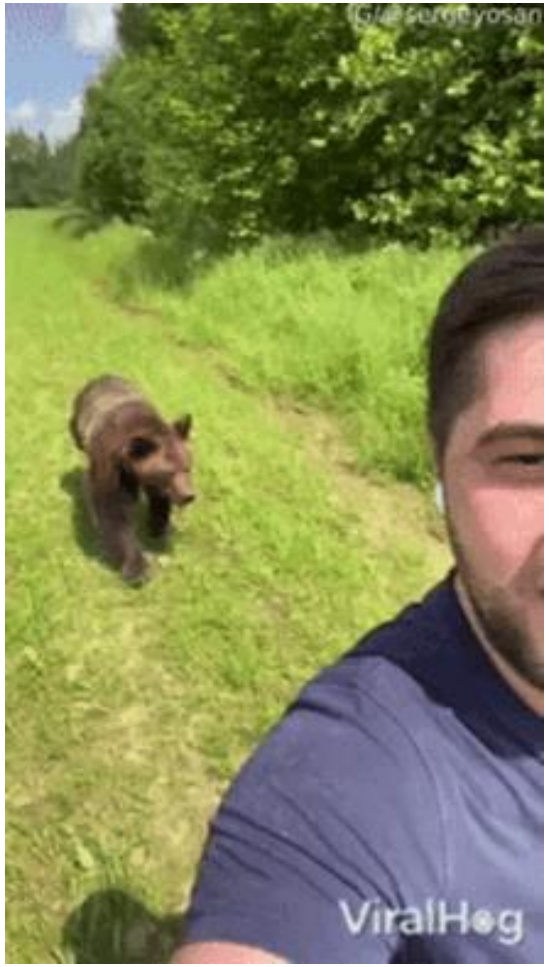
Reinforcement learning

Reinforcement learning (RL) is a type of machine learning where an agent learns how to make decisions by interacting with an environment and receiving feedback in the form of rewards or penalties.



Reinforcement learning in humans

Source



Source

Humans appear to learn behaviors through “trial and error”

The RL problem

Environment

The RL problem

Environment

Agent

The RL problem

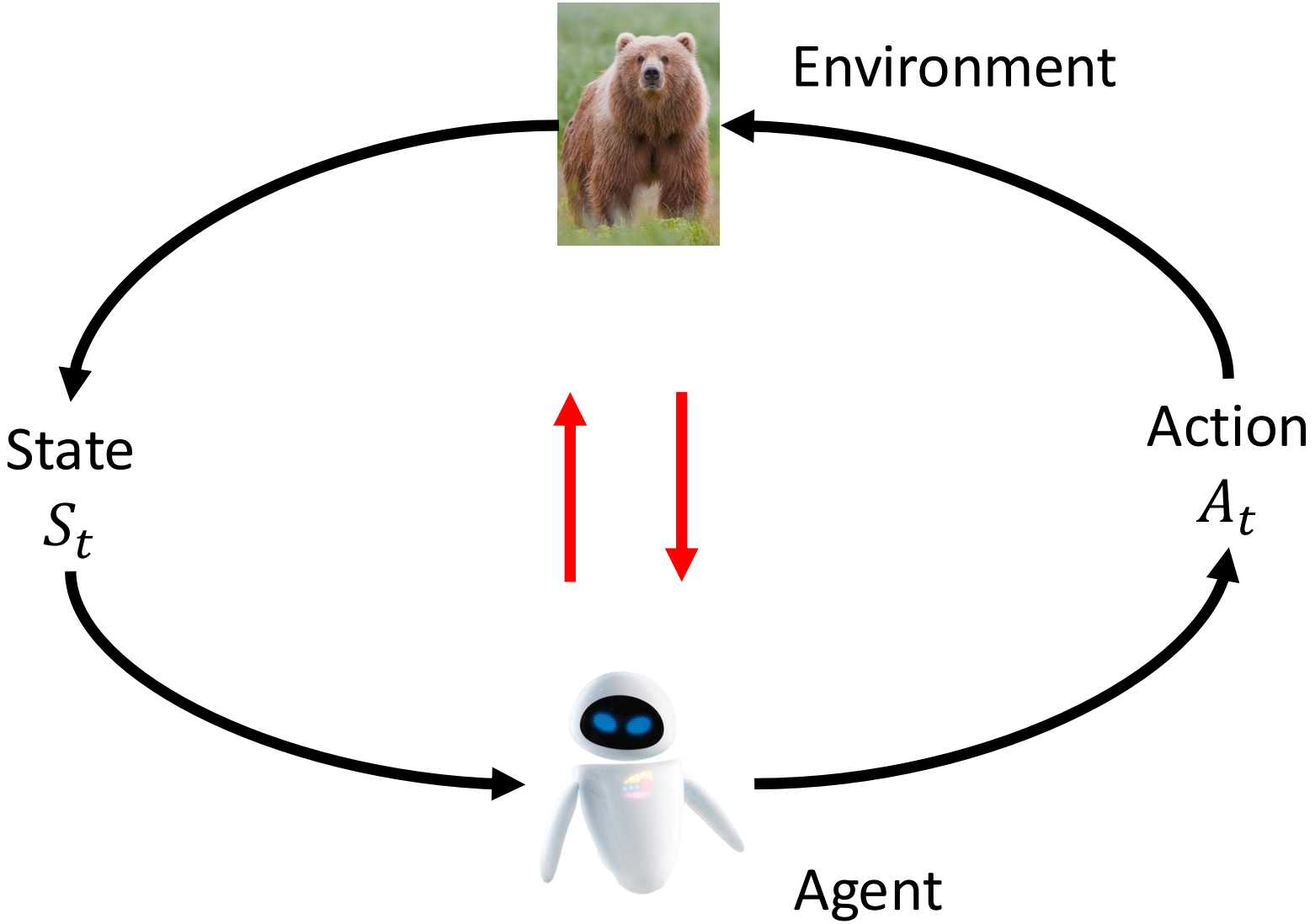
Environment

State

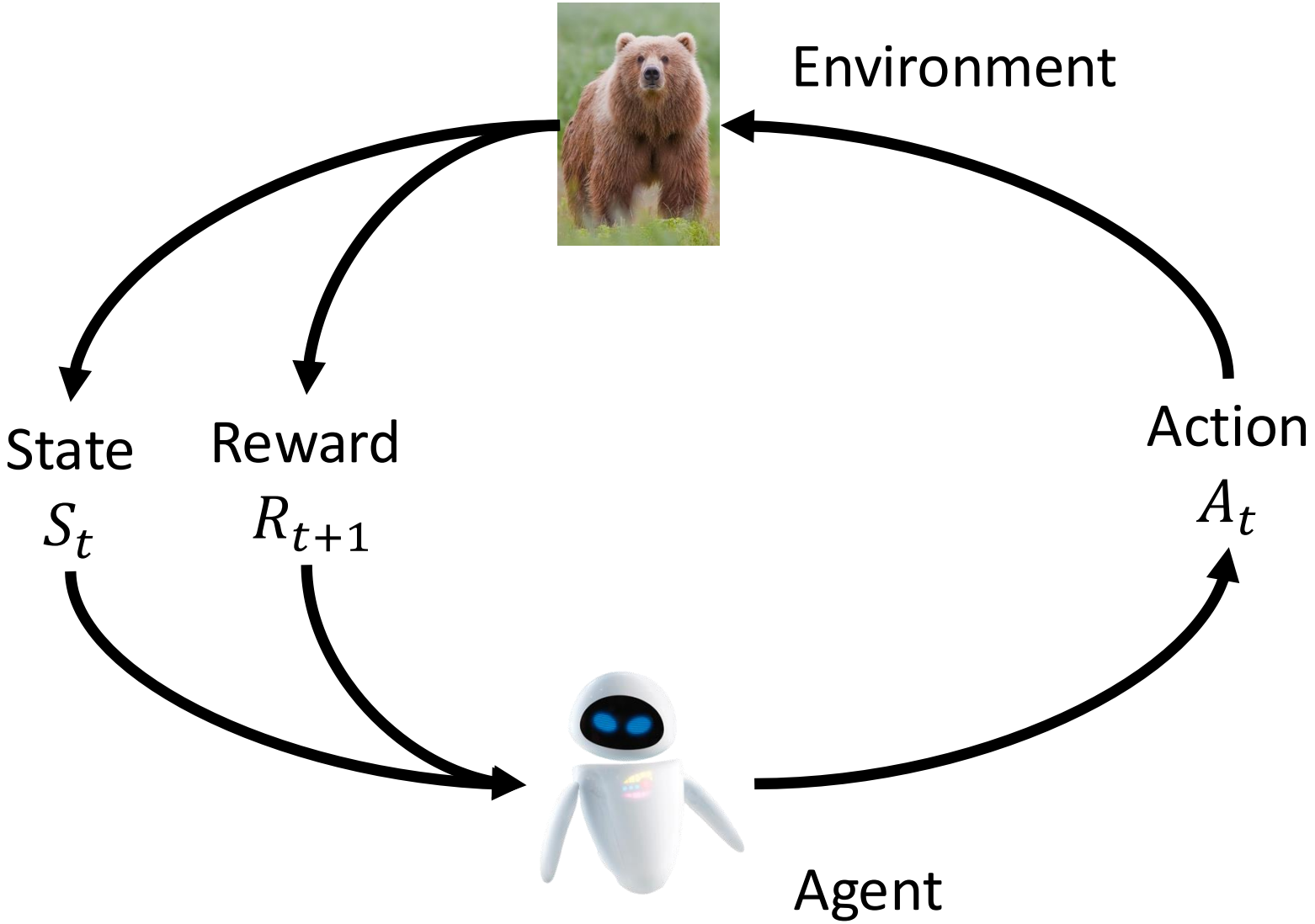
S_t

Agent

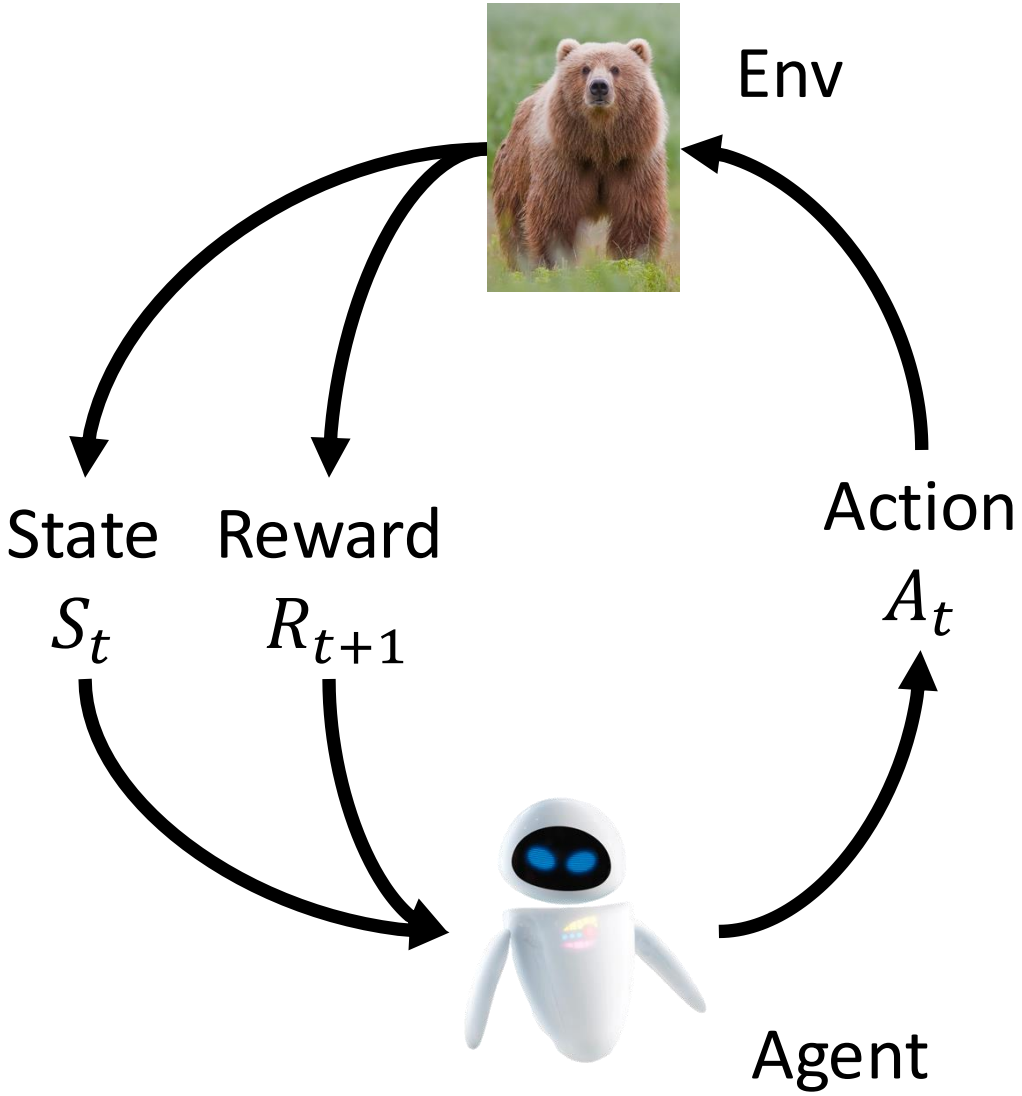
The RL problem



The RL problem

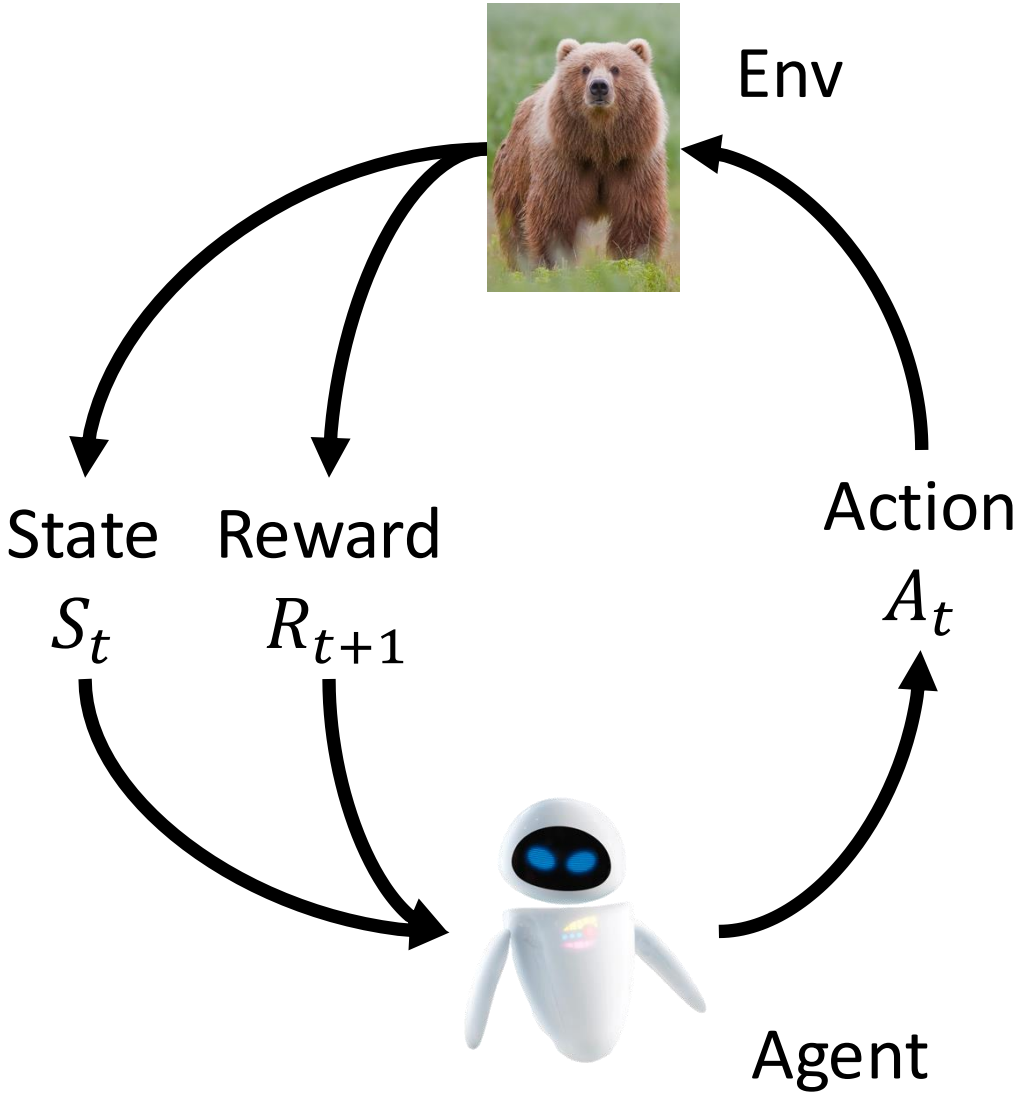


The RL problem



- The agent:
  - Observes state S_t
  - Takes action A_t
  - Receives reward R_t
- The environment:
  - Receives action A_t
  - Emits next state S_{t+1}
  - Emits reward R_{t+1}
- Iteration: $t \rightarrow t+1$

The RL problem: MDP

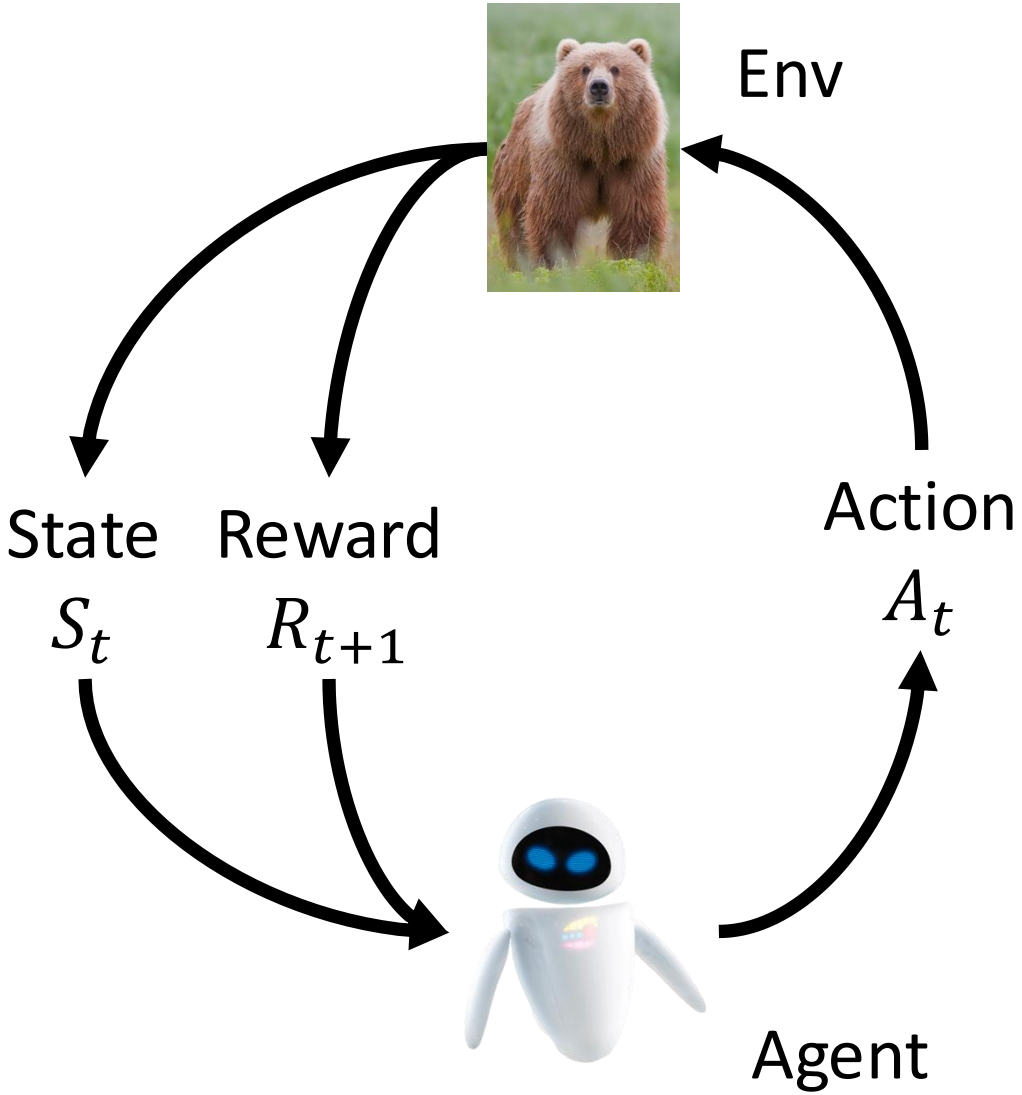


Definition (MDP)

A **Markov Decision Process (MDP)** is a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix
  $\mathcal{P}_{ss'}^{a} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$
- $\mathcal{R}$ is a reward function
  $\mathcal{R}_s^a = \mathbb{E}[\mathcal{R}_{t+1} \mid S_t = s, A_t = a]$
- $\gamma \in [0,1]$ is a discount factor

The RL problem: policy



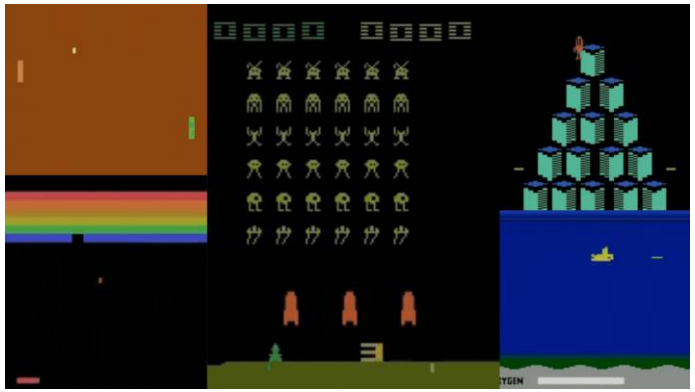
- A **policy** π is a distribution over actions given states,

$$\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$$

- Goal is to maximize the expected return of the policy

$$\pi_* = \arg\max_{\pi} \mathbb{E}_{\pi}[R_{t+1} + \gamma R_{t+2} + \cdots]$$

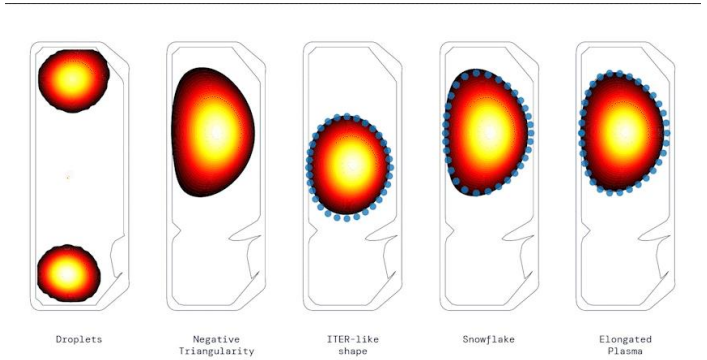
RL before LLMs



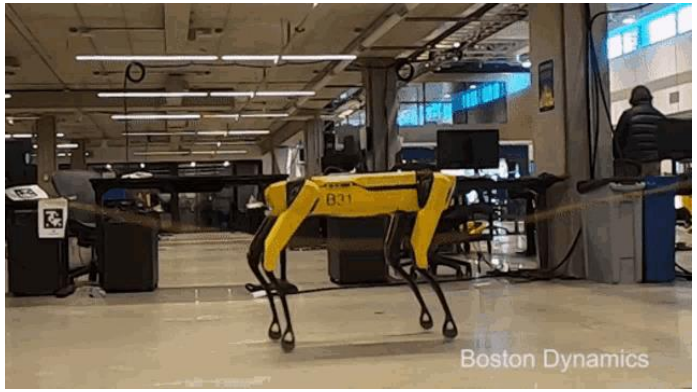
Atari games (DQN)

Computer Go

Data center cooling

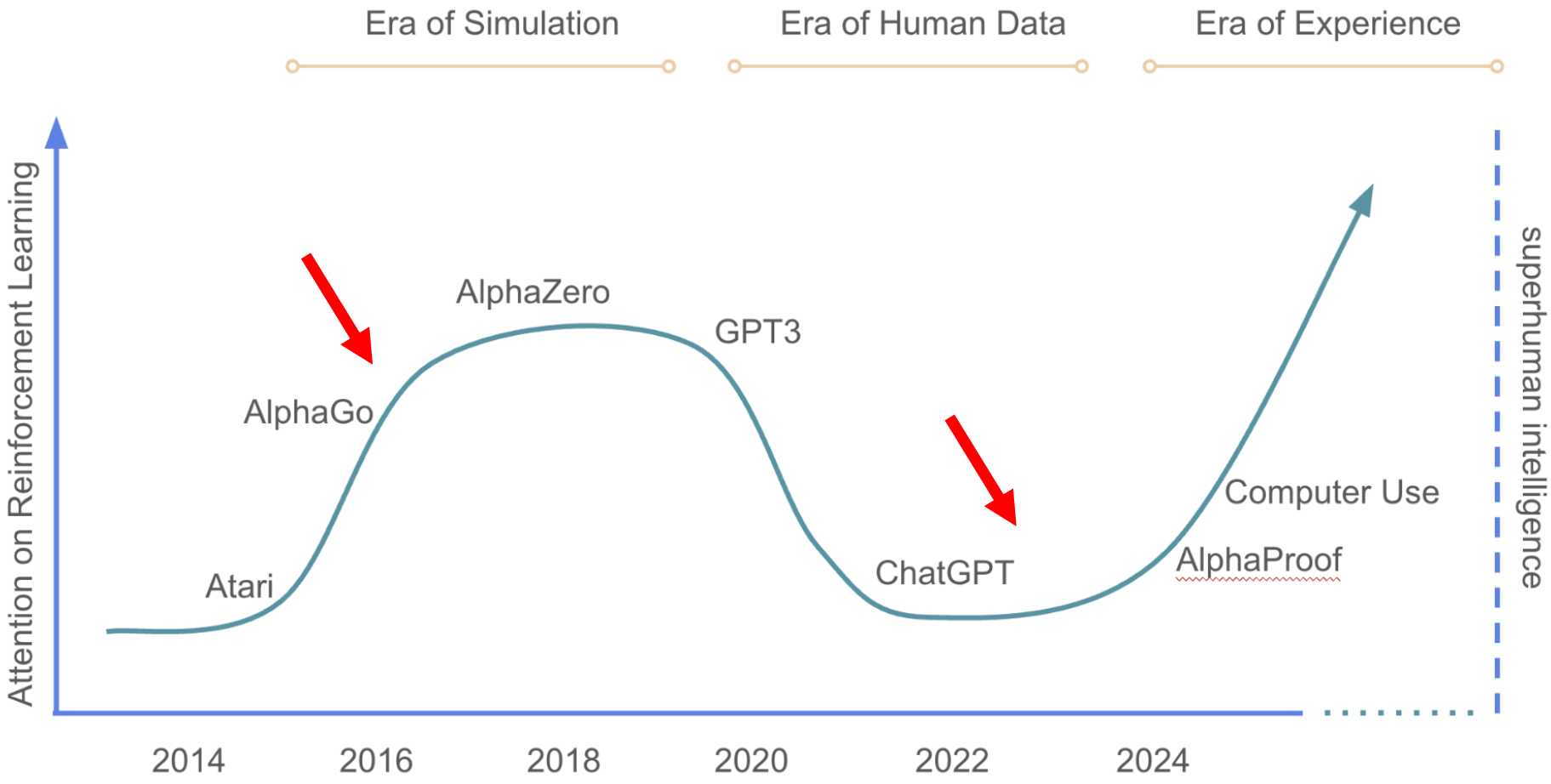


Nuclear fusion control



Robotic control

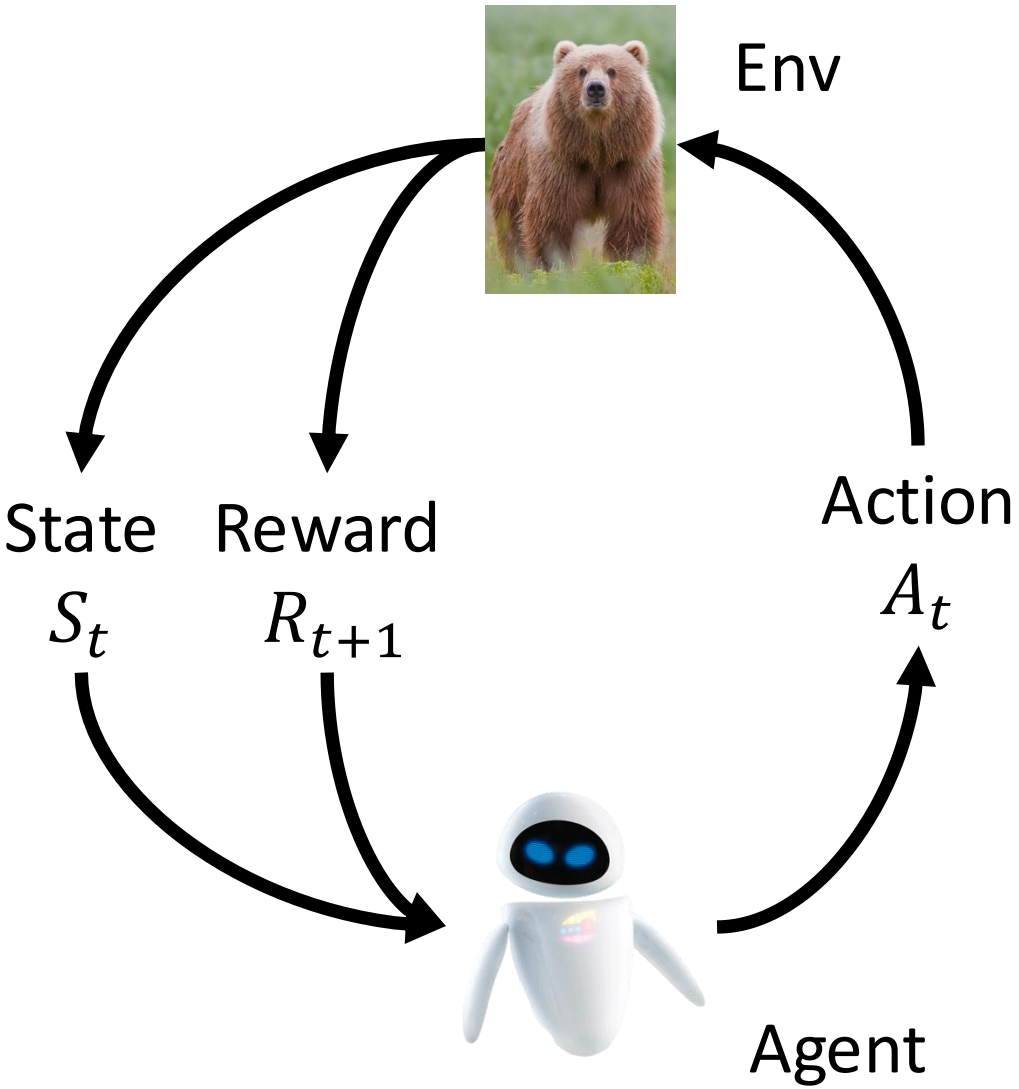
Welcome to the Era of Experience



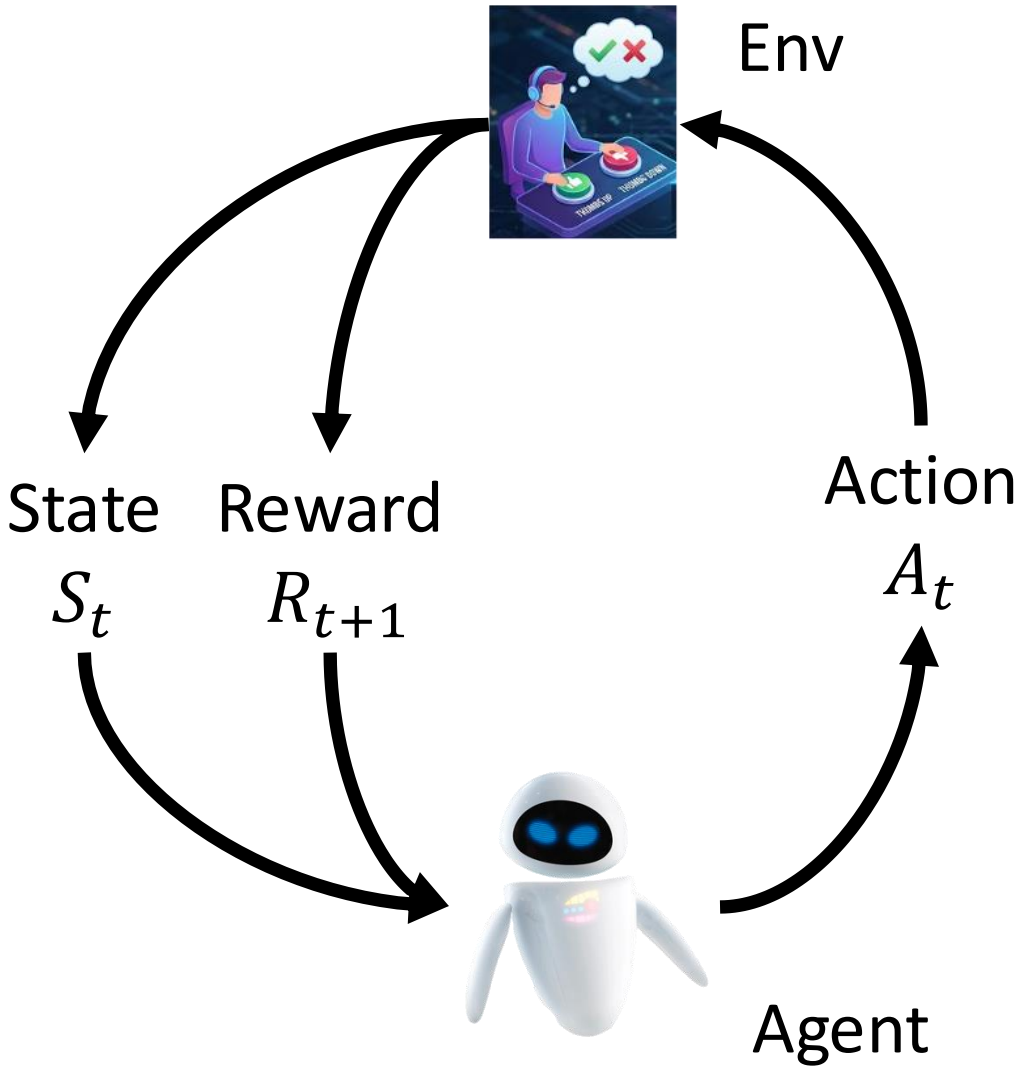
http://incompleteideas.net/papers/TheEraOfExperience.pdf

W&M DATA SCIENCE

Reinforcement Learning



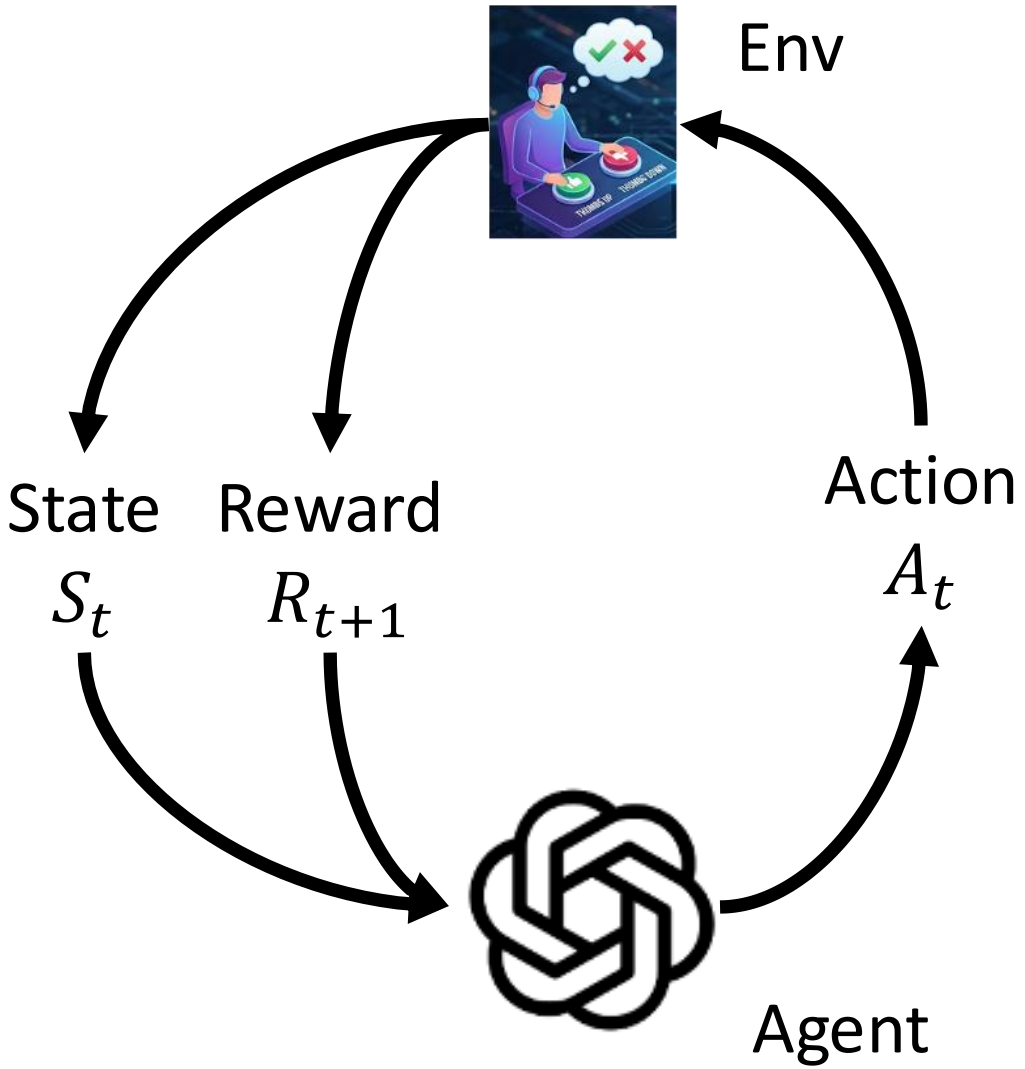
Reinforcement Learning from Human Feedback



- **Env**: humans
- **State:** the user prompt + the tokens generated so far
- **Action:** the generation of a token
- **Reward:** human feedback

Ouyang et al. "Training language models to follow instructions with human feedback." *NeurIPS, 2022*.

Reinforcement Learning from Human Feedback

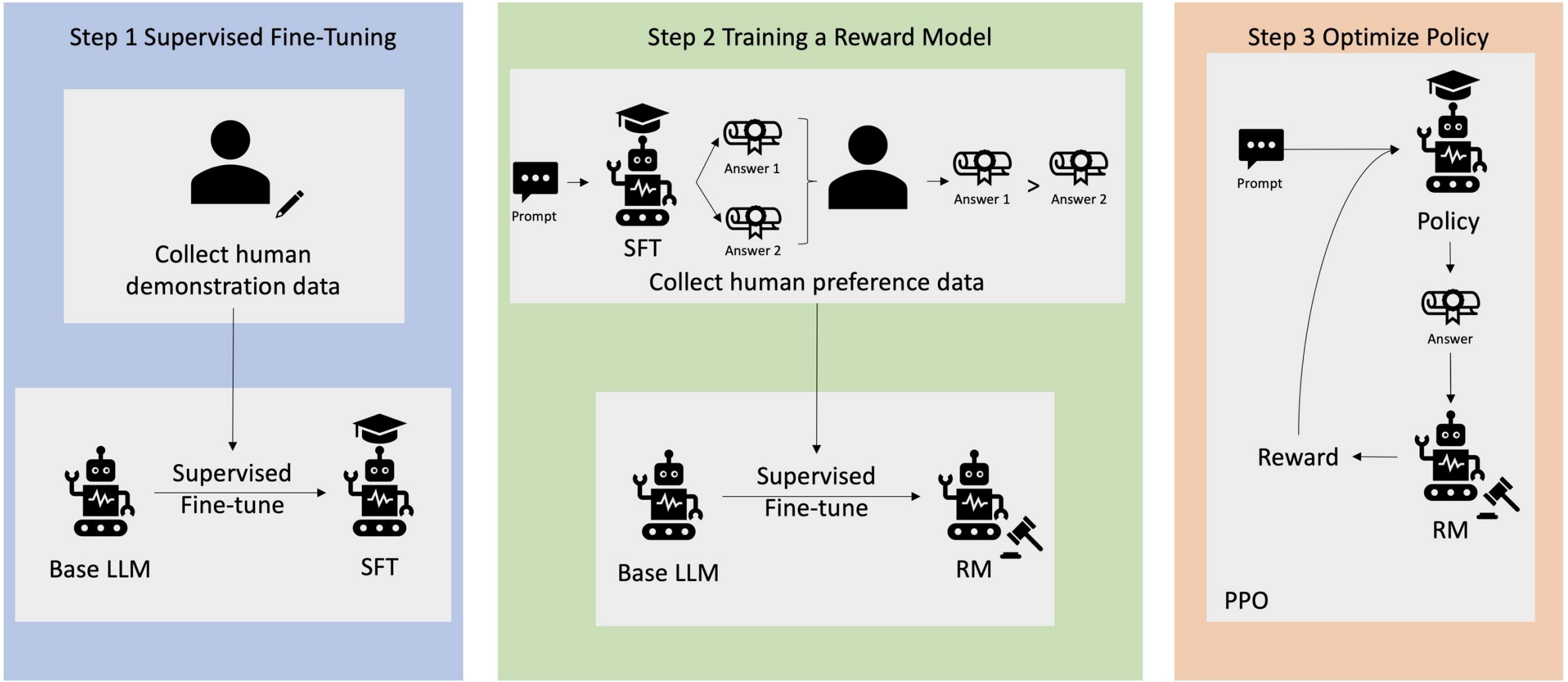


- **Env**: humans
- **State:** the user prompt + the tokens generated so far
- **Action:** the generation of a token
- **Reward:** human feedback

- **Agent**: LLM
- **Policy** π is next token distribution

Ouyang et al. "Training language models to follow instructions with human feedback." *NeurIPS, 2022*.

RLHF in OpenAI's InstructGPT



Source

Proximal policy optimization (PPO)

Algorithm 1 PPO, Actor-Critic Style

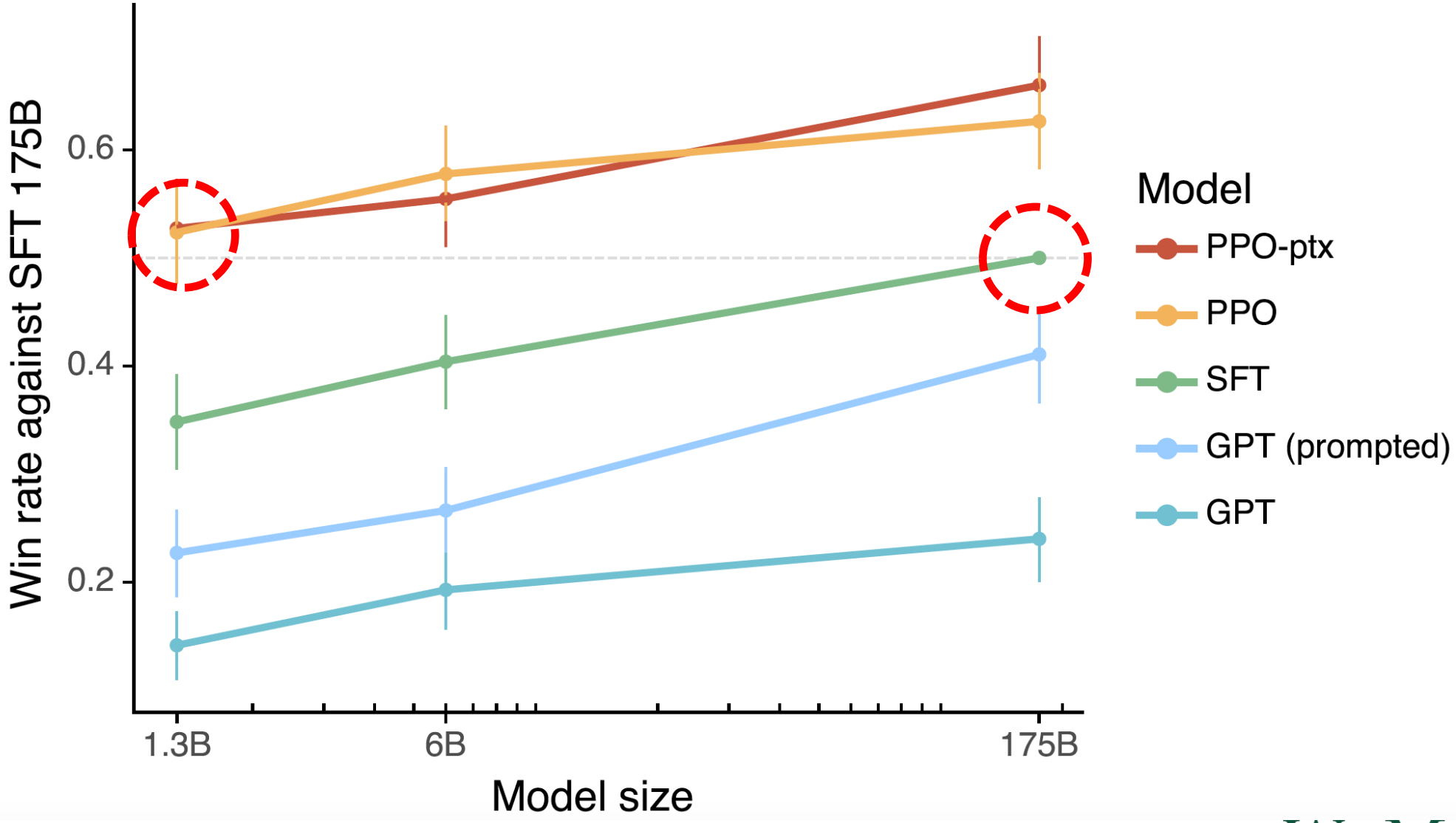
for iteration=$1, 2, \ldots$ **do**
  for actor=$1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

$$L^{\text{PPO}}(\theta) = \mathbb{E}_{(s_t, a_t) \sim \pi_{\theta_{\text{old}}}} \left[\min \left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t,\ \text{clip}\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) \right]$$

$$\hat{A}_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \left(r_{t+l} + \gamma V(s_{t+l+1}) - V(s_{t+l}) \right)$$

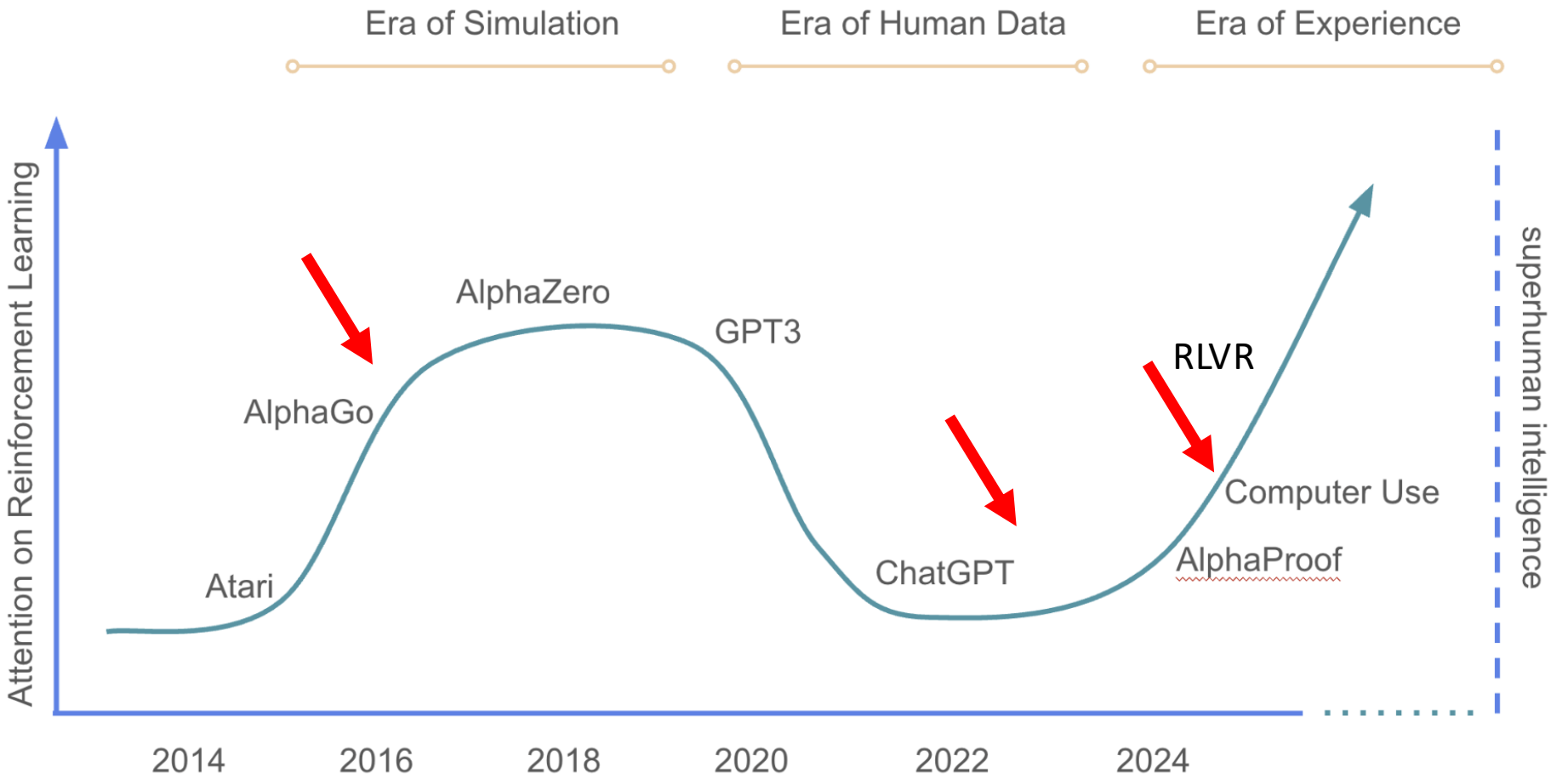
Schulman et al. “Proximal Policy Optimization Algorithms”. Arxiv, 2017

OpenAI InstructGPT



Ouyang et al. "Training language models to follow instructions with human feedback." *NeurIPS, 2022*.

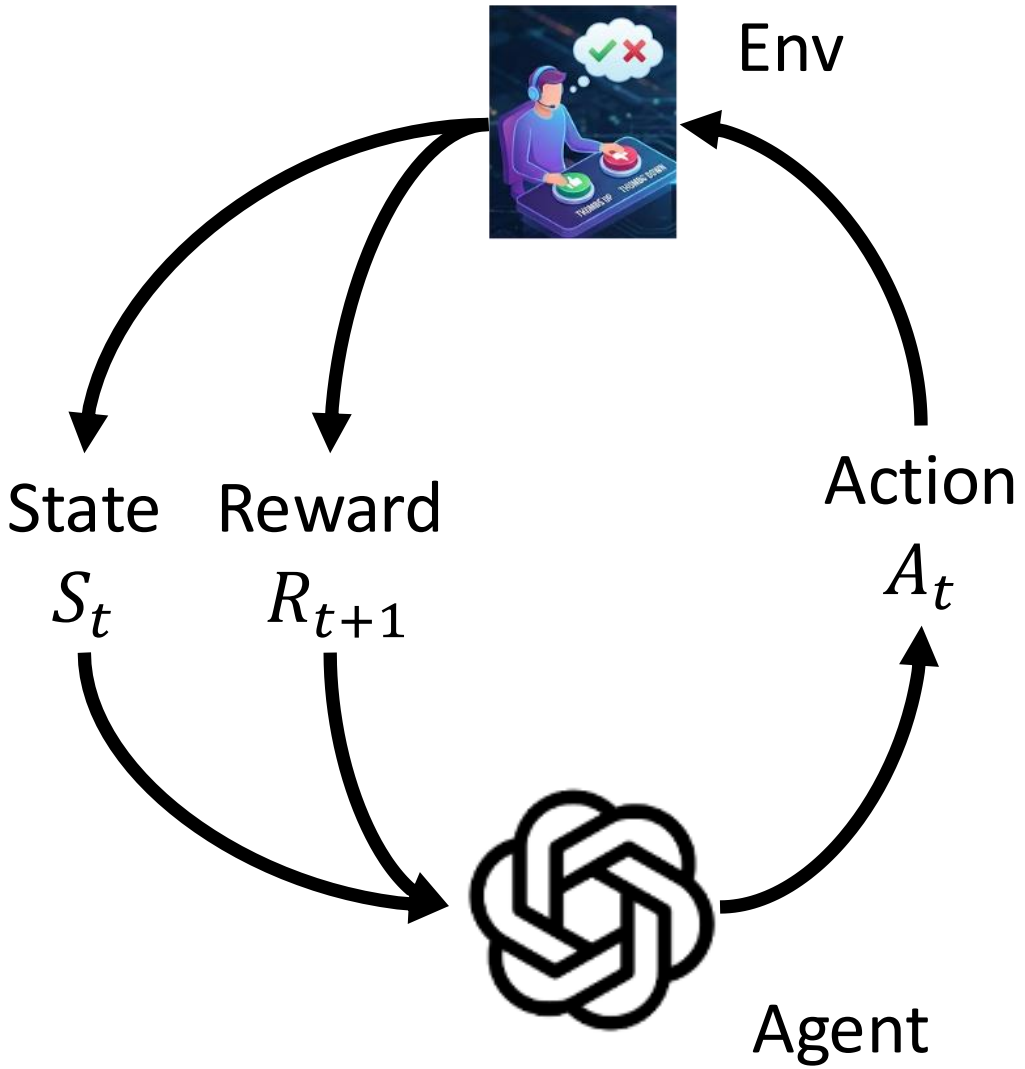
Welcome to the Era of Experience



http://incompleteideas.net/papers/TheEraOfExperience.pdf

W&M DATA SCIENCE

Reinforcement Learning from Human Feedback

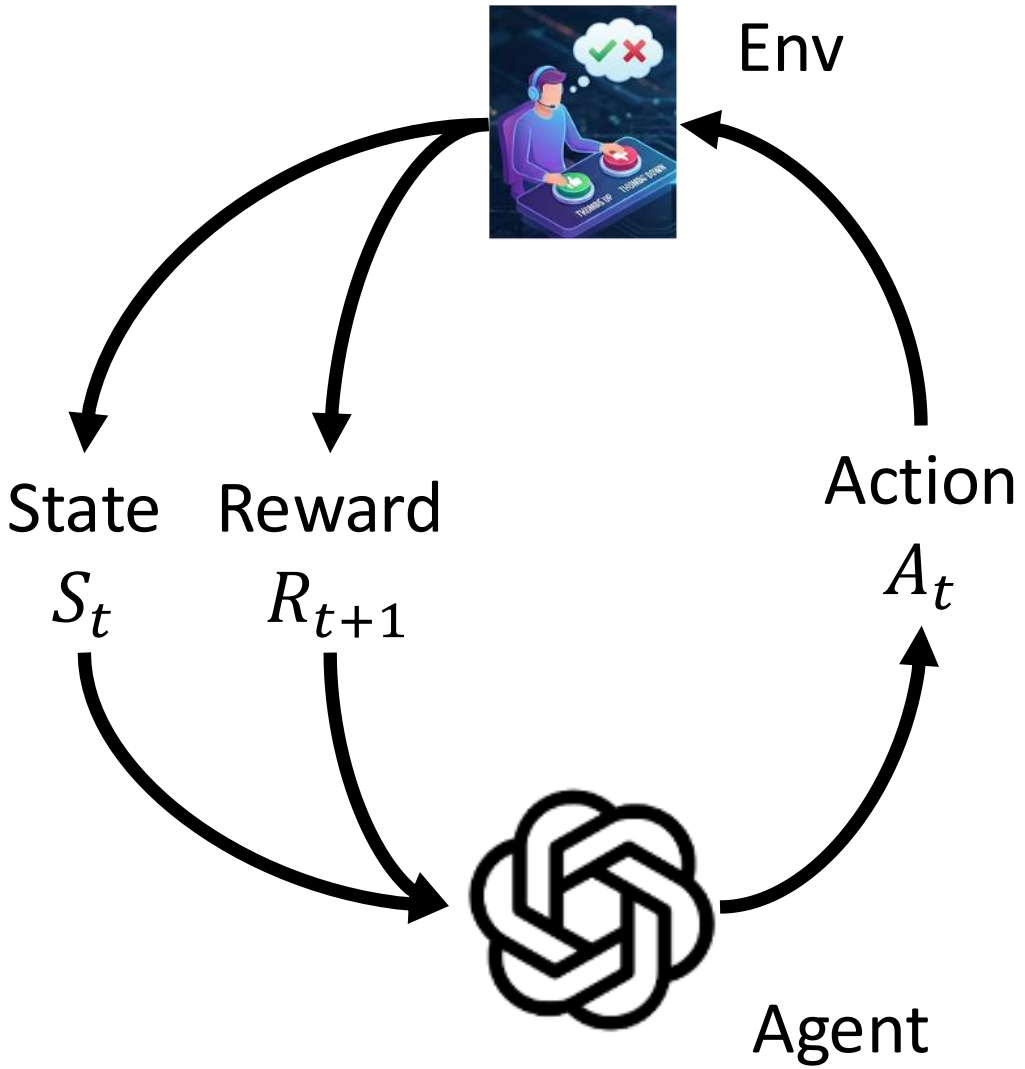


- **Env**: humans
- **State:** the user prompt + the tokens generated so far
- **Action:** the generation of a token
- **Reward:** human feedback

- **Agent**: LLM
- **Policy** π is next token distribution

Guo et al. "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning." *Nature,* 2025.

Reinforcement Learning from ~~Human Feedback~~ Experience

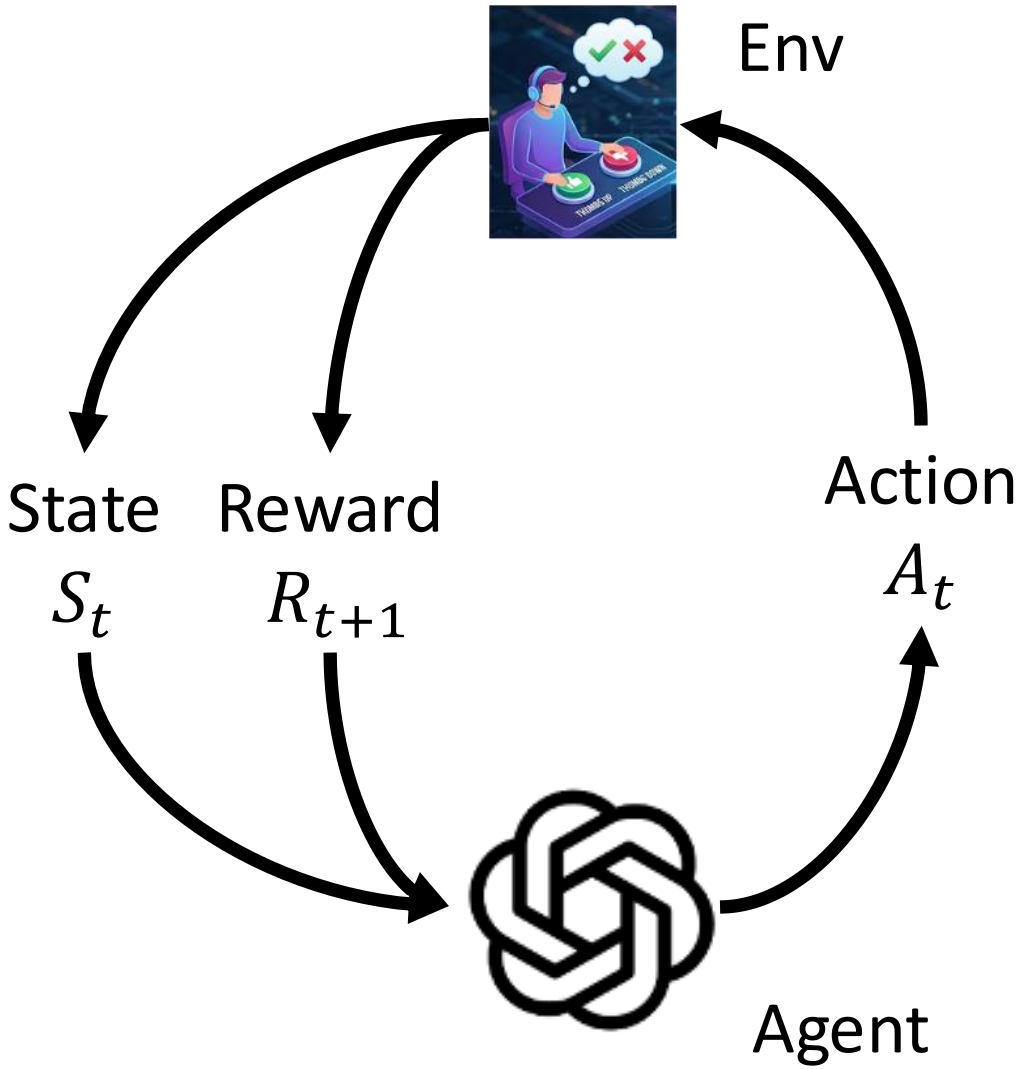


- **Env**: humans
- **State:** the user prompt + the tokens generated so far
- **Action:** the generation of a token
- **Reward:** human feedback

- **Agent**: LLM
- **Policy** π is next token distribution

Guo et al. "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning." *Nature,* 2025.

Reinforcement Learning from ~~Experience~~ Verifiable Reward

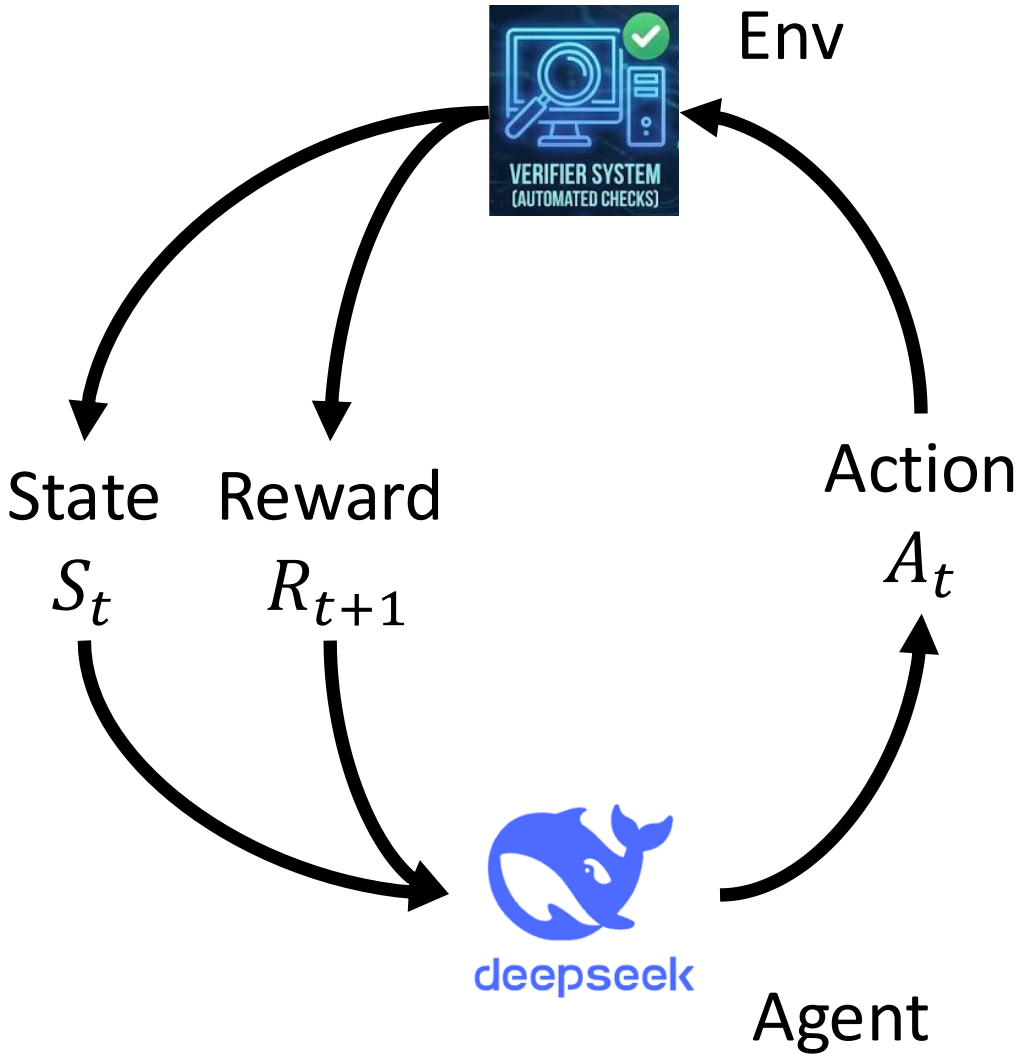


- **Env**: humans
- **State:** the user prompt + the tokens generated so far
- **Action:** the generation of a token
- **Reward:** human feedback

- **Agent**: LLM
- **Policy** π is next token distribution

Guo et al. "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning." *Nature,* 2025.

Reinforcement Learning from Verifiable Reward

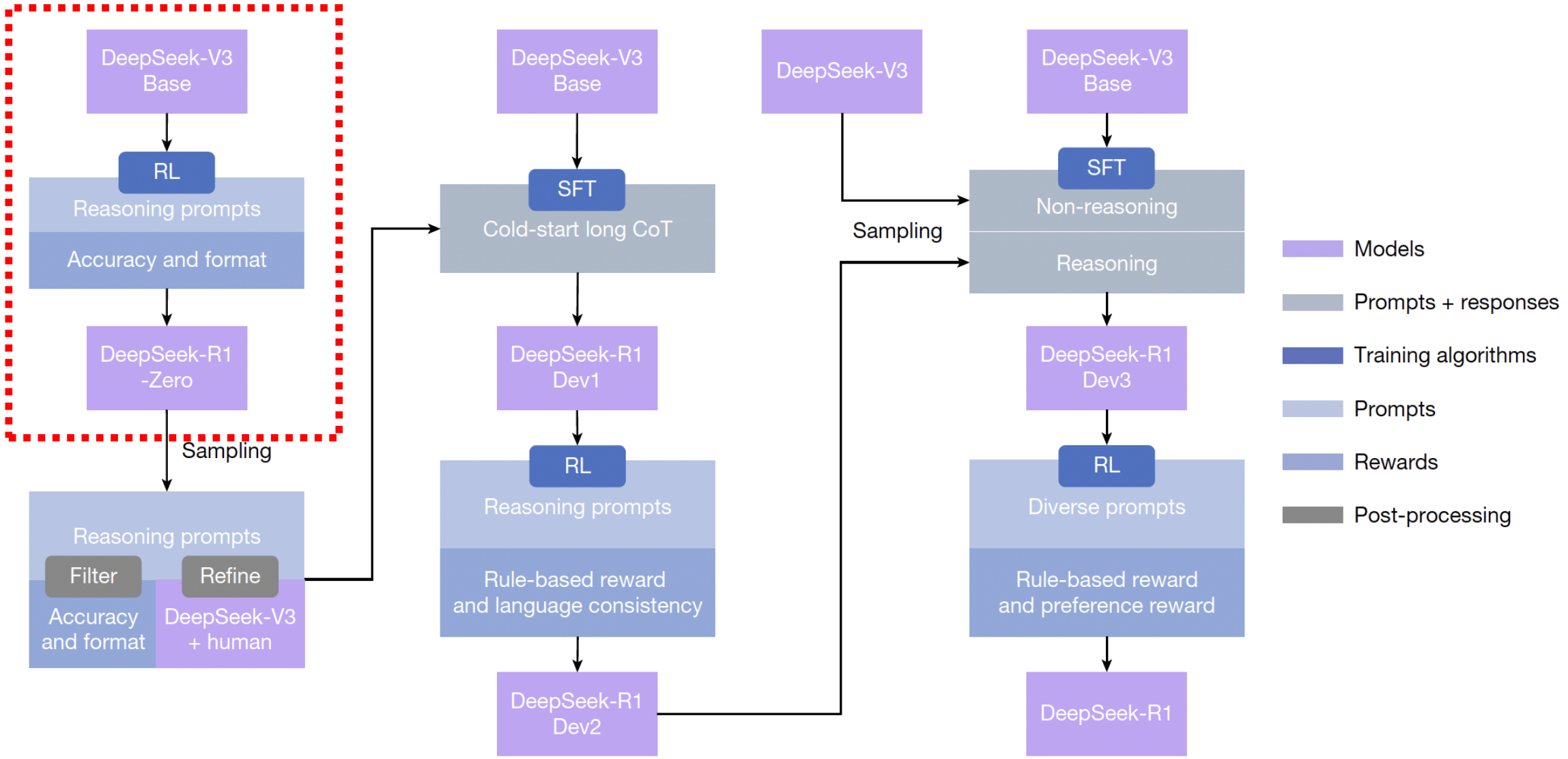


- **Env**: ~~humans~~ verifier system
- **State:** user prompt + tokens generated so far
- **Action:** the generation of a token in a *reasoning chain*
- **Reward:** ~~human~~ verifier feedback

- **Agent**: LLM
- **Policy** π is next token distribution

Guo et al. "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning." *Nature,* 2025.

DeepSeek-R1 training pipeline



Guo et al. "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning." *Nature,* 2025.

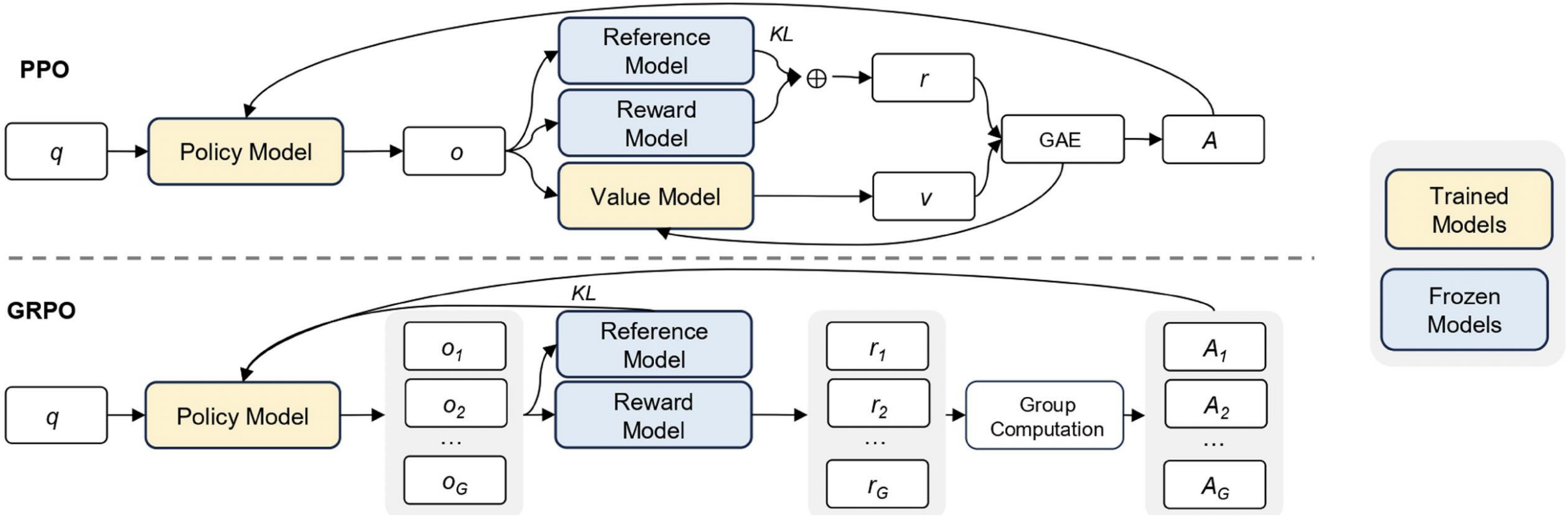
W&M DATA SCIENCE

GRPO algorithm

Algorithm 1 GRPO

for iteration=$1, 2, \ldots$ **do**
  Sample batch of prompts Q
  for each prompt $q \in Q$ **do**
    Sample group of G outputs using $\pi_{\theta_{\text{old}}}$
    Compute rewards and group-normalized advantages $A_1, \ldots, A_G$
  end for
  Optimize policy surrogate L wrt θ, with K epochs and minibatch size M
  $\theta_{\text{old}} \leftarrow \theta$
end for

Group Relative Policy Optimization (GRPO)



Guo et al. "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning." *Nature,* 2025.

GRPO loss

$$L^{\text{PPO}}(\theta) = \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}}\left[\min\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t,\ \text{clip}\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right]$$

$$\hat{A}_t = \sum_{l=0}^{T-t-1}(\gamma\lambda)^l\left(r_{t+l} + \gamma V(s_{t+l+1}) - V(s_{t+l})\right)$$

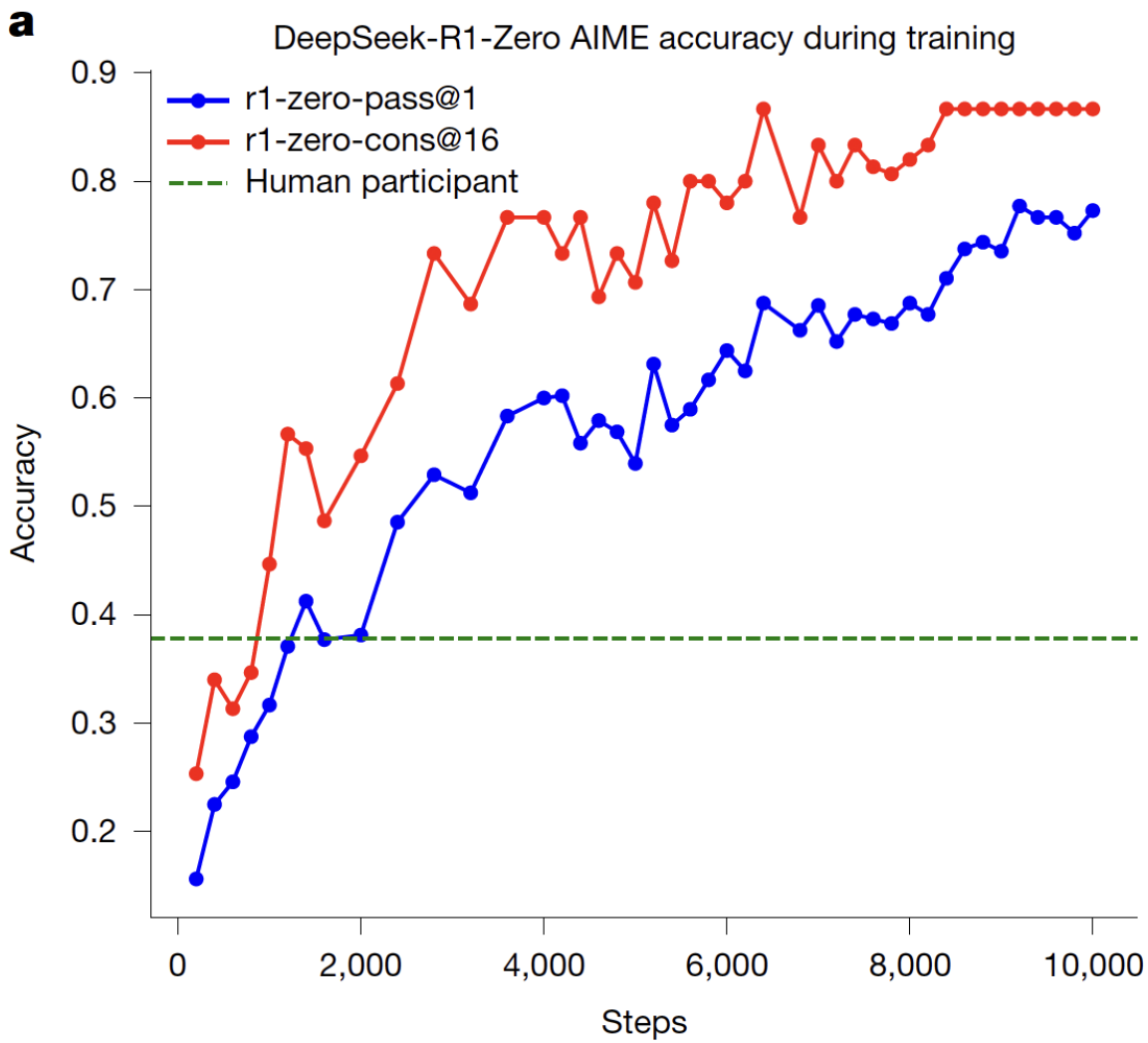
$$L^{\text{GRPO}}(\theta) = \mathbb{E}_s\left[\frac{1}{K}\sum_{i=1}^{K}\min\left(\frac{\pi_\theta(a_i \mid s)}{\pi_{\theta_{\text{old}}}(a_i \mid s)}\hat{A}_i,\ \text{clip}\left(\frac{\pi_\theta(a_i \mid s)}{\pi_{\theta_{\text{old}}}(a_i \mid s)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_i\right)\right]$$

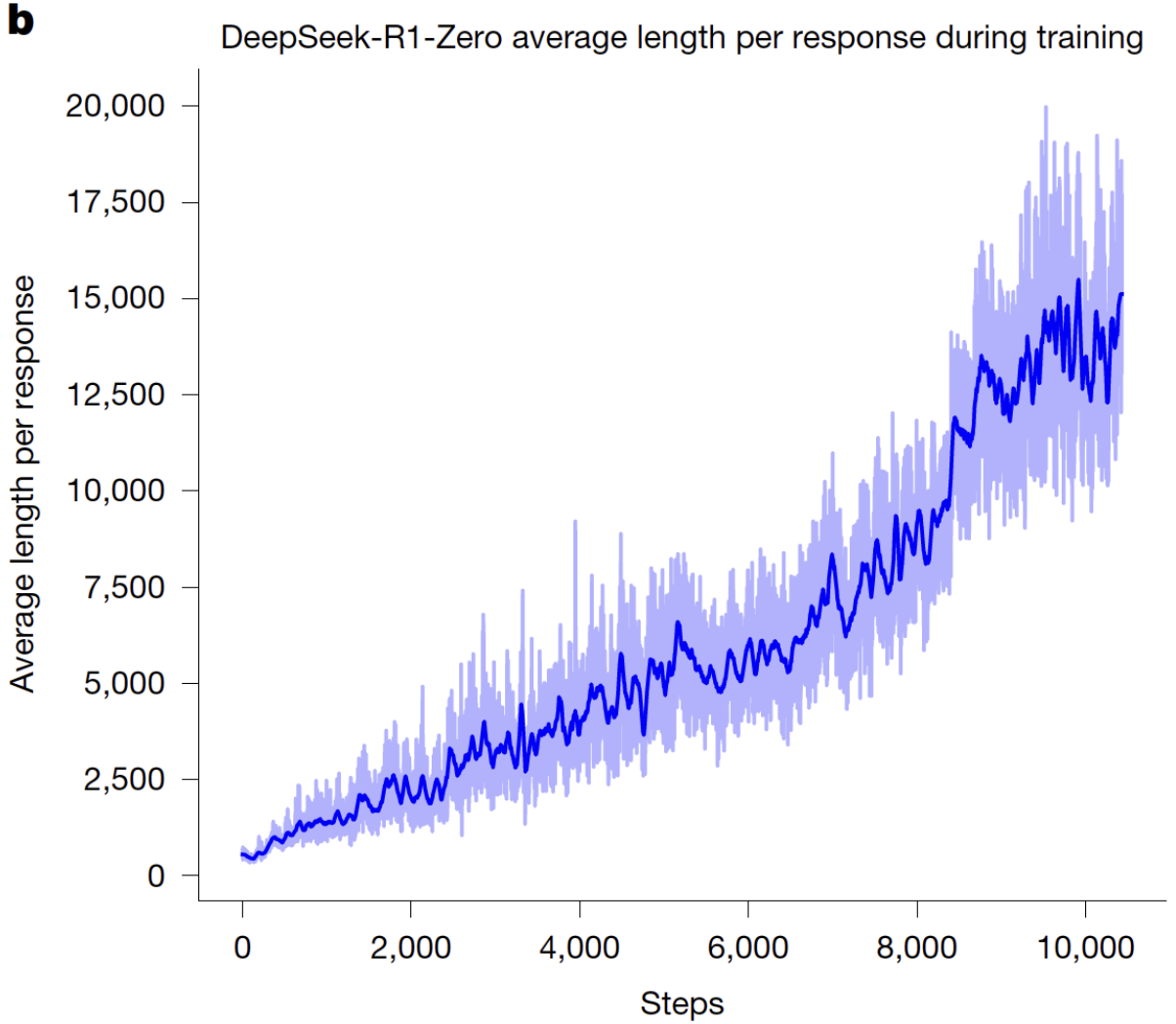
$$\hat{A}_i = r_i - \frac{1}{K}\sum_{j=1}^{K} r_j$$

👍 GRPO does not rely on a value function

Guo et al. "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning." *Nature,* 2025.

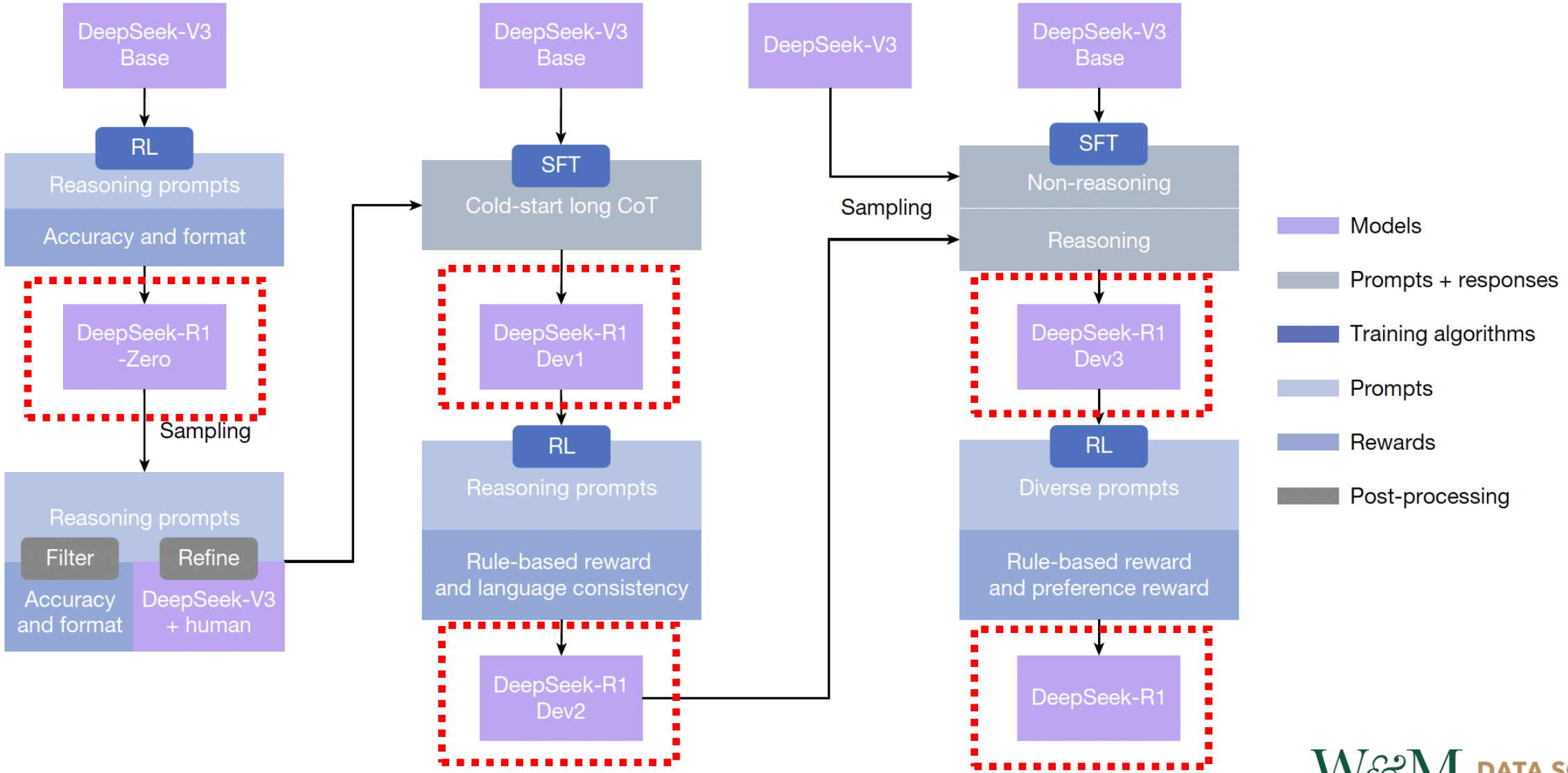
DeepSeek-R1 Zero





W&M DATA SCIENCE

DeepSeek-R1 training pipeline



Guo et al. "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning." *Nature,* 2025.

W&M DATA SCIENCE

DeepSeek-R1

Table 2 | Experimental results at each stage of DeepSeek-R1

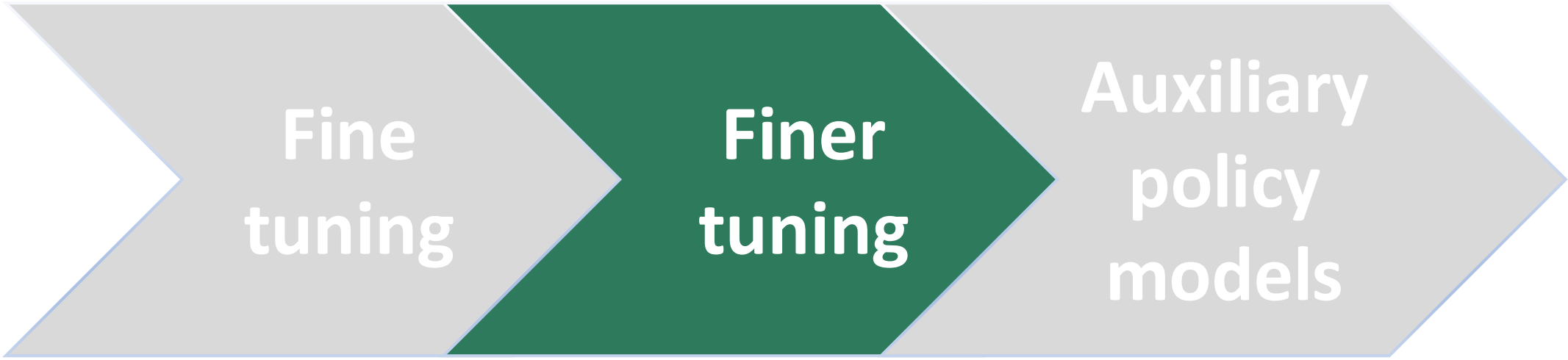
	Benchmark (metric)	R1-Zero	R1Dev1	R1Dev2	R1Dev3	R1
English	MMLU (EM)	88.8	89.1	**91.2**	91.0	90.8
	MMLU-Redux (EM)	85.6	90.0	93.0	93.1	92.9
	MMLU-Pro (EM)	68.9	74.1	83.8	83.1	**84.0**
	DROP (3-shot F1)	89.1	89.8	91.1	88.7	**92.2**
	IF-Eval (Prompt Strict)	46.6	71.7	72.0	78.1	**83.3**
	GPQA Diamond (Pass@1)	**75.8**	66.1	70.7	71.2	71.5
	SimpleQA (Correct)	30.3	17.8	28.2	24.9	30.1
	FRAMES (Acc.)	82.3	78.5	81.8	81.9	**82.5**
	AlpacaEval2.0 (LC-winrate)	24.7	50.1	55.8	62.1	**87.6**
	Arena-Hard (GPT-4-1106)	53.6	77.0	73.2	75.6	**92.3**
Code	LiveCodeBench (Pass@1-COT)	50.0	57.5	63.5	64.6	**65.9**
	Codeforces (Percentile)	80.4	84.5	90.5	92.1	**96.3**
	Codeforces (Rating)	1,444	1,534	1,687	1,746	**2,029**
	SWE-bench Verified (Resolved)	43.2	39.6	44.6	45.6	**49.2**
	Aider-Polyglot (Acc.)	12.2	6.7	25.6	44.8	**53.3**
Maths	AIME2024 (Pass@1)	77.9	59.0	74.0	78.1	**79.8**
	MATH-500 (Pass@1)	95.9	94.2	95.9	95.4	**97.3**
	CNMO2024 (Pass@1)	**88.1**	58.0	73.9	77.3	78.8
Chinese	CLUEWSC (EM)	93.1	92.8	92.6	91.6	92.8
	C-Eval (EM)	**92.8**	85.7	91.9	86.4	91.8
	C-SimpleQA (Correct)	66.4	58.8	64.2	66.9	63.7

Numbers in bold denote that the performance is statistically significant (t-test with $P<0.01$).

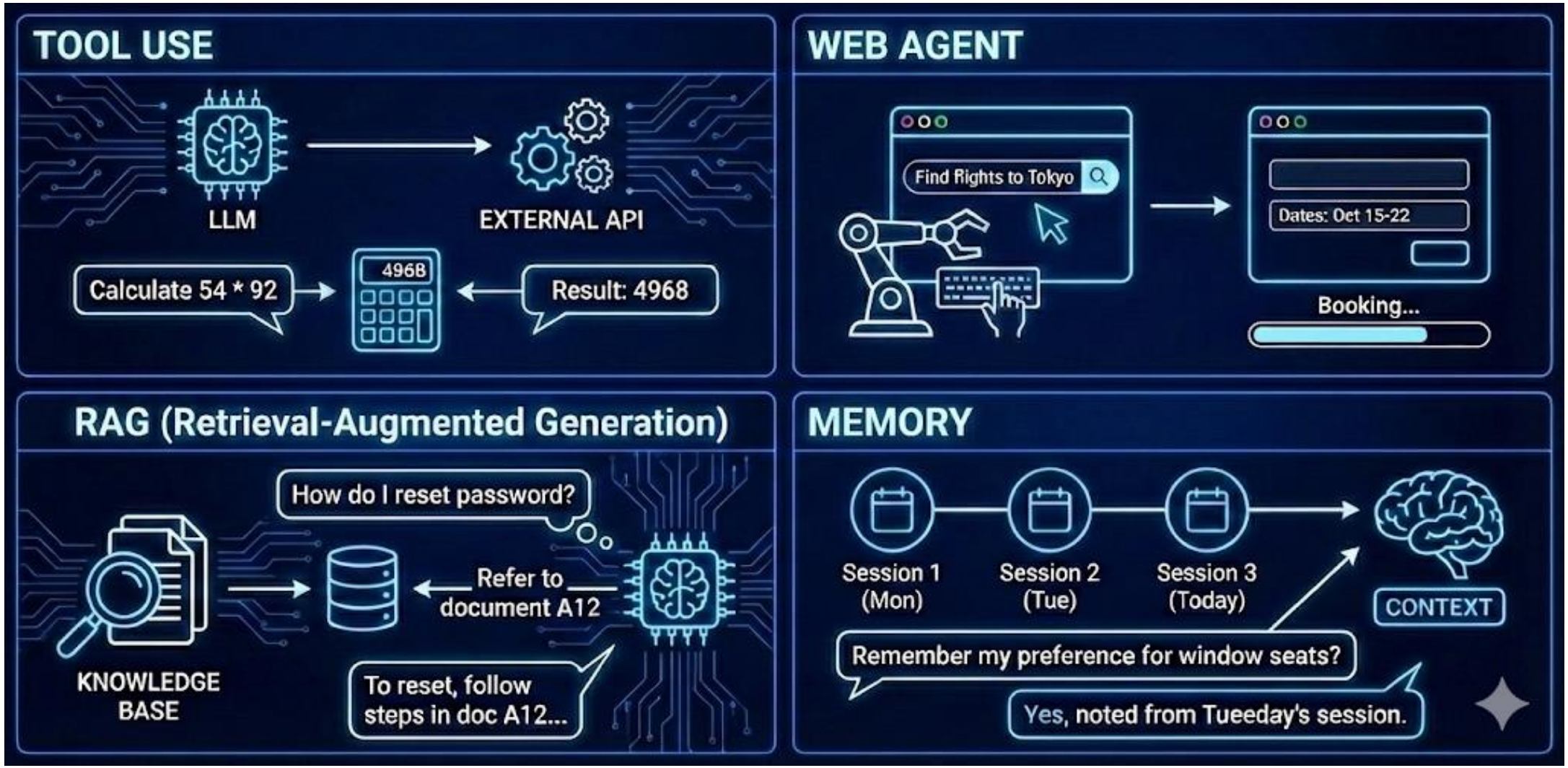
DeepSeek-R1

- **RLVR:**
  A*utomatically verifiable* rewards instead of human preference models
- **Strong reasoning via pure RL:**
  Large-scale RL alone can induce strong reasoning behaviors without extensive supervised chain-of-thought data
- **GRPO-style optimization:**
  GRPO + iterative RL training improves LLMs at scale

Outline

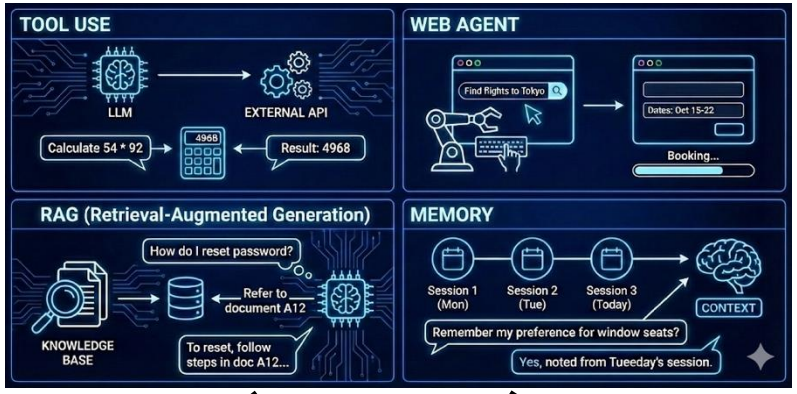


Steering/optimization in LLM downstream tasks



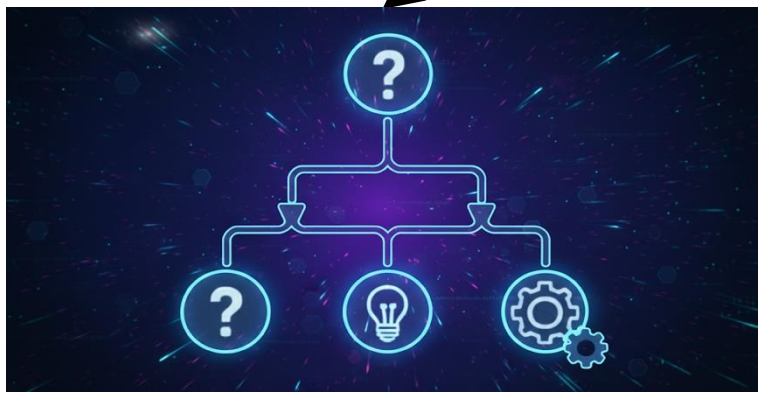
Current practice of steering/optimization in LLM tasks

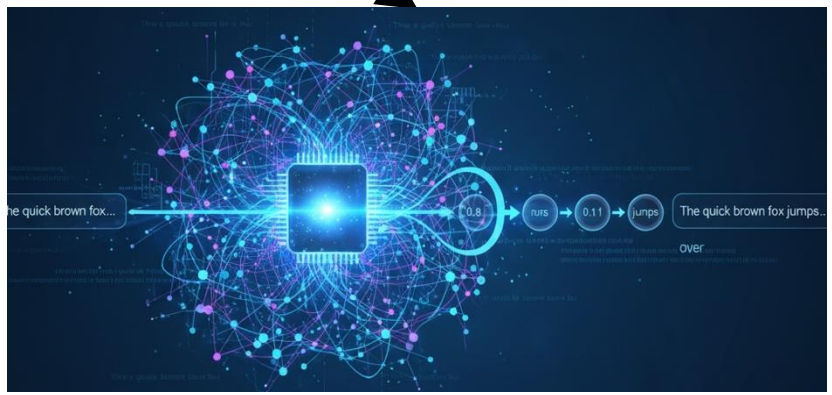
Steering / OPT
in LLM tasks



OPT as training free
heuristics (e.g., tree-search)

OPT as LLM decoding via
"finer-tuning"





W&M DATA SCIENCE

OPT as training-free heuristics

Optimization as heuristics

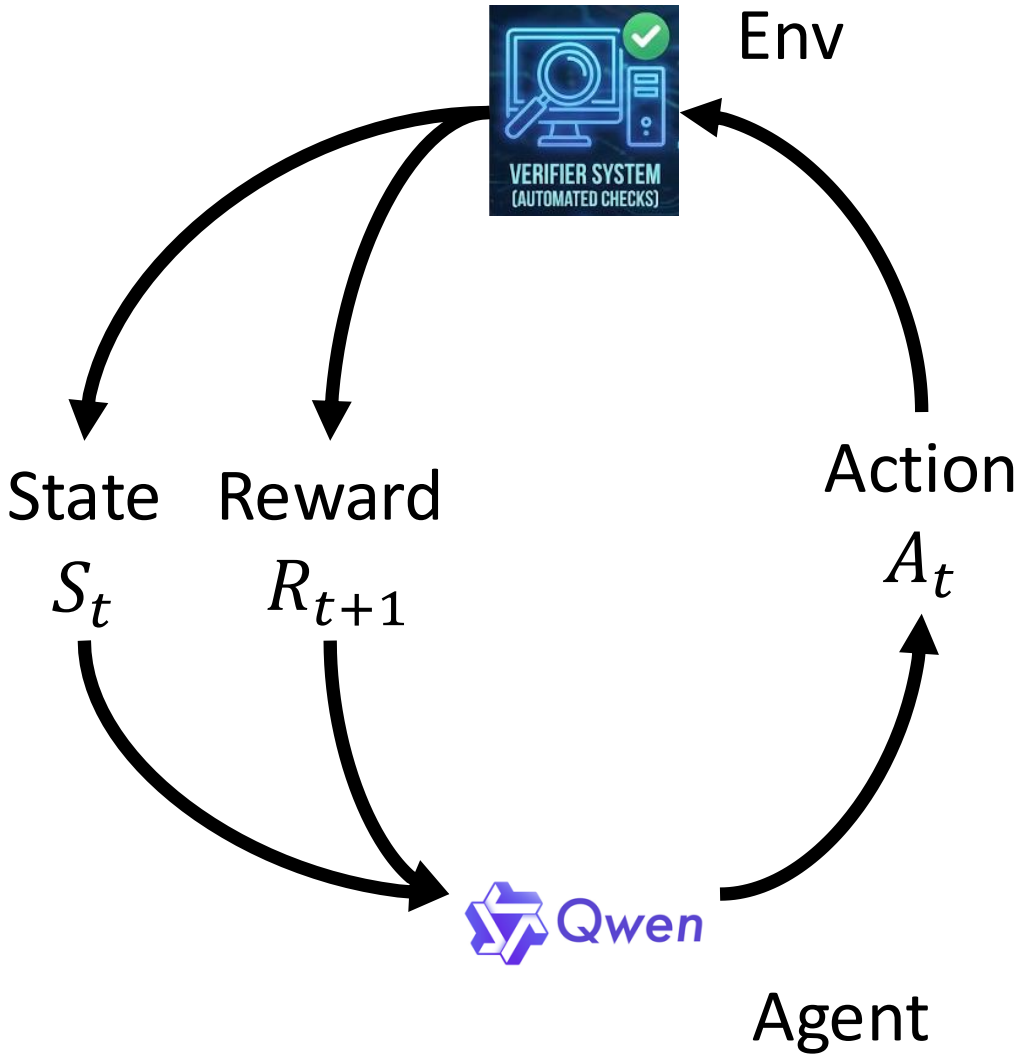
- [RAG] Lewis, Patrick, et al. ***Retrieval-augmented generation for knowledge-intensive NLP tasks***. NeurIPS, 2020.
- [Reasoning] Yao, S., et al. ***Tree of Thoughts: Deliberate Problem Solving with Large Language Models***. NeurIPS, 2023.
- [Speculative sampling] Li et al., ***EAGLE-2: Faster Inference of Language Models with Dynamic Draft Trees*.** EMNLP, 2024.

OPT as LLM decoding via ***finer-tuning***

Optimization as part of LLM decoding, fine-tuned via task-level rewards

- [Tool use] Qian et al. ***ToolRL: Reward is all tool learning needs***." NeurIPS, 2025.
- [Web agent] Qi et al. ***WebRL: Training LLM Web Agents via Self-Evolving Online Curriculum Reinforcement Learning***. ICLR, 2025.
- [RAG] Ma et al. **Query rewriting in retrieval-augmented large language models**. EMNLP, 2023.
- [Memory] Yan et al. ***Memory-R1: Enhancing Large Language Model Agents to Manage and Utilize Memories via Reinforcement Learning.*** Arxiv, 2026.

Finer-tuning example 1: Tool use

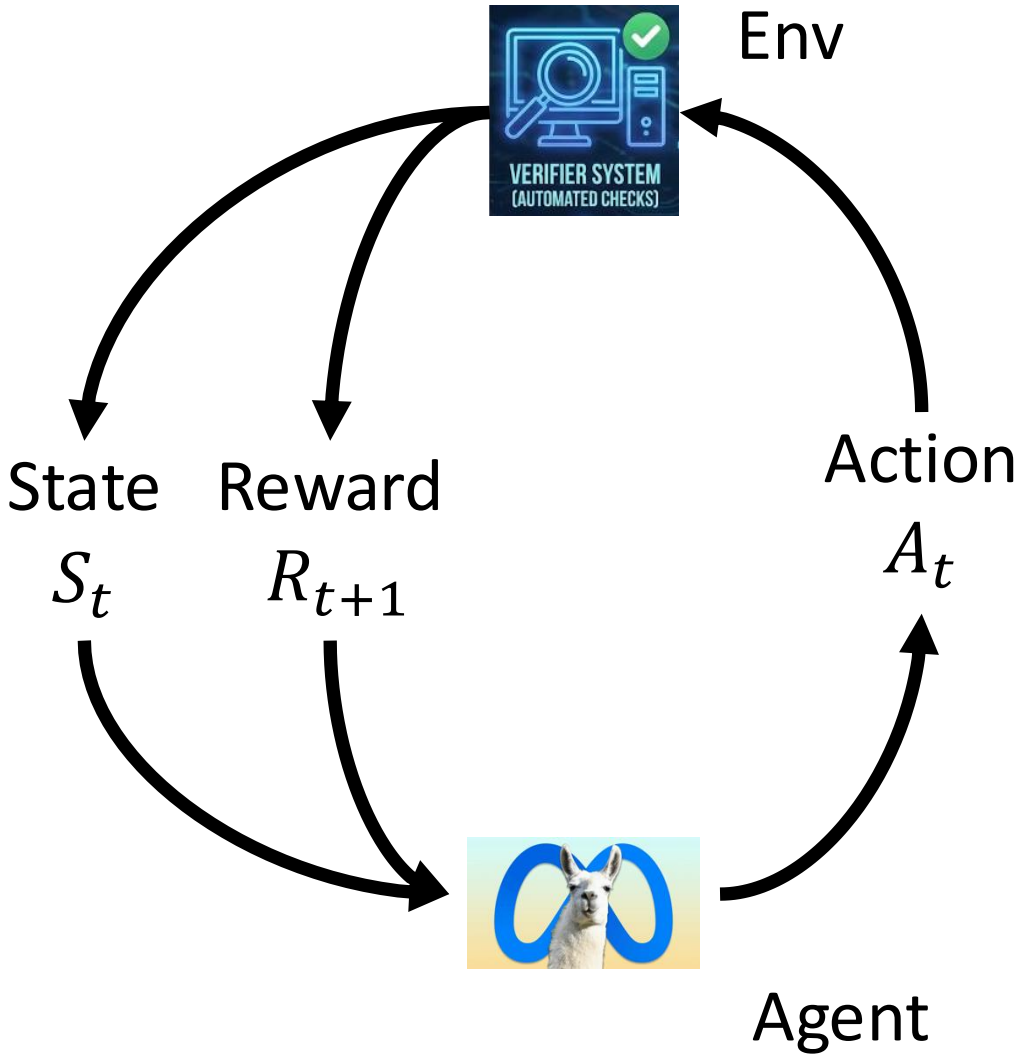


- **Env**: tools + verifier system
- **State:** user prompt + tool calling history and execution so far
- **Action:** generation of tool calling instruction
- **Reward**: verifier feedback

- **Agent:** LLM as a tool usage agent

Qian, Cheng, et al. "Toolrl: Reward is all tool learning needs." *NeurIPS, 2025*

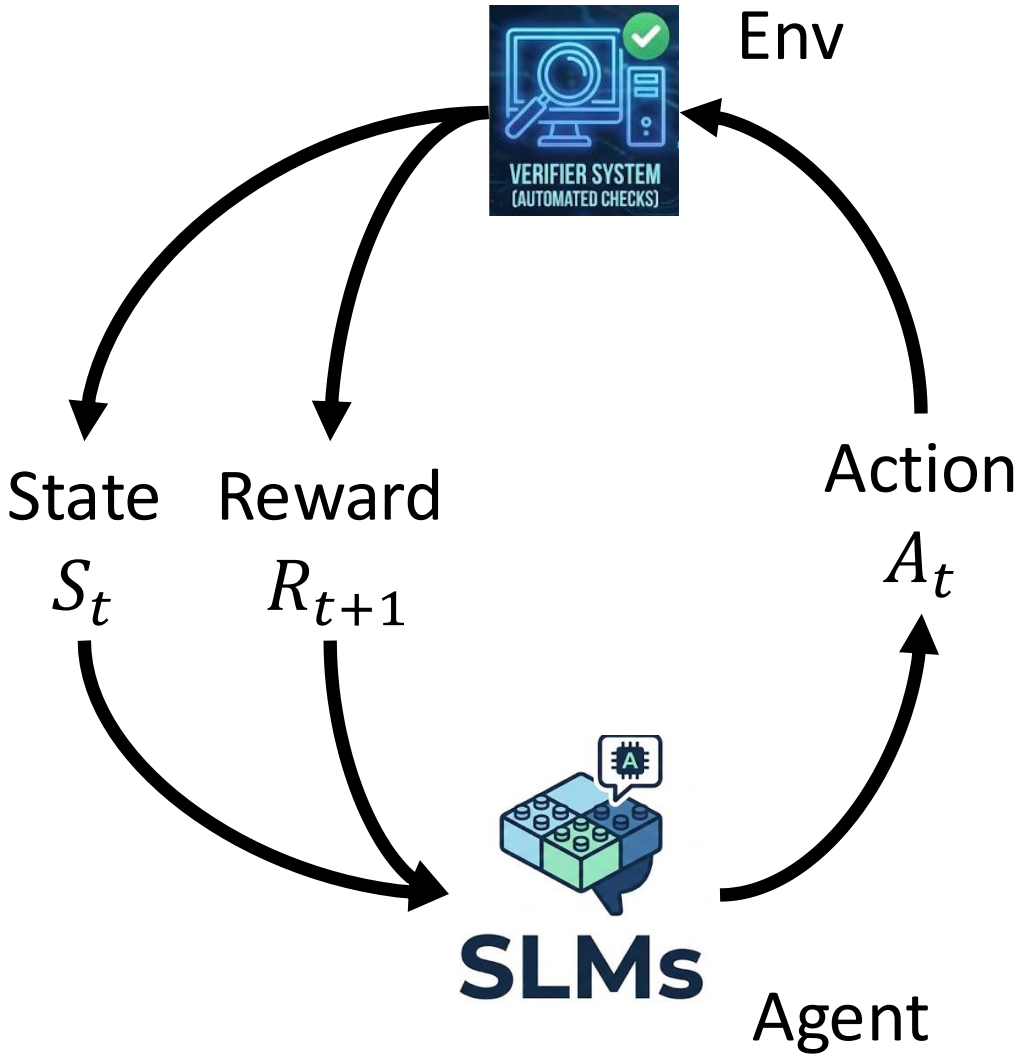
Finer-tuning example 2: Web agent



- **Env**: website engines **+** verifier system
- **State:** query + action history + website HTML
- **Action:** generate the next move (e.g., click, Scroll, search,…)
- **Reward**: verifier feedback
- **Agent:** LLM as a web agent

Qi, Zehan, et al. "WebRL: Training LLM Web Agents via Self-Evolving Online Curriculum Reinforcement Learning." *ICLR 2025*

Finer-tuning example 3: RAG

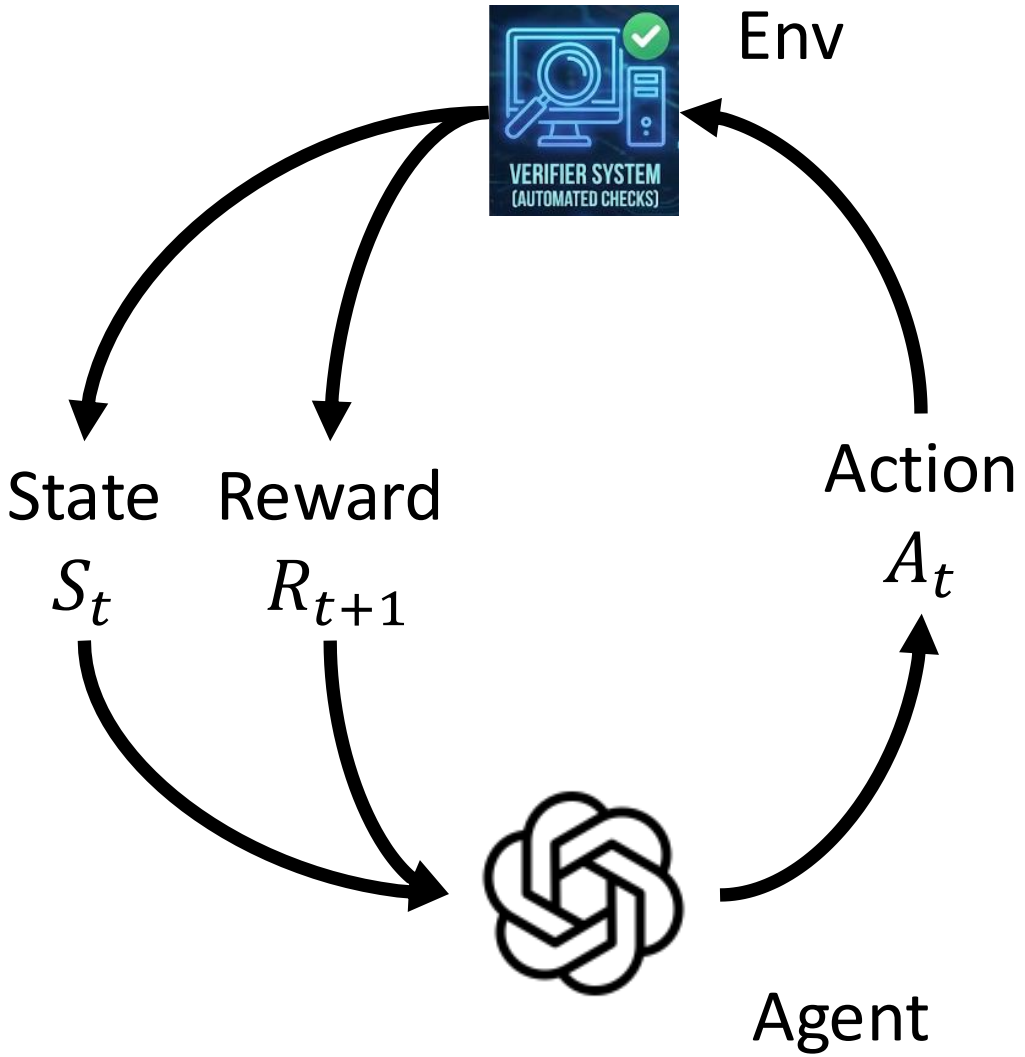


- **Env:** search engine as retriever + frozen LLM as reader + verifier system
- **State:** query + the tokens generated by the rewriter so far
- **Action:** the generation of a token
- **Reward:** verifier feedback

- **Agent:** SLM (T5-large) as rewriter

Ma et al. " Query Rewriting for Retrieval-Augmented Large Language Models" *EMNLP, 2023*.

Finer-tuning example 4: Memory management



- **Env:** memory bank + frozen LLM as answer agent + verifier system
- **State:** retrieved memories + new information
- **Action:** the operations {ADD, UPDATE, DELETE, NOOP} + updated content.
- **Reward:** verifier feedback
- **Agent:** LLM as memory manager

Yan et al. " Memory-R1: Enhancing Large Language Model Agents to Manage and Utilize Memories via Reinforcement Learning" arXiv, 2025

Finer-tuning **is great, but is it really necessary?**

- Finer-tuning of LLMs usually leads to **model bias**
- Still needs substantial **compute**
- Needs access to the model (**white-box**)

- Can we do even better?

Outline

Auxiliary Policy Models (APMs)

- Key idea: **use light-weight reinforcement learning to train separate small APMs** to steer LLMs in down-stream tasks
- Still follows the **RLVR** framework
- Down stream tasks are formalized as Markov Decision Processes (MDPs)
- A **frozen LLM** is treated as part of the environment

Why APMs?

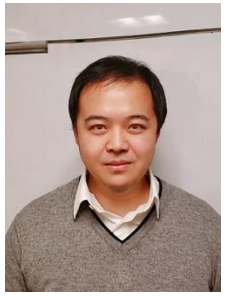
- More **flexible** solution
  - No need to touch the LLMs, model bias is injected to the APMs
- **Low training cost**
  - Ideal for low-resource computing, e.g., academia 🙂
- **Low inference cost**
  - Great for time-sensitive tasks and low-end device like edge computing
- (Sometimes) Works with a **black-box** model

Our exemplary studies

- Chenan Wang, Daniel Shi, Haipeng Chen. ***Re-SpS: Speculative Sampling with Reinforcement Learning***. AAAI, 2026.


- Mahmud Wasif Nafee*, Maiqi Jiang*, Haipeng Chen†, and Yanfu Zhang†. ***DR-IKE: Dynamic Retriever for In-Context Knowledge Editing via Policy Optimization***. EMNLP, 2025.





Our exemplary studies

- Chenan Wang, Daniel Shi, Haipeng Chen. ***Re-SpS: Speculative Sampling with Reinforcement Learning***. AAAI, 2026.

- Mahmud Wasif Nafee*, Maiqi Jiang*, Haipeng Chen†, and Yanfu Zhang†. ***DR-IKE: Dynamic Retriever for In-Context Knowledge Editing via Policy Optimization***. EMNLP, 2025.

Speculative Sampling (SpS)

- A technique used to accelerate LLM inference by parallelizing token generation through a “draft-then-verify” cycle

Leviathan et al. Fast inference from transformers via speculative decoding. *ICML*, 2023.
Chen et al. "Accelerating large language model decoding with speculative sampling." *ArXiv,* 2023.

Background: Tree-based drafting

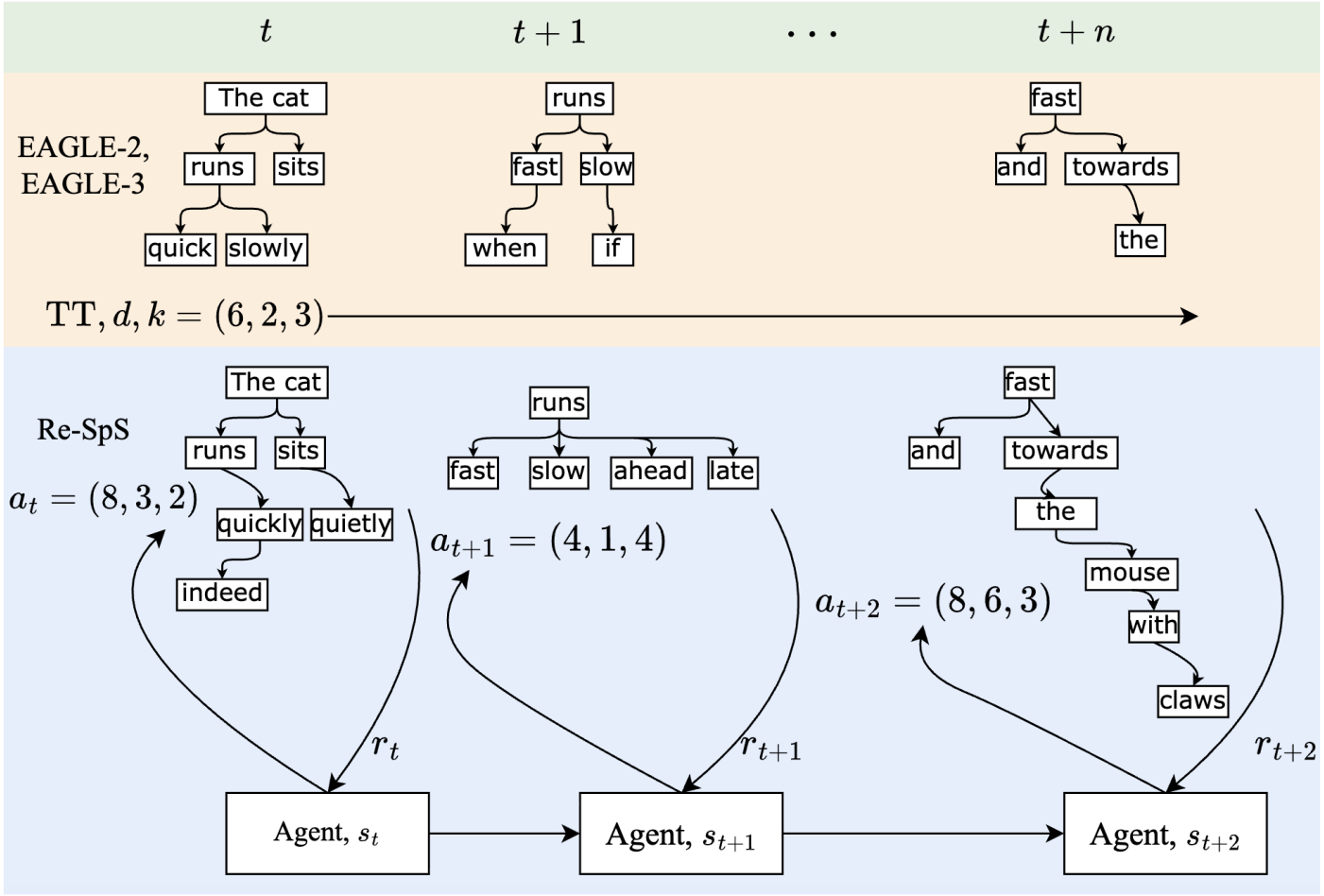
- **Tree-based SpS:** Uses tree-based drafting to verify multiple token paths in parallel
- **Current SOTA (EAGLE 2 & 3**, Li et al., 2024-2025**)**

- **Gap**: Relies on **static hyperparameters** for the tree structure, failing to adapt to contexts of varying complexity. E.g, total tokens, tree depth, expansion factor

Miao et al. "Specinfer: Accelerating large language model serving with tree-based speculative inference and verification." ASPLOS 2024.
Li et al. "EAGLE: Speculative Sampling Requires Rethinking Feature Uncertainty." *ICML*, 2024.
Li et al. "Eagle-3: Scaling up inference acceleration of large language models via training-time test." *ArXiv,* 2025.

Re-SpS: Speculative Sampling with Reinforcement Learning

- **EAGLE-2 & 3**: Relies on static hyperparameters for the tree structure
- **Re-SpS**: Dynamically optimizes draft tree hyperparameters based on the context



Re-SpS vs. EAGLE-2 & 3

Wang et al. *Speculative Sampling with Reinforcement Learning*. AAAI, 2026.

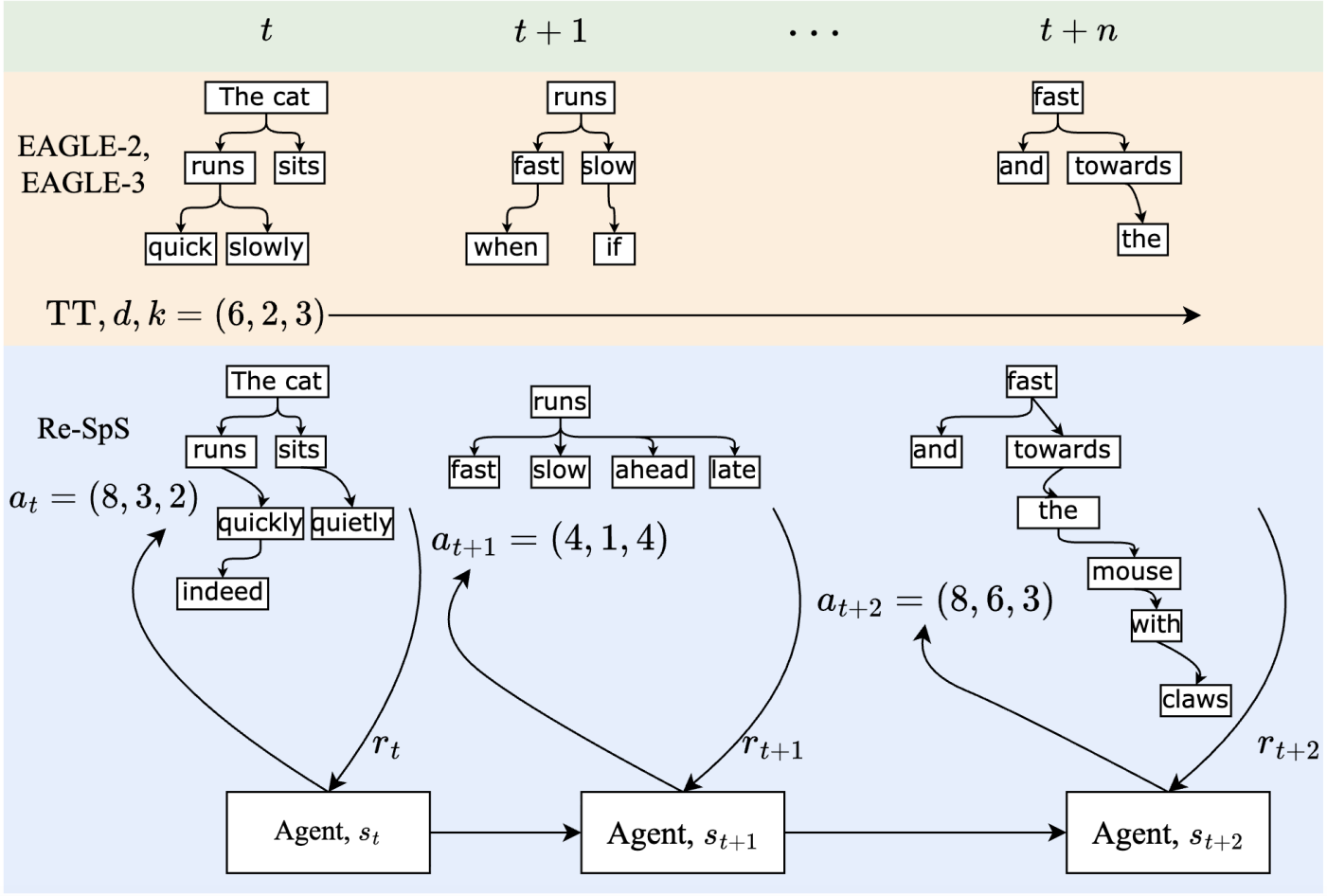
W&M DATA SCIENCE

The RL formulation of SpS

- **Env:** frozen LLM + speculative decoding framework (EAGLE-3)
- **State:** prompt + all previous tokens
- **Action:** tree structure hyperparameters (TT=total tokens, d=depth, k=expansion factor)
- **Reward:** token/second

$$r_t = \frac{\text{accepted tokens}}{\text{elapsed time (seconds)}}$$

- **Agent**: **APMs (2-layer MLP)**



Re-SpS vs. EAGLE-2 & 3

Wang et al. *Speculative Sampling with Reinforcement Learning*. AAAI, 2026.

W&M DATA SCIENCE

Computation overhead

- Optimizing SpS with RL introduces computation/memory overhead:
  - **State representation overhead:** Generating rich semantic embeddings (e.g., SentenceBERT) at every generation step incurs significant latency (~5–15 ms per call) and memory overhead.
  - **Frequent policy calls**: Performing frequent policy calls (e.g., per drafting step) further aggravated the above overhead.

Overcoming RL overhead

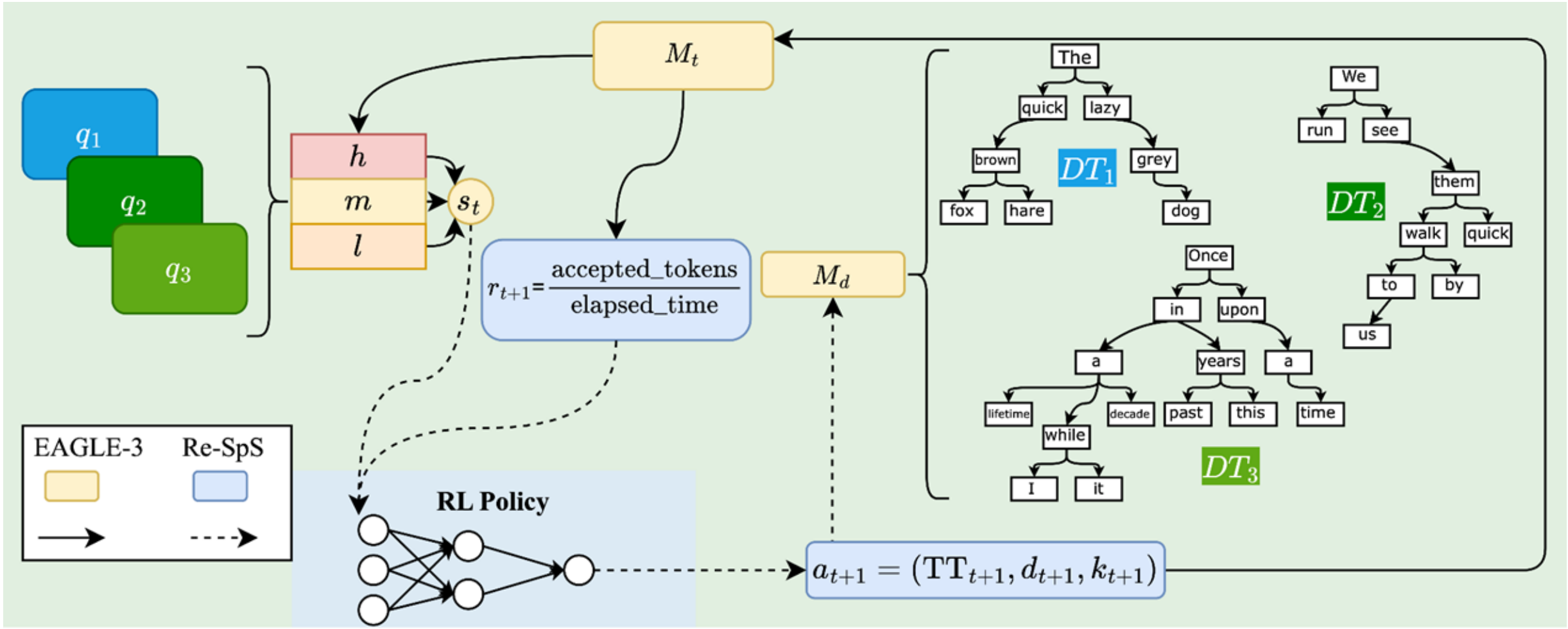
- **Innovation 1: Efficient feature reuse**
  - Instead of costly external embeddings (e.g., SentenceBERT), we reuse the **target model's internal hidden states**:

$$s_t = [h_{\text{LM}}^{(h,m,l)}]$$

🙂 *Free lunch - Rich context representation with zero extra inference cost*

- **Innovation 2: Multi-step action persistence**
  - Amortize policy inference cost by caching the selected action for every N decoding steps (e.g., N = 10 or 30)

Overview of Re-SpS



Wang et al. *Speculative Sampling with Reinforcement Learning*. AAAI, 2026.

W&M DATA SCIENCE

Experimental setup

- Target models: LLaMA 3.1-8B, Vicuna-13B, LLaMA 3.3-70B
- Across 5 benchmarks (MT-Bench, HumanEval, GSM8K, Alpaca, CNN/DM)

- LLaMA 3.1-8B and Vicuna-13B runs on a **single NVIDIA A40 GPU (40GB)**
- LLaMA 3.3-70B runs on **4 NVIDIA H100 GPUs (80GB each)**

- All experiments use PPO as the RL algorithm
- **APM represented as a 128x128 MLP**
- **Training time**: LLaMA3.3-8B – **8.65 hrs**; Vicuna-13B – **10.13 hrs**; LLaMA3.1-70B – **11.05 hrs**

Re-SpS empirical results

Backbone	Method	MT-Bench	HumanEval	GSM8K	Alpaca	CNN/DM	Mean	p-value
LLaMA 3.1-8B	Medusa	2.07×	2.50×	2.23×	2.08×	1.71×	2.12×	–
	Hydra	2.88×	3.28×	2.93×	2.86×	2.05×	2.80×	–
	EAGLE-3	3.39×	3.65×	3.52×	3.67×	**2.96**×	3.44×	–
	Re-SpS	**3.43**×	**3.89**×	**3.62**×	**3.90**×	2.87×	**3.54**×	$< 10^{-4}$
Vicuna-13B	Medusa	2.07×	2.50×	2.23×	2.08×	1.71×	2.12×	–
	Hydra	2.88×	3.28×	2.93×	2.86×	2.05×	2.80×	–
	EAGLE-3	3.75×	4.28×	3.85×	3.76×	**3.35**×	3.80×	–
	Re-SpS	**3.76**×	**4.64**×	**3.99**×	**3.99**×	3.24×	**3.92**×	$< 10^{-9}$
LLaMA 3.3-70B	EAGLE-3	4.35×	4.87×	4.74×	4.77×	**4.09**×	4.46×	–
	Re-SpS	**4.47**×	**5.45**×	**5.13**×	**5.34**×	4.03×	**4.88**×	$< 10^{-29}$

- **Average speedup vs EAGLE-3:**
  1) **LLaMA 3.1-8B:** 1.03×. 2) **Vicuna-13B:** 1.04×. 3) **LLaMA 3.3-70B:** 1.06×.
- **Fidelity**: Lossless (exact match to greedy decoding)

Wang et al. *Speculative Sampling with Reinforcement Learning*. AAAI, 2026.

Our exemplary studies

- Chenan Wang, Daniel Shi, Haipeng Chen. ***Re-SpS: Speculative Sampling with Reinforcement Learning***. AAAI, 2026.

- Mahmud Wasif Nafee*, Maiqi Jiang*, Haipeng Chen†, and Yanfu Zhang†. ***DR-IKE: Dynamic Retriever for In-Context Knowledge Editing via Policy Optimization***. EMNLP, 2025.

In-context knowledge editing

- **In-Context Knowledge Editing**: Edit model knowledge via prompted demonstrations

🕐 Before Editing

👤 **Question:** *Who is the current Prime Minister of the United Kingdom?*

📄 **Model's Stored Knowledge (Pre-Update):** "Rishi Sunak has been serving as Prime Minister from 2022"

🤖 **LLM:** *Rishi Sunak.* ❌

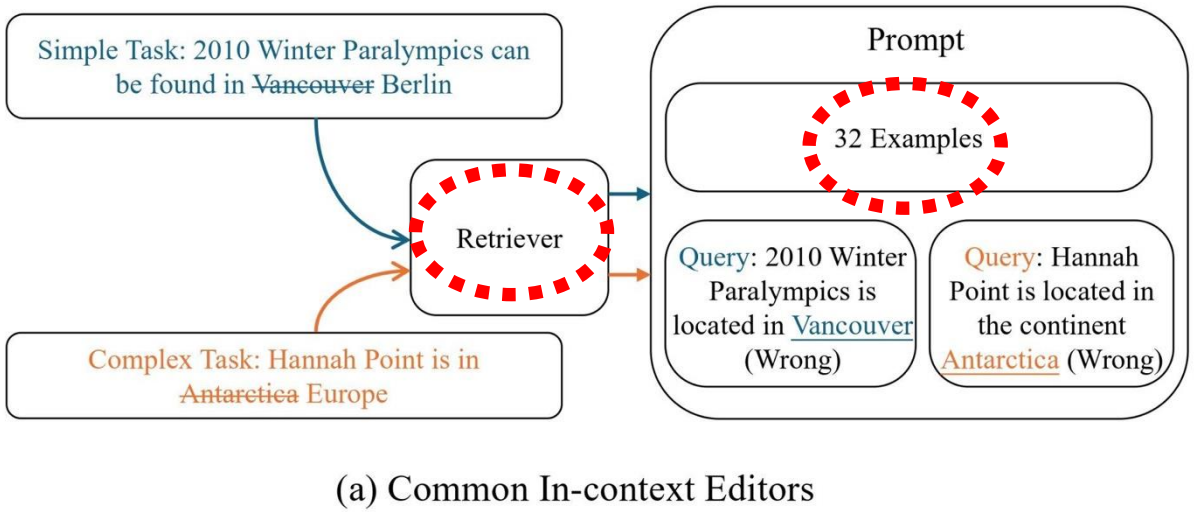
🧩 After Editing

👤 **Question :** *Who is the current Prime Minister of the United Kingdom?*

📄 **Model's Injected Knowledge (Post-Update):** "Keir Starmer is the UK Prime Minister now"

🤖 **LLM (with in-context prompt or modified parameters):** *Keir Starmer.* ✅

In-context knowledge editing



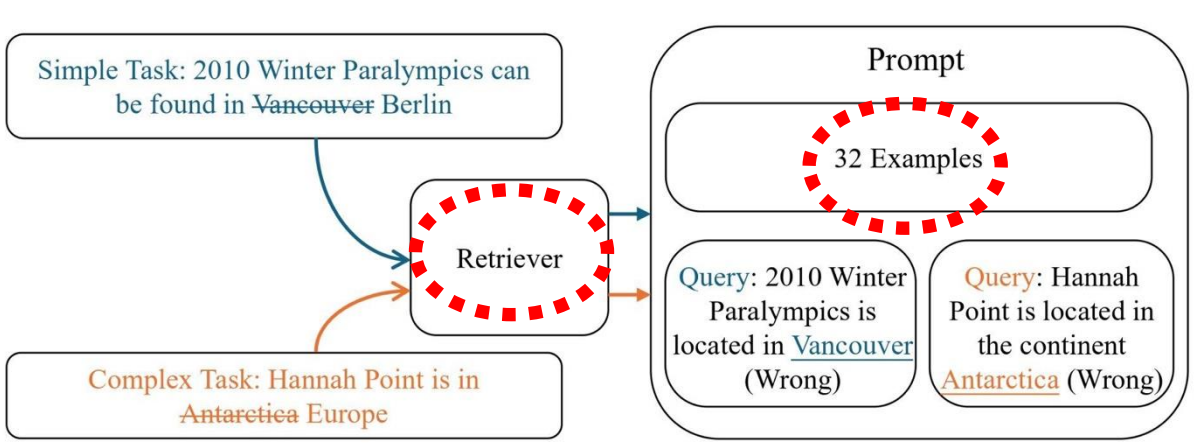
(a) Common In-context Editors

Current in-context editors (e.g., IKE [Zheng et al. 2023])

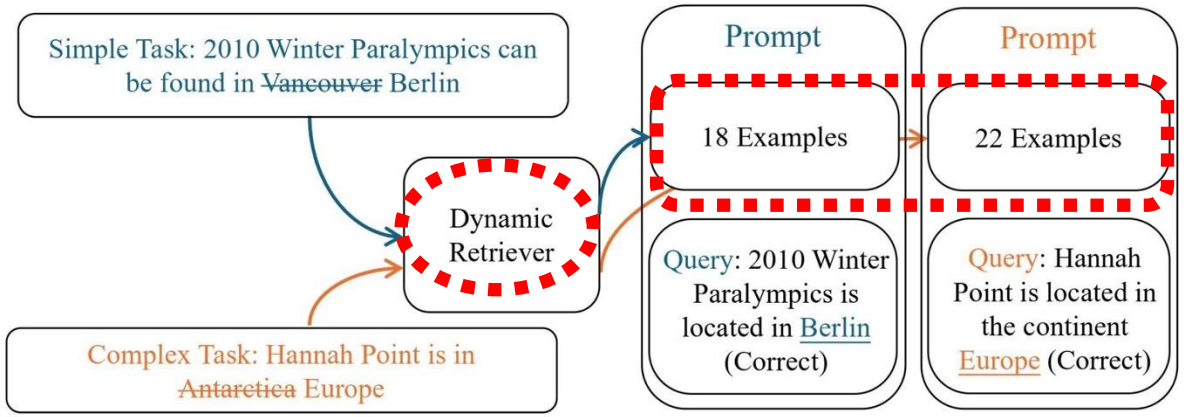
- Use **static retrieval** and **fixed number of prompt templates**
- **Ignore task difficulty** — simple and complex edits get equal-length prompts
- Become unstable and less generalizable when examples are **redundant or poorly aligned**

Zheng et al. "Can We Edit Factual Knowledge by In-Context Learning?." *EMNLP, 2023*.

Dynamic Retriever for In-Context Knowledge Editing via Policy Optimization



(a) Common In-context Editors

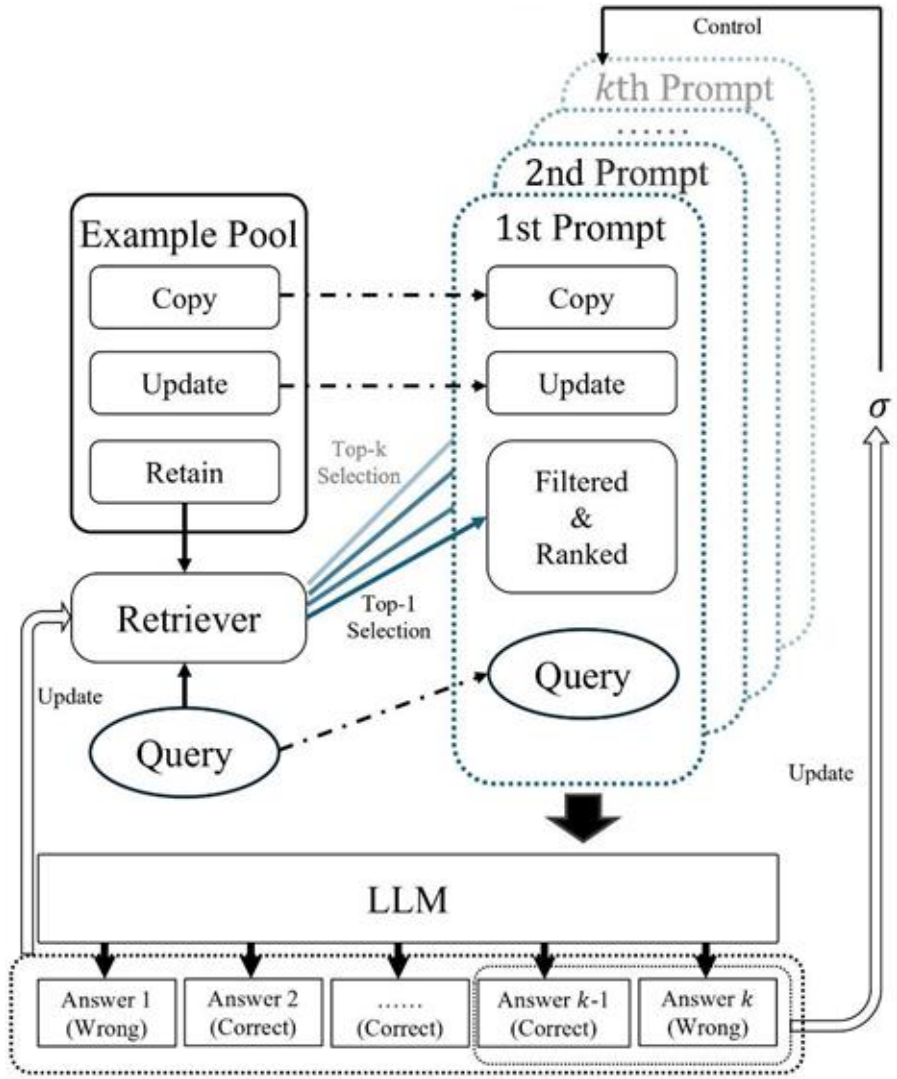


(b) Our Dynamic Retriever In-context Editor

Our approach: **DR-IKE** (Dynamic Retriever for In-Context Knowledge Editing)

- **Adaptive context construction:** Adjusts the **ranking and number** of examples per query
- **Auxiliary retrieval policy model:** Incorporates LLM response correctness as reward signals to train a retrieval APM

Overview of DR-IKE

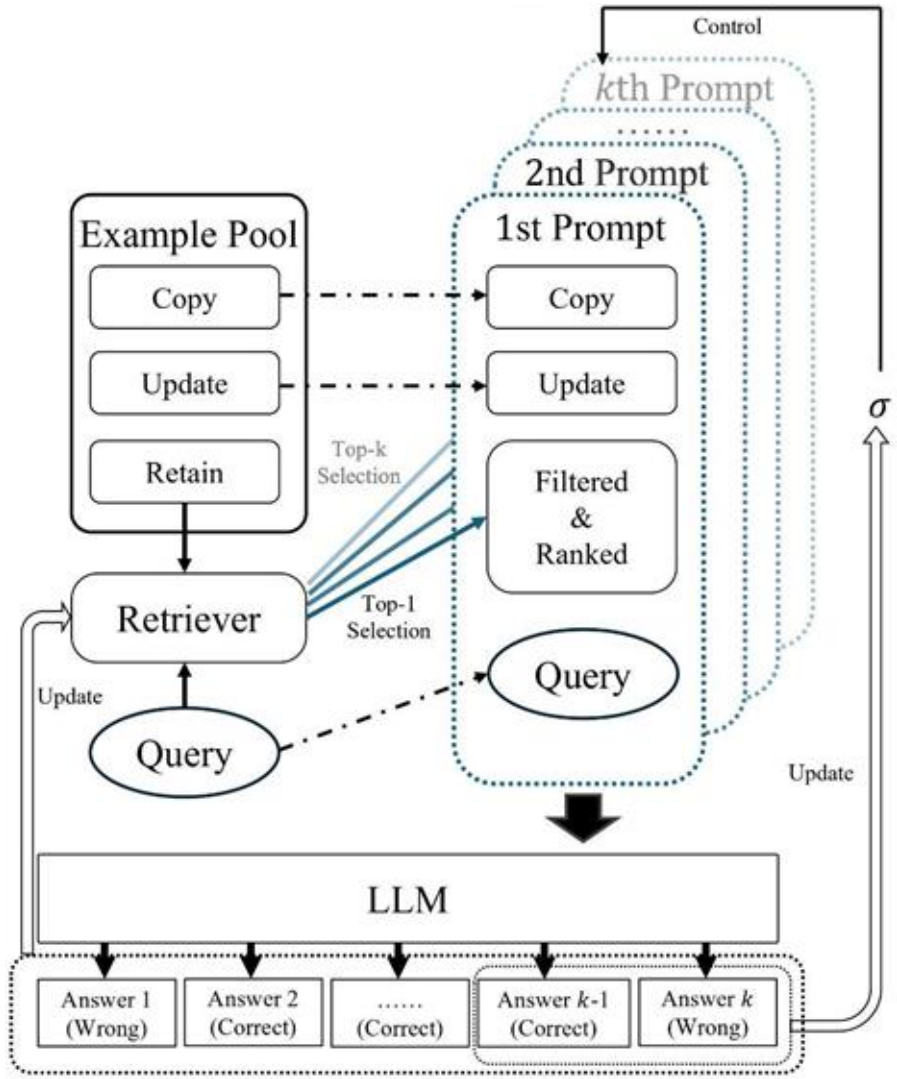


- **Example pool:** Three factual types *Copy*, *Update*, *Retain*.
- **Retriever:** frozen BERT + trainable linear layer → scores retain examples by relevance.
- **Prompt assembler:** Combines Copy + Update + Top-k Retain examples.
- **LLM editor:** Generates response; reward signal drives training.
- **Budget controller (σ):** Controls how many retain examples enter the prompt. Can be a hyperparameter or a learnable parameter.

Nafee et al. DR-IKE: Dynamic Retriever for In-Context Knowledge Editing via Policy Optimization. EMNLP, 2025.

W&M DATA SCIENCE

Overview of DR-IKE



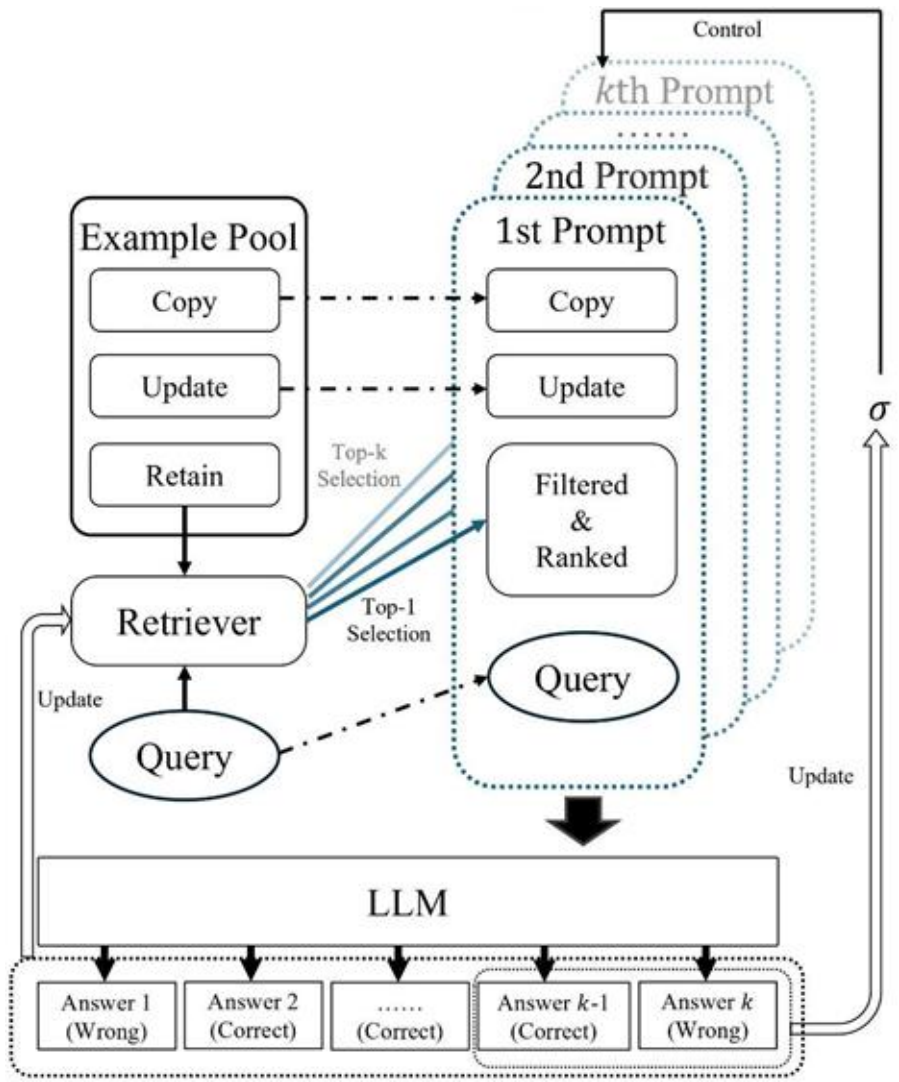
Dynamic Retriever as the APM

- **Frozen BERT encoder** for contextual features
- **Trainable linear head** producing candidate scores
- Softmax over scores gives selection probabilities
- High-probability retains → presented first

Nafee et al. DR-IKE: Dynamic Retriever for In-Context Knowledge Editing via Policy Optimization. EMNLP, 2025.

W&M DATA SCIENCE

Overview of DR-IKE



The RL formulation of DR-IKE

- **Env**: Frozen target LLM + in-context knowledge editing task and feedback
- **State**: Query + candidate retain examples
- **Action**: Ranking and selection of examples for the prompt
- **Reward**: +1 → if LLM's output matches the correct answer, –1 → otherwise
- **Agent: APM (linear head** after frozen Sentence-BERT**)**

Nafee et al. DR-IKE: Dynamic Retriever for In-Context Knowledge Editing via Policy Optimization. EMNLP, 2025.

W&M DATA SCIENCE

Experimental setup

- **Three evaluation datasets** (CounterFact, zsRE, Wikidata-CF)
- Evaluation **Metrics**:
  - **ESR** — Edit Success Rate: Accuracy on edited facts
  - **PC** — Paraphrase Consistency: Consistency across reworded queries
  - **RR** — Retain Rate: Preservation of unrelated original facts

Experimental setup

- **APM training**:
  - Same frozen LLM backbone (Llama-3.1-8B-Instruct) is used across all datasets
  - BERT-small (29M) as the **frozen encoder**, a **trainable linear layer** as the APM
  - Only needs **300 examples** for the APM training
  - All training done with a **single NVIDIA L4 GPU (24G), training time < an hour**

Results

CounterFact

Method	ESR ↑	PC ↑	RR ↑
FactPrompt	0.61	0.34	0.43
EditCoT	0.70	0.33	0.40
IKE	0.76	0.67	**0.76**
DR-IKE	**0.89**	**0.81**	0.66

- DR-IKE outperforms all baselines in edit accuracy and consistency
- Offers a better trade-off between successful edits and knowledge retention

Nafee et al. DR-IKE: Dynamic Retriever for In-Context Knowledge Editing via Policy Optimization. EMNLP, 2025.

Results

zsRE

Method	ESR ↑	PC ↑	RR ↑
IKE	0.30	0.22	0.49
DR-IKE	**0.33**	**0.26**	**0.51**

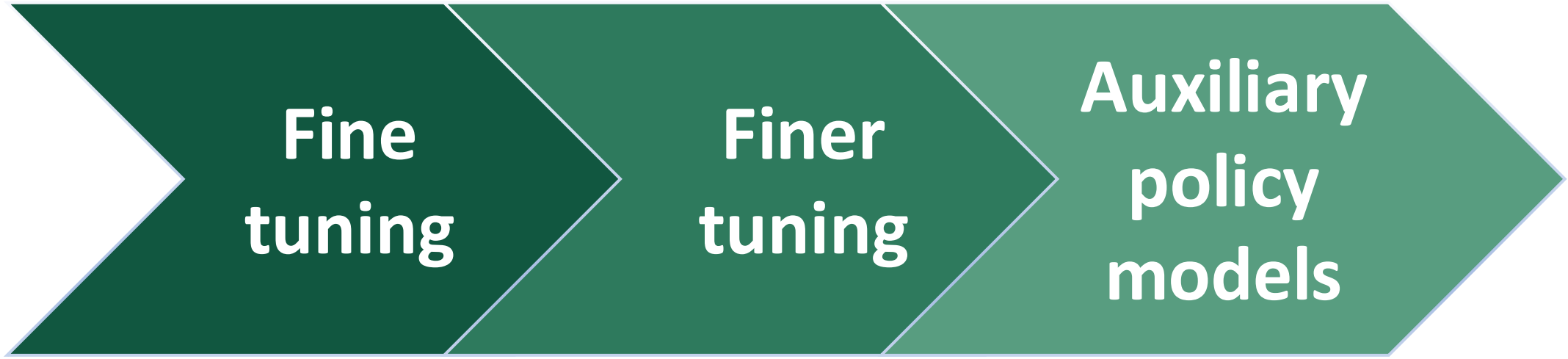
Wikidata-CF

Method	ESR ↑	PC ↑	RR ↑
IKE	0.39	0.40	0.63
DR-IKE	**0.42**	**0.43**	**0.64**

Nafee et al. DR-IKE: Dynamic Retriever for In-Context Knowledge Editing via Policy Optimization. EMNLP, 2025.

APM takeaways

- Lightweight APMs can be used to steer LLM downstream tasks such as speculative sampling and in-context knowledge editing
- **Low-cost, fast, flexible**, and (sometimes) works for **black-box** settings
- Down stream tasks are formalized as MDPs
  - Still follows the **RLVR** framework – needs **automatically verifiable reward**
  - A frozen LLM is part of the environment
- **Code repositories for both projects are released on GitHub**



The future of AGI is not a single gigantic model, but a "community" of big foundational and small/tiny auxiliary models.

Thanks! Questions?

Haipeng Chen

Data Driven Decision Intelligence (D3i) Lab

Assistant professor, William & Mary