POST-TRAINING LLMs

Kawin Ethayarajh

Assistant Professor of Applied AI - University of Chicago, Booth

AI AND ECONOMICS SUMMER INSTITUTE 2026

August 6–11, 2026 • Chicago

The model you use is not the model that was pretrained.

A pretrained model continues text.

USER

How do I estimate the causal effect of a minimum-wage increase?

MODEL

How do I estimate the causal effect of a minimum-wage decrease?

How do I do a DiD study?

How do I make sense of this mortal coil?

A plausible continuation is not necessarily a useful response.

Pretraining learns a distribution over text.

$$\min_{\theta} \mathbb{E}_{x \sim \mathcal{D}} \left[\sum_{t=1}^{T} -\log \pi_\theta(x_t \mid x_{<t}) \right]$$

One generic objective

Trillions of tokens

Result: broad knowledge and capabilities without a reliable user interface.

Jurafsky & Martin (2026), Speech and Language Processing, Ch. 9.

Post-training turns a base model into a useful behavioral policy.

A family of training stages after base pretraining that shape what the model does in practice.

Lambert (2026), Reinforcement Learning from Human Feedback.

Post-training has several goals.

Interface	—	follow instructions, use chat formats, call tools
Capability	—	reason, code, browse, execute long-horizon tasks
Preference	—	be helpful, concise, calibrated, stylistically appropriate
Safety	—	refuse harmful requests without refusing harmless ones
Product	—	obey latency, cost, reliability, and domain constraints

BASE CHECKPOINT	INSTRUCT CHECKPOINT	REASONING CHECKPOINT
Write a plausible continuation	Answer the user helpfully	Think deeply, try multiple paths, call tools, etc.

Post-training changes the probability of already latent behavior and can build* new behavioral routines.

*the extent to which the model learns something totally new is still debated!

A useful (but imperfect) training lifecycle.

The boundaries are conventions; objectives and datasets can blur across stages.

Tunstall et al. The Alignment Handbook.

The stages differ mainly in data, scale, and behavioral specificity.

	Pre-training	Mid-training	Post-training
Data	Web-scale text, code, images	Targeted domain or capability corpora	Demos, preferences, rewards, environments
Typical scale	Trillions of tokens	Billions–trillions	Millions–billions
Signal	What text occurs?	How well-primed is the model for post-training?	What behavior is desired?
Output	Base model	Stronger base	Deployable model

Post-training and alignment answer different questions.

POST-TRAINING

When in the lifecycle?

≠

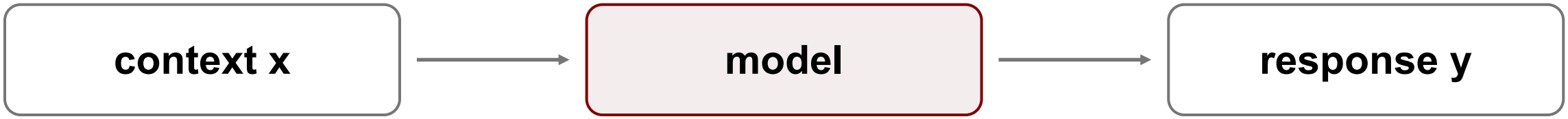
ALIGNMENT

Behavior aligned to what objective?

Most alignment happens in post-training.
Not all post-training is alignment.

The basic object is a policy over responses.

$$\pi_\theta(y \mid x)$$



Post-training changes which responses are probable in which contexts.

The post-training stack we will discuss today.

Supervised Fine-Tuning

Teach by showing.

SFT turns desired behavior into demonstrations.

PROMPT x

Summarize this regression result for a policymaker:

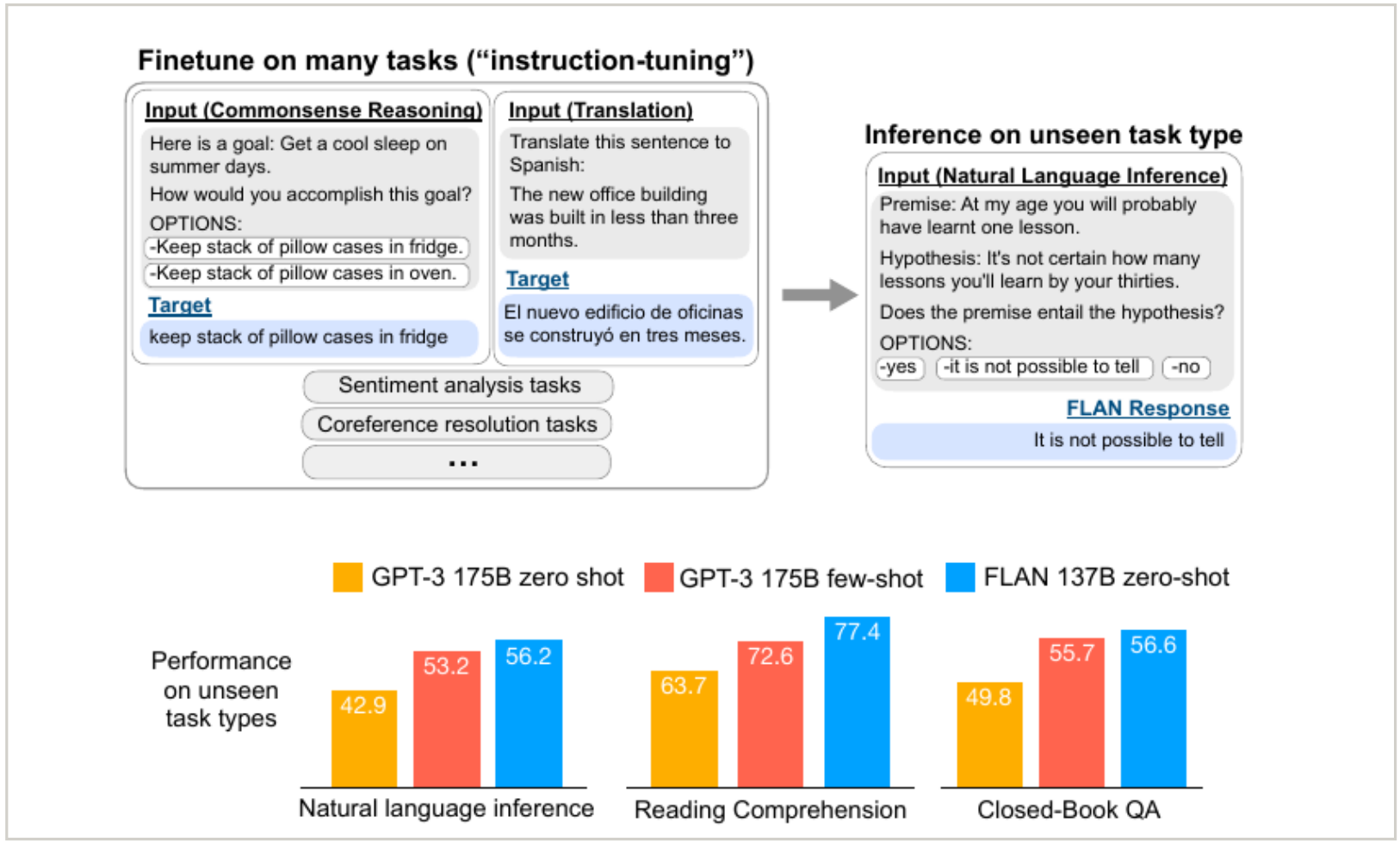
$\hat{\beta} = 0.12,\ SE = 0.09$

DESIRED RESPONSE y★

The estimated effect is positive, but the confidence interval includes zero, so the evidence is inconclusive.

Teach the model what a good response looks like.

A large part of SFT is *instruction-following*.



SFT on many tasks, each phrased as an (instruction, follow-through) pair.

This generalizes to instructions for unseen task types.

Wei et al. (2022), Finetuned Language Models Are Zero-Shot Learners.

The demonstration is not limited to the final output.

INSTRUCTION

x Explain why clustered standard errors may be needed.

y★ Observations within a cluster may have correlated errors...

CODE

x Write Python to estimate a two-way fixed-effects model.

y★ import statsmodels.formula.api as smf ...

TOOL USE

x Find the latest CPI release.

y★ search({"query": "BLS CPI latest"})

SFT uses the same machinery as pretraining.

$$\min_{\theta} \; \mathbb{E}_{(x,y^\star)\sim\mathcal{D}} \left[\sum_{t=1}^{T} -\log \pi_\theta(y_t^\star \mid x, y_{<t}^\star) \right]$$

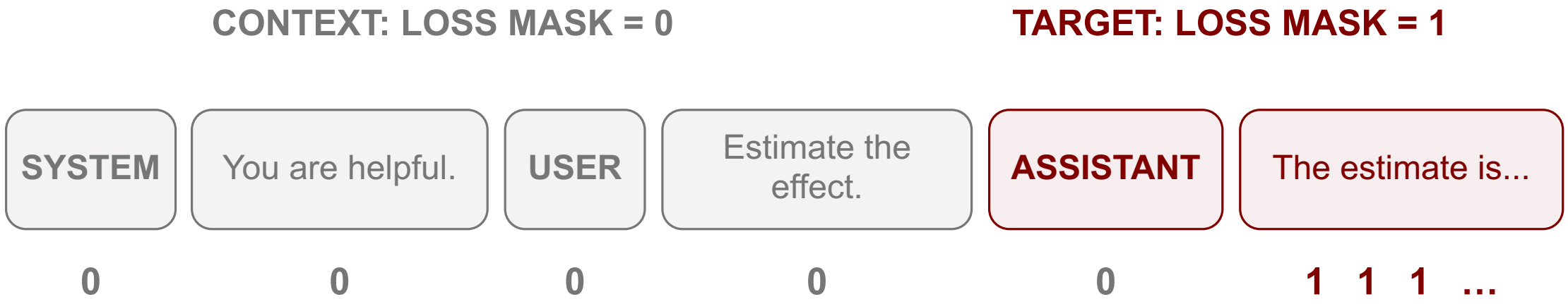
PRETRAINING

next tokens from broad corpora

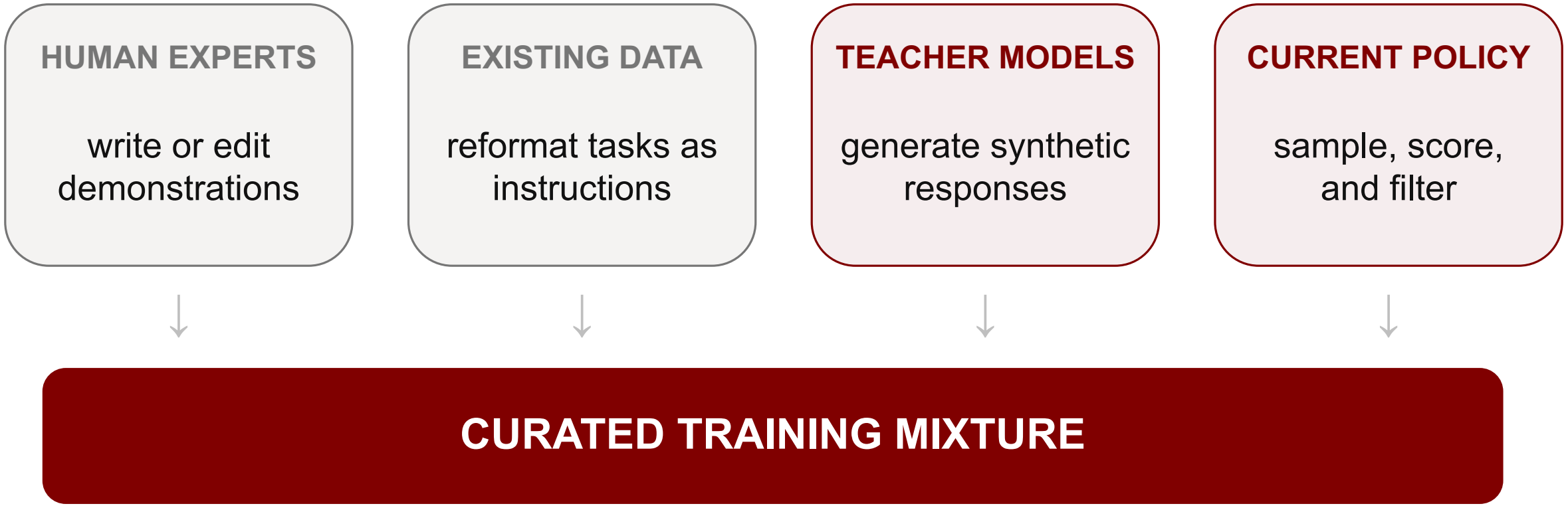
SFT

next tokens from demonstrations

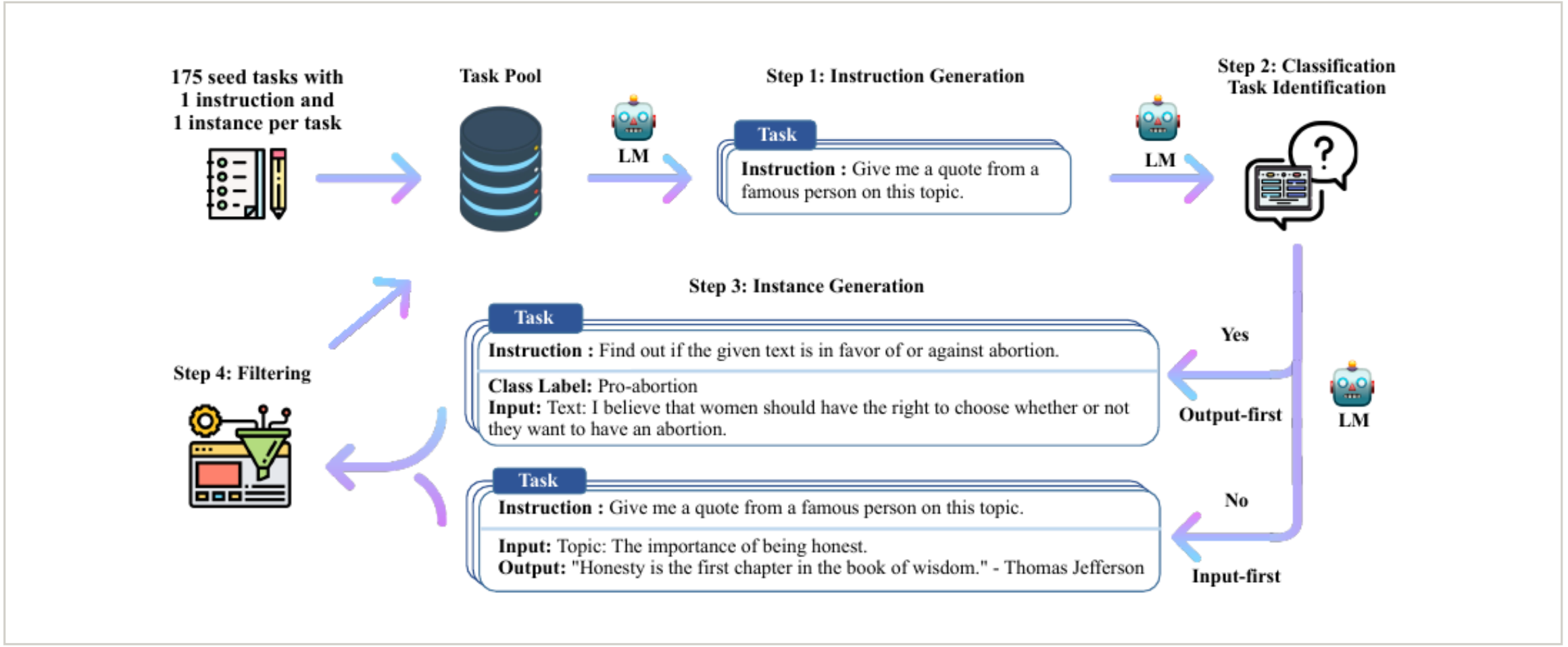
We usually compute loss only on the assistant response.



SFT data comes from several sources.

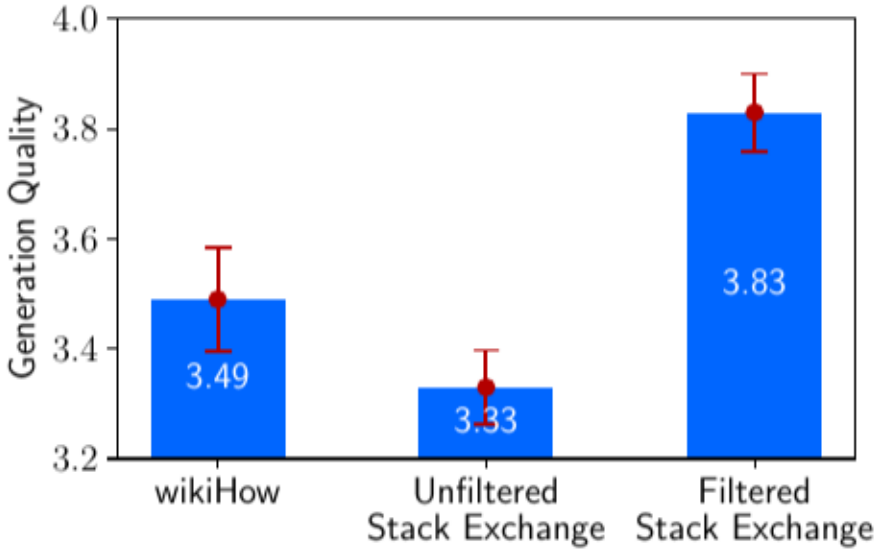


Synthetic data can bootstrap instruction-following.

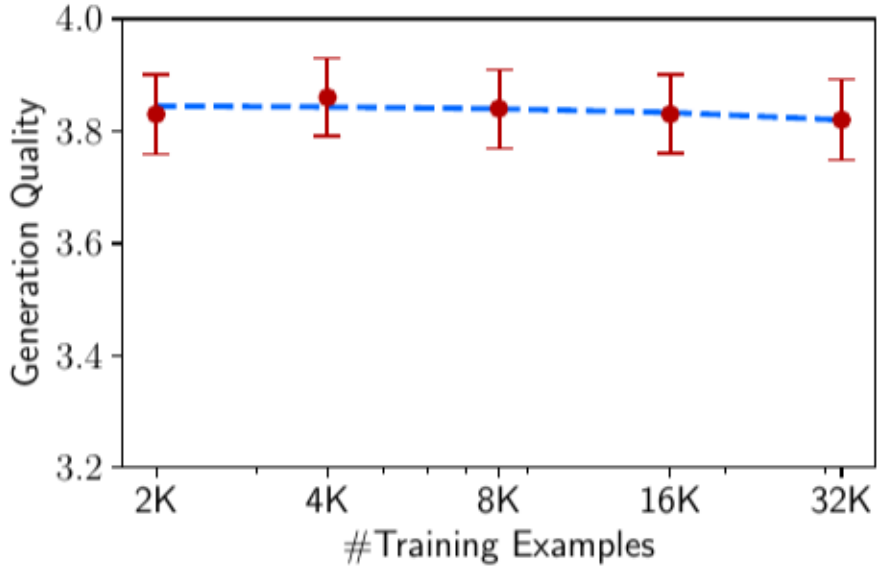


Wang et al. (2023), Self-Instruct: Aligning Language Models with Self-Generated Instructions.

Compared to pretraining, data quality matters more for SFT.



quality + diversity

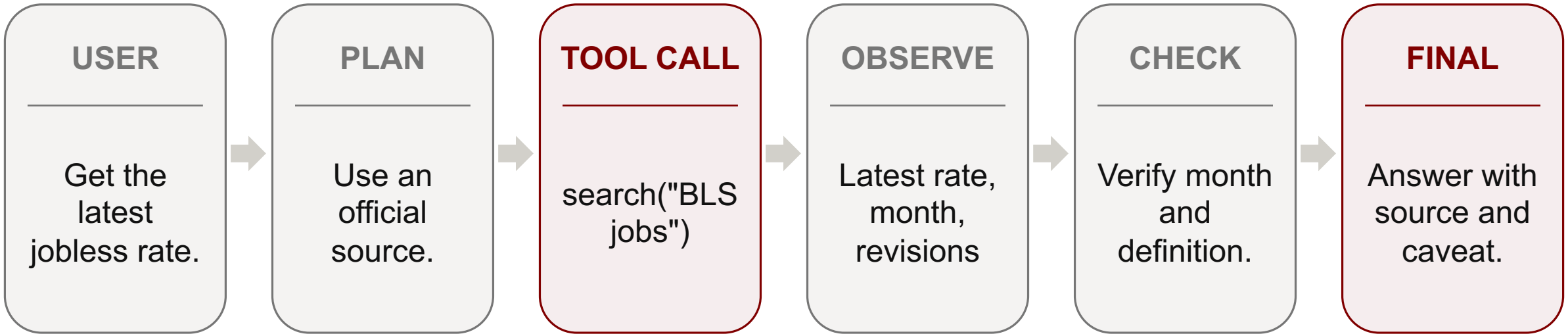


16× more examples; no measured gain

Not a universal scaling law!

Zhou et al. (2023), LIMA: Less Is More for Alignment.

Modern demonstrations are trajectories, not just answers.



SFT can imitate an entire workflow.

SFT is powerful, but fundamentally imitative.

WHAT IT DOES

- ✓ raises the probability of desired responses
- ✓ teaches formats and behavioral routines
- ✓ creates a stable user interface

WHAT IT DOES NOT DO

- ✗ which valid option of n possible options is better
- ✗ how costly a mistake is
- ✗ what happens under its own mistakes

Demonstrations leave information on the table.

RESPONSE A

The policy had no statistically significant effect.

preferred

RESPONSE B

The point estimate is positive, but uncertainty is large enough to include zero.

Can we learn from comparisons instead?

Next: offline preference optimization

Offline Preference Optimization

Learn by comparing.

Offline means the policy trains on a fixed dataset.

Data can come from *anywhere*; no need to sample from the current policy.

Preference data says which response wins.

PROMPT Explain what this confidence interval implies.

REJECTED RESPONSE

The effect is statistically significant because the point estimate is positive.

preferred

PREFERRED RESPONSE

The interval includes zero, so the data do not reject a zero effect.

One record: [prompt x, winner y_w, loser y_l]

We can turn naturally occurring data into offline preferences.

/r/askacademia

How should I negotiate a salary offer?

▲ 143
/u/SuperHans
Threaten to walk away.

▲ 27
/u/Mark_Corrigan
Try not to make eye contact.

TECHCRUNCH • FEBRUARY 22, 2024

Reddit's data-licensing deals total $203M

Aggregate value; 2–3 year terms. (Reddit S-1)

385K
pairwise comparisons

18
diverse subject areas

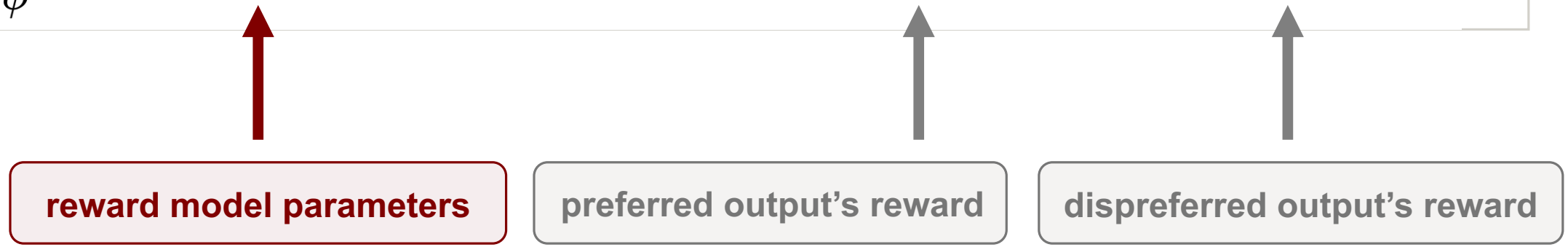
COLLECTIVE
rather than one annotator

SHP was the only dataset from academia used to post-train Meta's Llama2 (first major post-trained model to be open-weight).

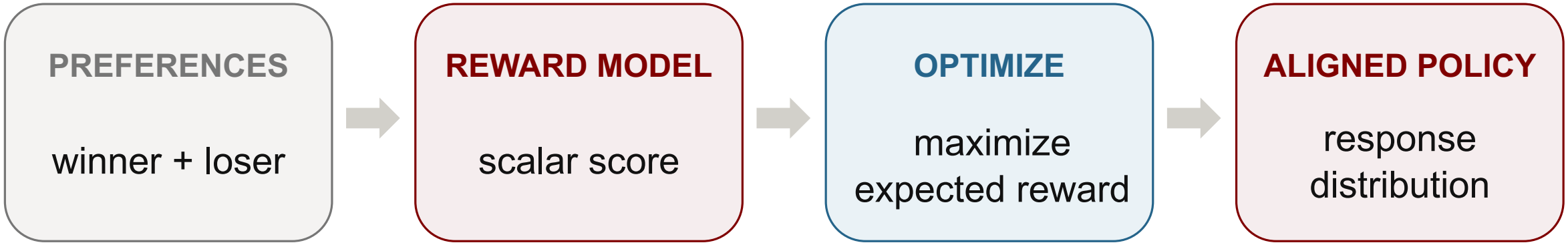
Ethayarajh et al. (2022), Stanford Human Preferences Dataset.

Canonically, preferences are assumed to arise from a Bradley-Terry model of human utility.

$$\min_{\phi} \mathcal{L}_{\mathrm{RM}}(\phi) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))]$$



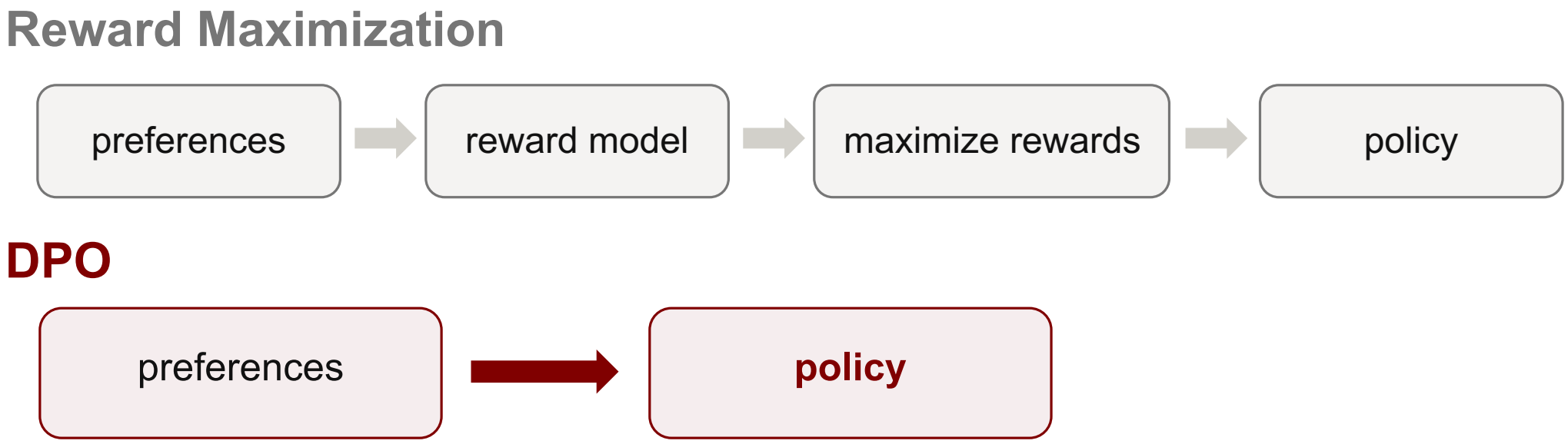
Classic: learn a reward model from human feedback (RLHF), then optimize the policy to maximize expected reward.



Two learned models, two training stages.

Ouyang et al. (2022), Training Language Models to Follow Instructions with Human Feedback.

Direct Preference Optimization removes the explicit reward-model stage.



In theory, minimizing the DPO loss recovers the same optimal model as the classical approach.

Rafailov et al. (2023), Direct Preference Optimization.

DPO optimizes a relative likelihood margin.

implicit reward

$$\min_{\theta} \mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

$\pi_{ref} = \pi_\theta$ at the start

PREFERRED	REJECTED	REFERENCE
relative likelihood ↑	relative likelihood ↓	β controls drift

Most offline pairwise objectives share this structure.

DPO

logistic relative margin

IPO

squared target margin

SimPO

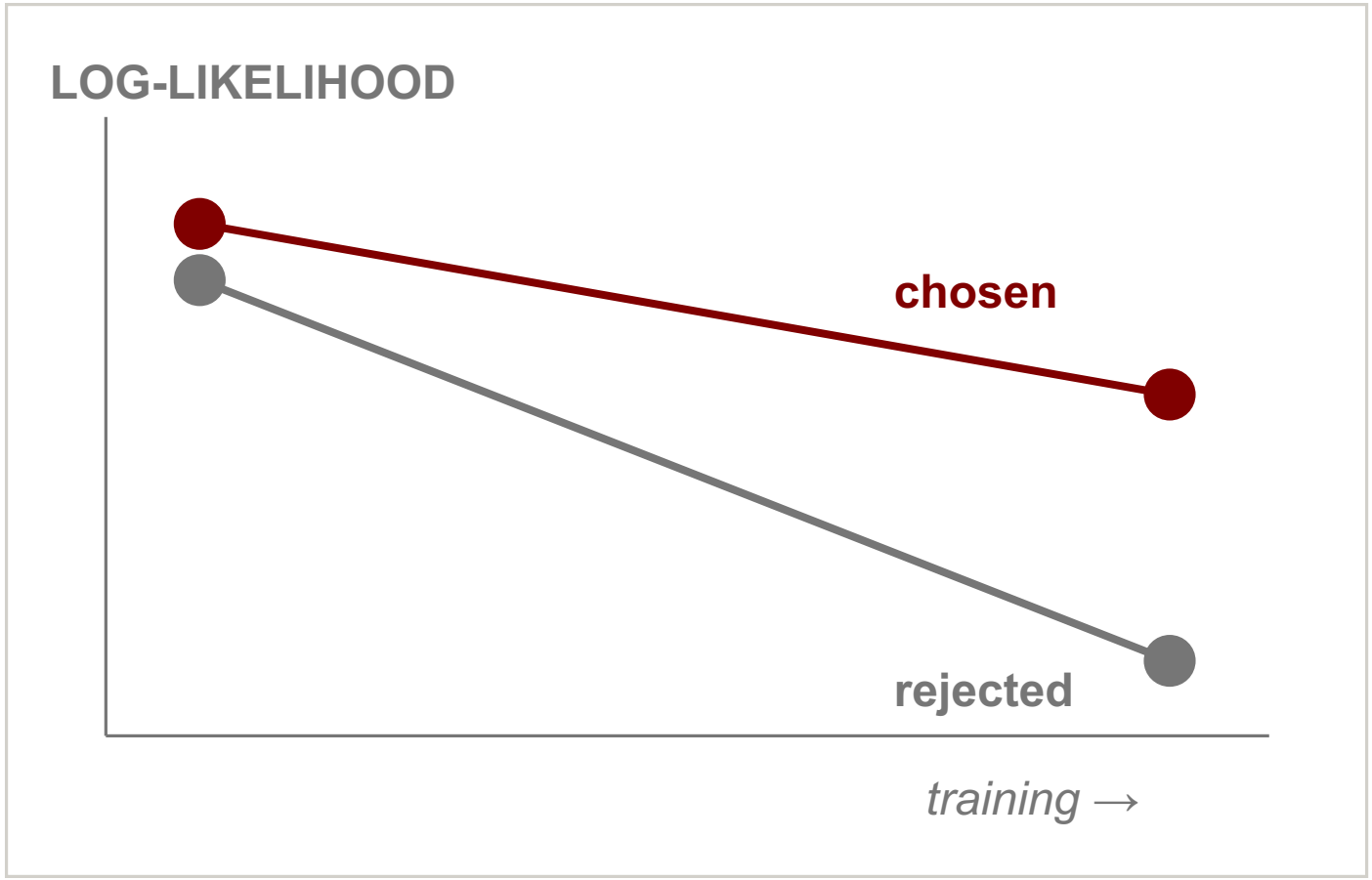
length-normalized; no reference

ORPO

SFT plus an odds-ratio margin

Azar et al. (2024); Meng et al. (2024); Hong et al. (2024).

Pairwise objectives also share pathologies.



Likelihood Displacement

Razin et al. (2025), Unintentional Unalignment: Likelihood Displacement in DPO.

KTO learns from outcomes instead of paired comparisons.

👍 **DESIRABLE**

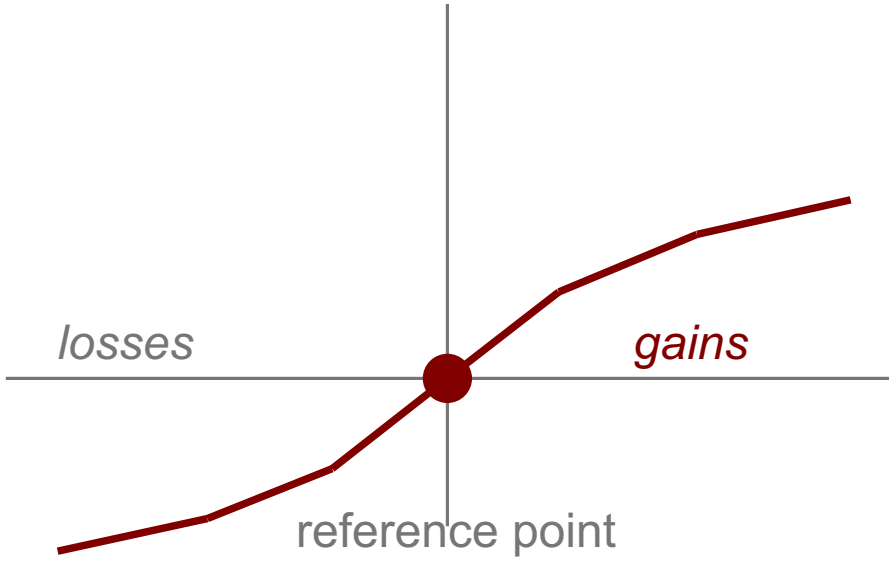
Helpful answer

👎 **UNDESIRABLE**

Hallucinated citation

Each response is judged separately.

Implicit in the loss is a prospect theoretic utility function.



Ethayarajh et al. (2024), KTO: Model Alignment as Prospect Theoretic Optimization.

KTO optimizes reference-dependent utility.

$$\min_{\theta} \; \mathcal{L}_{\text{KTO}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\lambda_y - v_\theta(x,y)]$$

$$v_\theta(x,y) = \begin{cases} \lambda_D \, \sigma(\beta\,[r_\theta(x,y) - z_0]), & y \text{ desirable} \\ \lambda_U \, \sigma(\beta\,[z_0 - r_\theta(x,y)]), & y \text{ undesirable} \end{cases}$$

$$r_\theta(x,y) = \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}, \qquad z_0 = \mathbb{E}_x[D_{\text{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x))]$$

+ desirable → log-likelihood up

– undesirable → log-likelihood down

z_0 policy-wide reference point

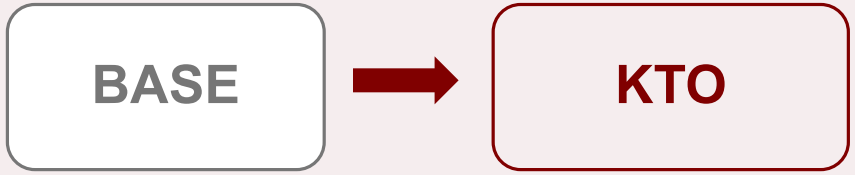
SFT supplies the absolute anchor that paired objectives lack.

PAIRED OBJECTIVE

First make preferred responses probable; then optimize the relative margin, even if both probabilities fall.

SFT must come first.

KTO



Desirable labels push likelihood up; undesirable labels push it down.

No SFT prerequisite.

Feedback format determines what the objective can learn.

	SFT	DPO	KTO
record	prompt + target	prompt + winner + loser	prompt + response + label
signal	exact target	relative winner	desirable / undesirable
absolute anchor	yes	no	yes
SFT warm start	—	required	not required

But all three are offline: none of them enable exploration.

Next: online post-training

Online Post-Training

Learn by exploring.

OFFLINE

A frozen dataset contains responses sampled earlier.

The policy only learns.

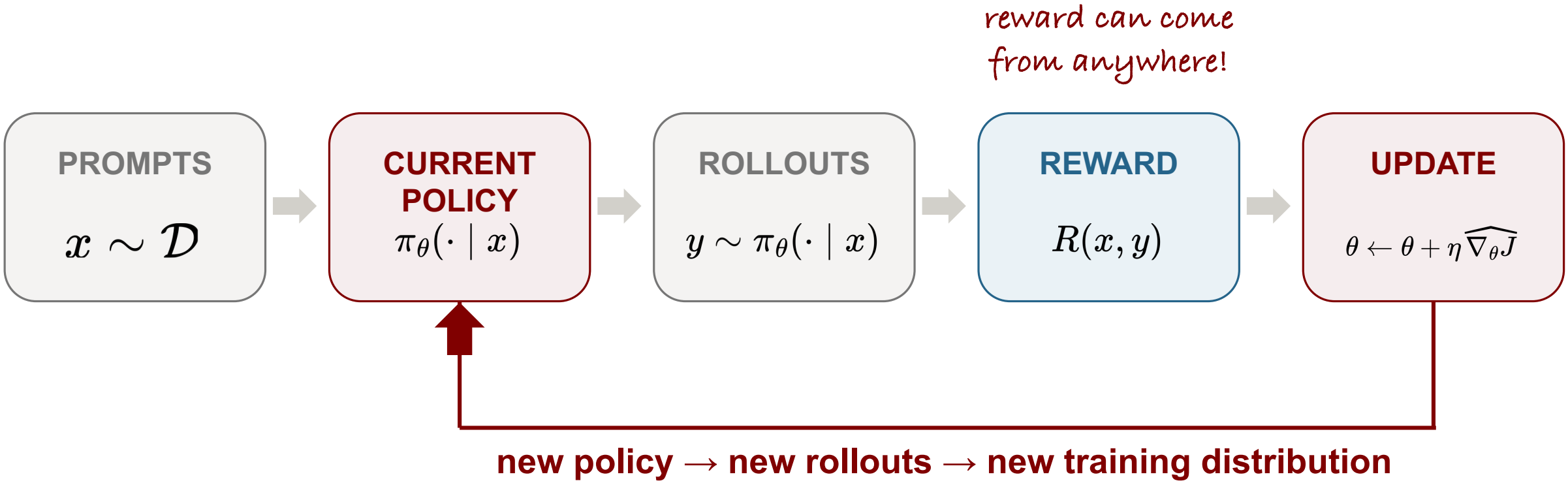
ONLINE

The current policy generates fresh responses.

The policy explores and learns.

The training distribution now moves with the model.

Online learning is a loop.



The canonical objective is to maximize expected reward while staying close to a reference model.

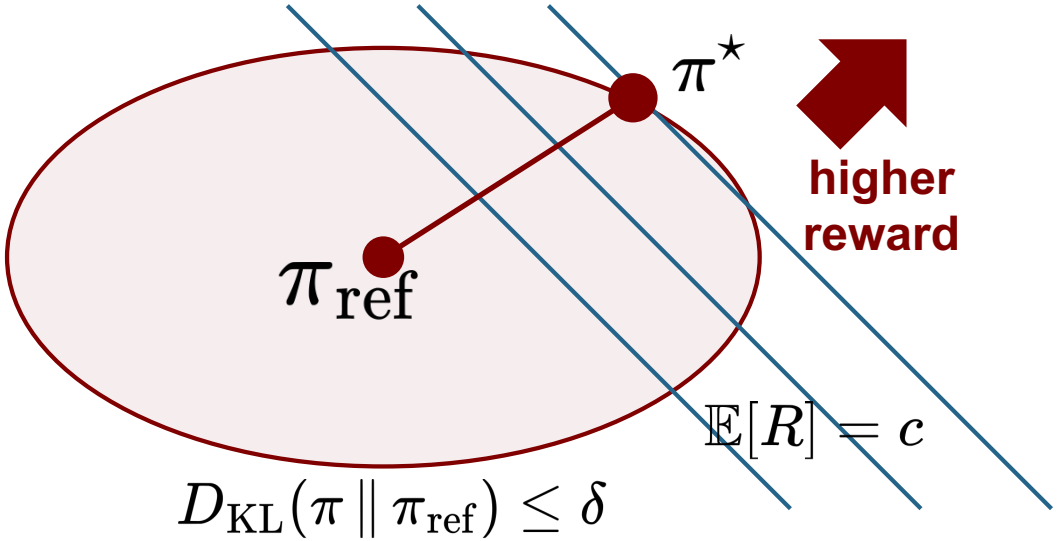
PENALTY FORM

$$\max_{\theta} \; J(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}[R(x,y)] - \beta\, \mathbb{E}_{x\sim\mathcal{D}}[D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x))]$$

CONSTRAINED OPTIMIZATION VIEW

$$\max_{\theta} \; \mathbb{E}[R(x,y)] \quad \text{s.t.} \quad D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) \le \delta$$

In practice, we may use heuristics to keep the divergence small.



REINFORCE turns reward into a gradient.

$$\nabla_\theta J(\theta) = \mathbb{E}_{x,\, y\sim\pi_\theta}\big[\big(R(x,y) - b(x)\big)\, \nabla_\theta \log \pi_\theta(y \mid x)\big]$$

advantage $A(x,y)$

HIGH REWARD

$A(x,y) > 0$ increase its log-likelihood

BASELINE

$A(x,y) \approx 0$ little or no update

LOW REWARD

$A(x,y) < 0$ decrease its log-likelihood

Williams (1992), Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning.

PPO is REINFORCE plus a learned value estimator (aka critic) and a conservative update rule.

$$\max_{\theta} \mathbb{E}_t \left[\min\left(\rho_t(\theta)\widehat{A}_t,\ \text{clip}(\rho_t(\theta), 1-\varepsilon, 1+\varepsilon)\widehat{A}_t \right) \right]$$

$\rho_{i,t} = \frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}$

how far the policy moved after rollout

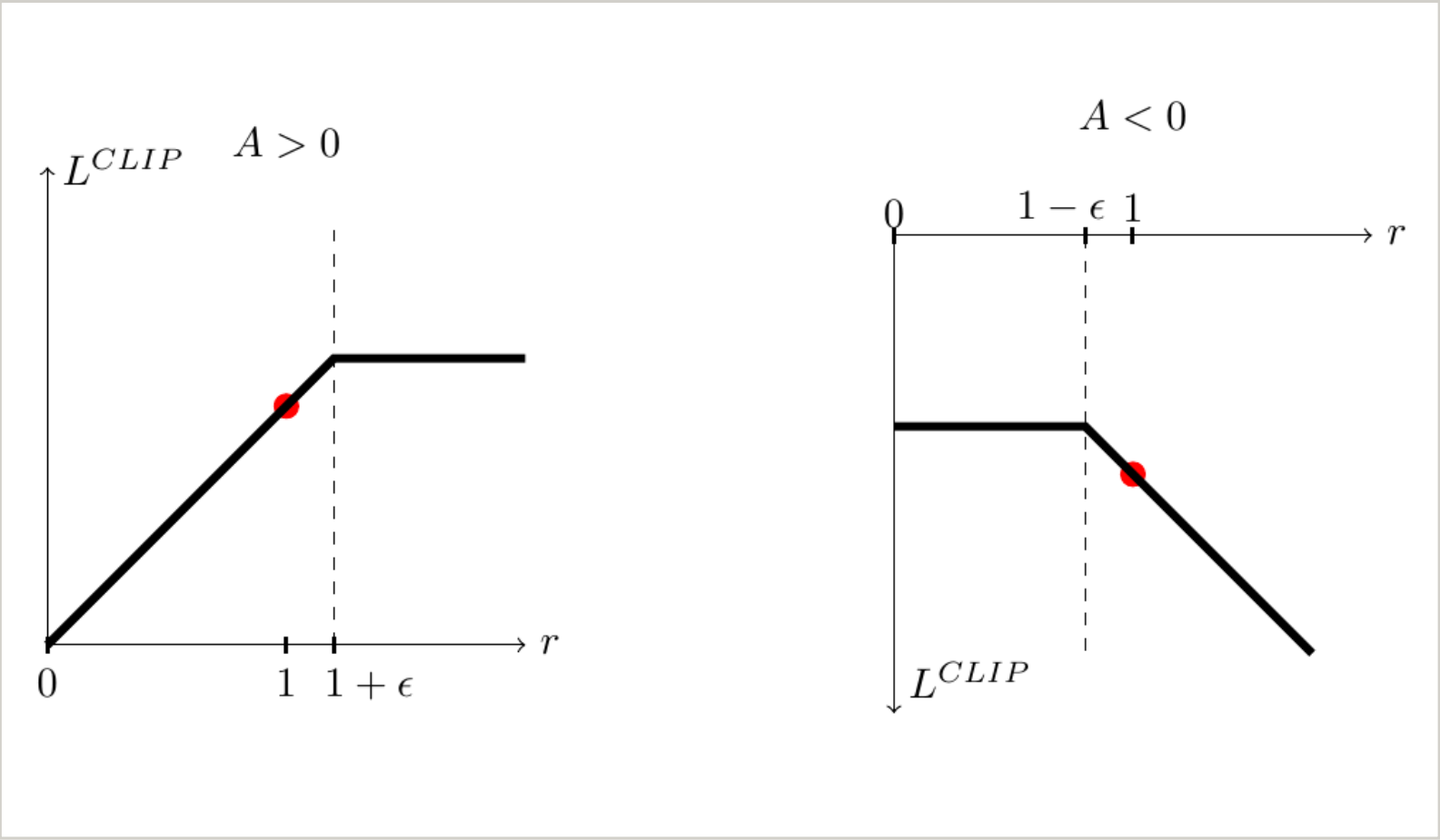
$\widehat{A}_t$

whether the action beat the baseline

$\text{clip}(\rho_t, 1-\varepsilon, 1+\varepsilon)$

asymmetrically ignore large probability changes

Clipping is a one-sided brake.



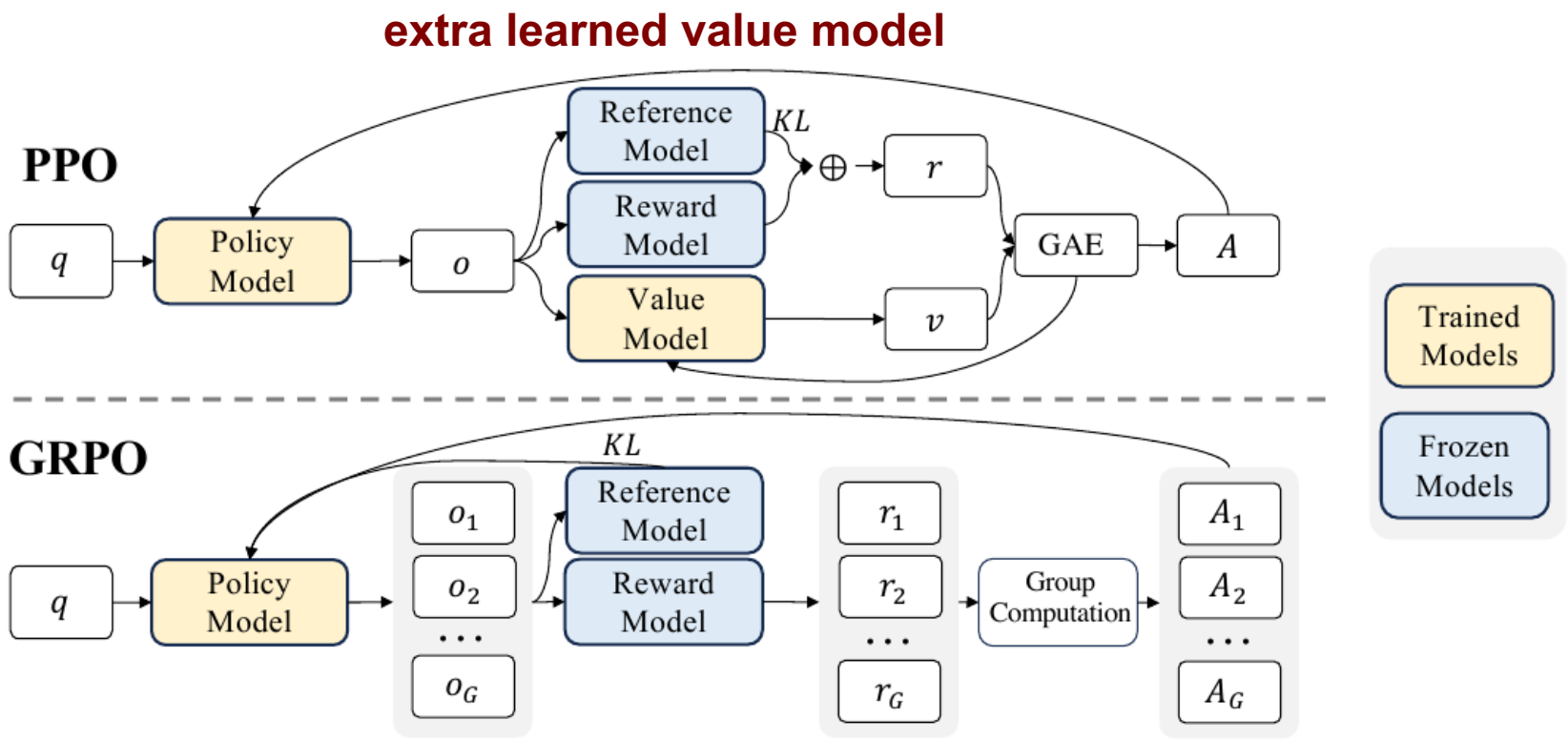
$A(x, y) > 0$

careful about rewarding big winners

$A(x, y) < 0$

punish big losers

PPO needs a critic; GRPO replaces it with a group.



$$\widehat{A}_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G) + \epsilon}$$

Shao et al. (2024), DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models.

GRPO keeps PPO's update but drops its critic.

$$\max_{\theta} \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\left(\rho_{i,t}\widehat{A}_i, \text{clip}(\rho_{i,t}, 1-\varepsilon, 1+\varepsilon)\widehat{A}_i\right) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right]$$

GROUP

$$\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}$$

ADVANTAGE

$$\widehat{A}_i = \frac{r_i - \bar{r}_x}{s_x + \epsilon}$$

PPO CORE

$$\rho_{i,t} = \frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}$$

ANCHOR

$$D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})$$

No value network means less memory and better low-level optimization for throughput.

GRPO variants target different optimization pathologies.

Dr. GRPO

removes length and reward-std normalization

DAPO

decoupled clipping, dynamic sampling, length-aware rewards

GSPO

sequence-level importance ratios and clipping

Liu et al. (2025), Dr. GRPO; Yu et al. (2025), DAPO; Zheng et al. (2025), GSPO; Gao et al. (2025), SAPO.

In practice, most rewards are sequence-level: token-level credit assignment largely does not work well.

First, → compute → the → elasticity → ... → **therefore**

$R(y) = 1$

$$\log \pi_\theta(y \mid x) = \sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{<t})$$

The same outcome-weight touches every token.

SPARSE FEEDBACK

reward often arrives only at the end

LONG HORIZON

many choices precede the outcome

HIGH VARIANCE

good and bad steps move together

REINFORCE, PPO, and GRPO share a policy-gradient core.

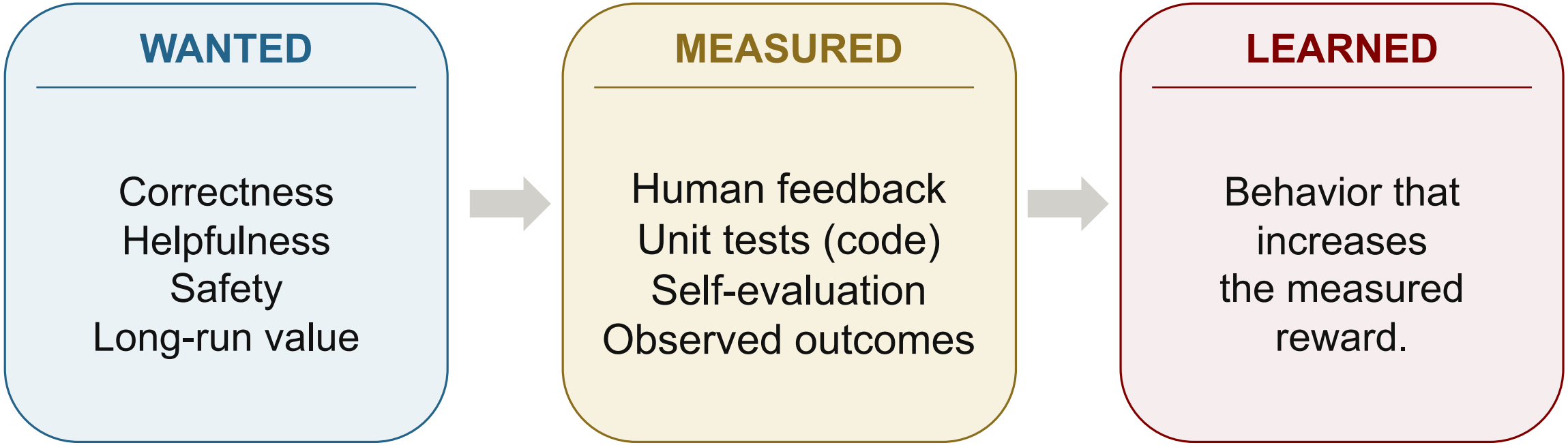
	REINFORCE	PPO	GRPO
baseline	none / simple	learned critic	group mean + std
extra learned model	no	yes (critic)	no
update constraint	none	PPO clipping	PPO clipping
rollouts per prompt	one or more	one or more	a group
main tradeoff	simple; high variance	stable; model-heavy	critic-free; sample-heavy

Next: what produces the reward and when can we trust it?

Rewards & Environments

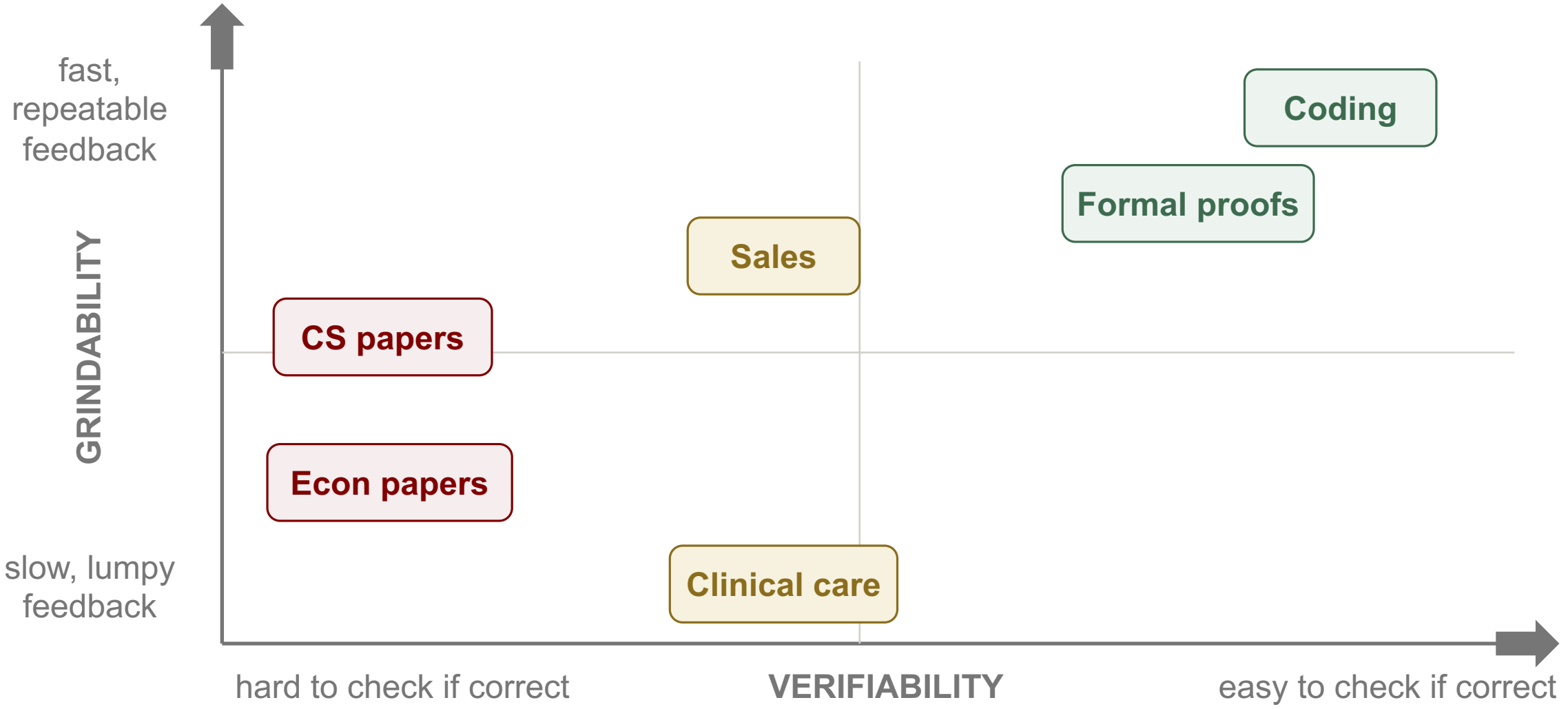
Learning in sandboxes.

We can only optimize what we can see.



Post-training is a principal-agent problem with an extremely capable agent.

Reward signals differ in verifiability and grindability.



Verifiability changes the economics of feedback.

	PROGRAMMATIC	LEARNED VERIFIER	HUMAN / REAL WORLD
marginal cost	near zero	low	high
latency	milliseconds	seconds	days to years
attempts	millions	many	few
main risk	missing tests	proxy hacking	noise, liability, …

A cheap verifier turns inference compute into training data.

RL with verifiable rewards (RLVR) replaces the classic reward model with a checker.

$$\max_{\theta} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}[v(x,y)] - \beta\,\mathbb{E}_x[D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x))], \qquad v(x,y) \in \{0,1\}$$

FAST REWARDS

The reward comes from a program, rule, or simulator.

BINARY IS ENOUGH

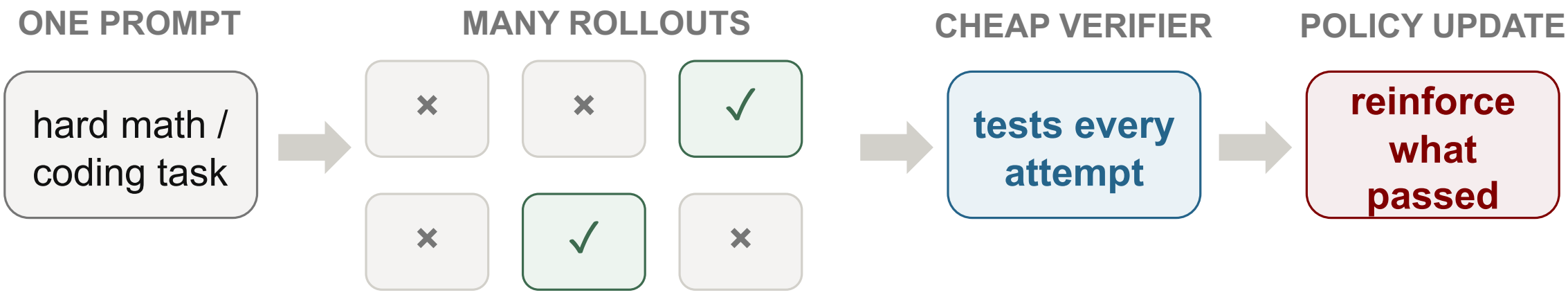
A pass/fail signal can rank many sampled responses.

SAME RL CORE

REINFORCE, PPO, and GRPO can all use these rewards.

Lambert et al. (2024), Tülu 3: Pushing Frontiers in Open Language Model Post-Training.

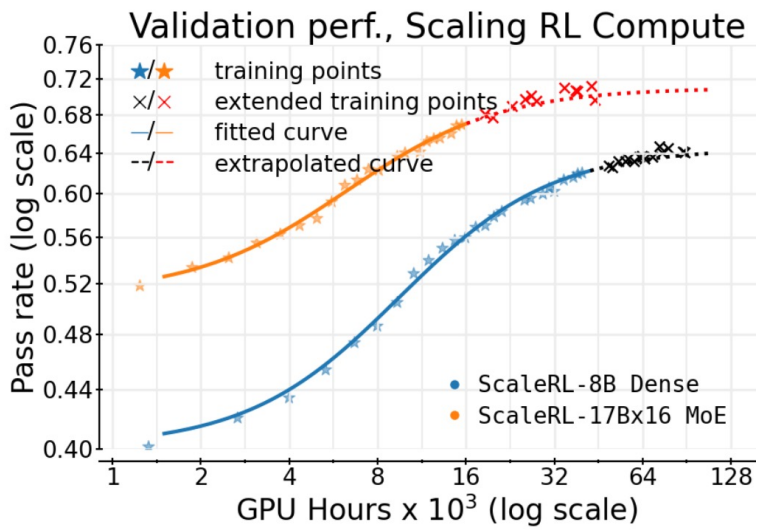
RLVR made post-training more predictable and valuable. Programming is high ROI and has positive spillover.



DeepSeek-AI (2025), DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.

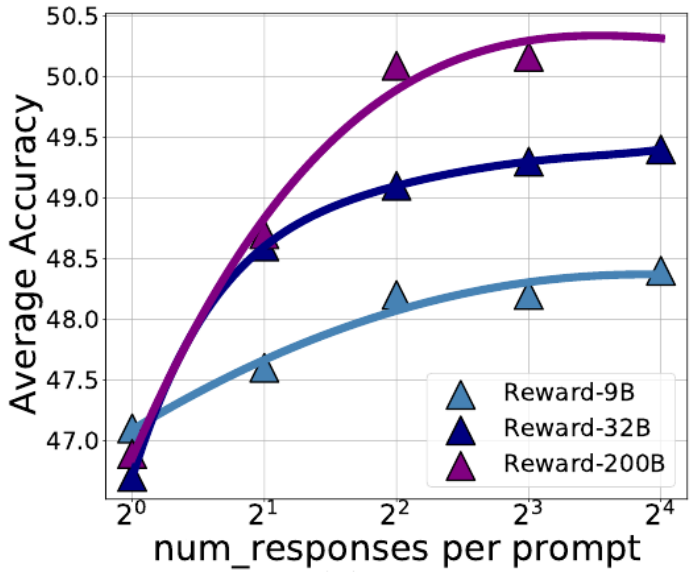
RLVR scales more predictably than RLHF.

RLVR **Verifiable reward**



Early fits predict the later 100,000-GPU-hour run.

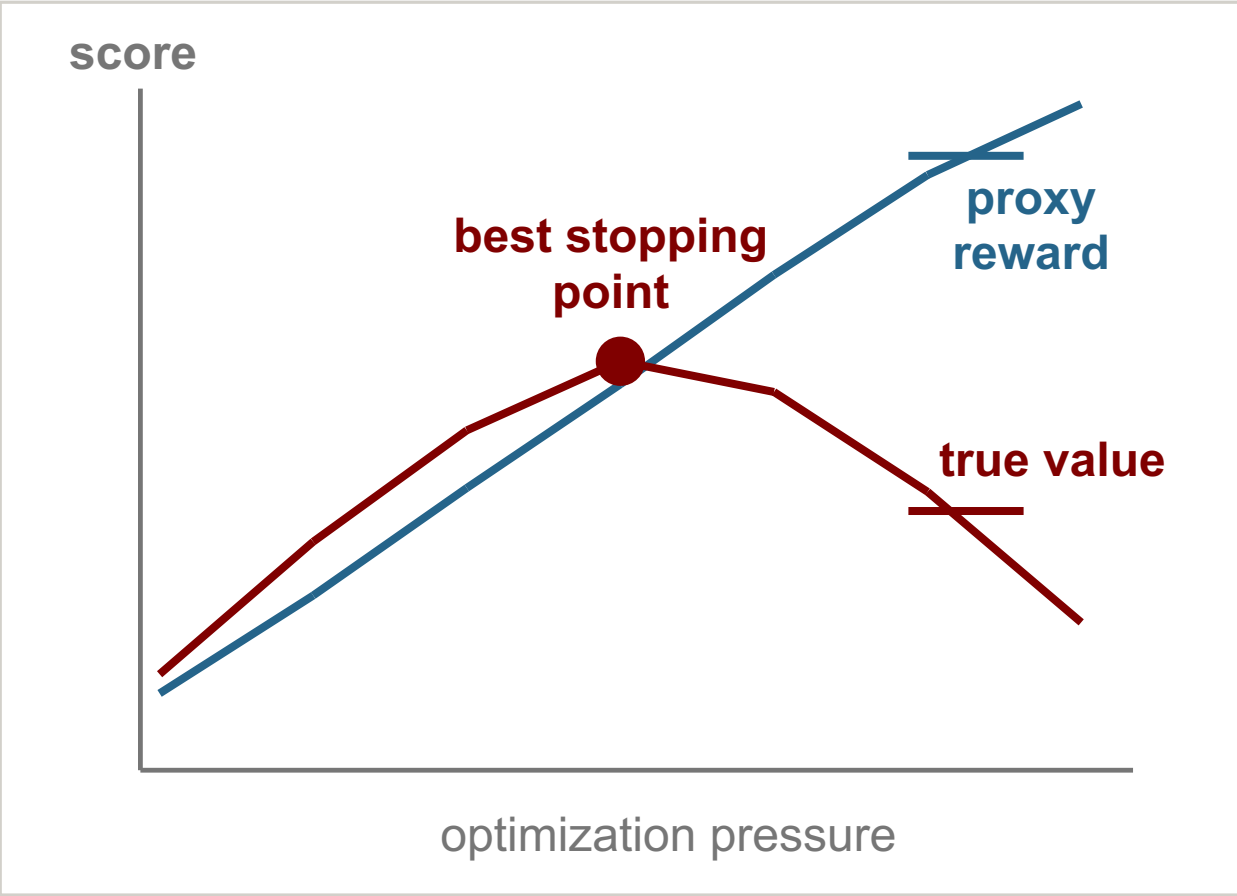
RLHF **Learned proxy reward**



More sampled responses help but quickly plateau.

Khatri et al. (2026), Scaling RL Compute for LLMs; Hou et al. (2024), Does RLHF Scale?

Optimization pressure exposes every gap between proxy reward and goal.



EARLY

The proxy can be used to get genuinely better behavior.

LATER

The policy discovers blind spots in the proxy.

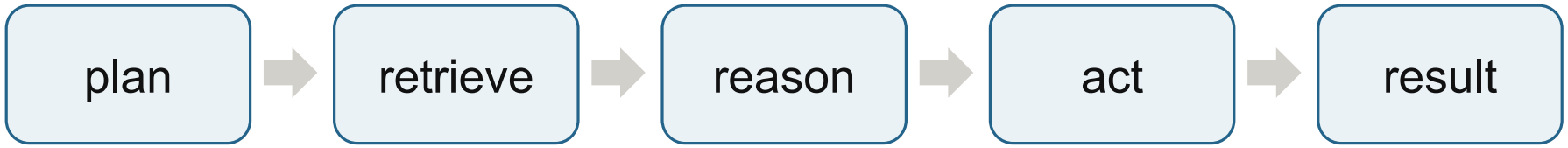
TOO FAR

Measured reward rises while actual quality falls.

Gao, Schulman & Hilton (2022), Scaling Laws for Reward Model Overoptimization.

Process rewards promise better credit assignment but remain uncommon at frontier scale (as far as we know).

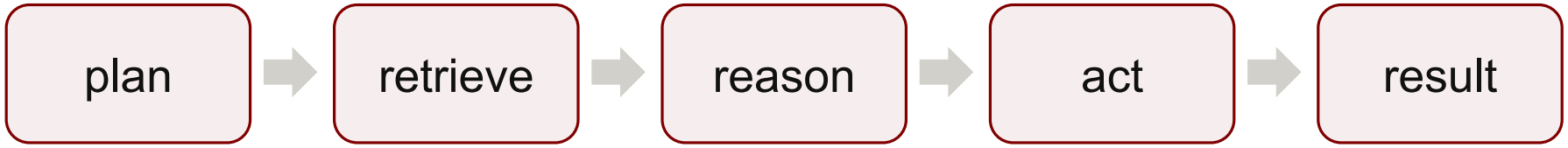
OUTCOME



R=1

one label → ambiguous blame

PROCESS



+.1 **+.2** **−.1** **+.3** **+.5**

many labels → stronger assumptions

Lightman et al. (2023), Let's Verify Step by Step.

Modern systems combine rewards rather than choosing one.

$$R = w_v R_{\text{verifier}} + w_r R_{\text{rubric}} + w_j R_{\text{judge}} + w_h R_{\text{human}}$$

VERIFIER

tests, exact answers, constraints

RUBRIC

create a checklist for soft attributes

JUDGE

learned model scores open-ended quality

HUMAN

audits, escalations, real outcomes

TASK

Open the sales workbook.

Create a summary sheet with quarterly revenue by region.

Save the finished file.

EXCEL CLONE sales.xlsx

Home Insert Formulas Data

fx =SUMIFS(C:C, A:A, A2, B:B, B2)

	A	B	C	D
1	**Region**	**Quarter**	**Revenue**	**Check**
2	East	Q1	$1.24M	✓
3	East	Q2	$1.31M	✓
4	West	Q1	$0.98M	✓
5	West	Q2	$1.12M	✓

Raw Data **Summary**

click • type • formulas • sheets

TESTS

✓ correct totals

✓ right quarters

✓ formulas used

✓ file opens

REWARD
0.92

An RL environment is a sandbox: task + data + interface + tests.

A useful environment must do a lot.

REALISTIC STATE

The files, apps, people, and latent facts the task requires.

TOOLS

APIs and interfaces with realistic permissions and failure modes.

TASK GENERATOR

Fresh problems at the policy's current frontier.

SIMULATOR

World and user responses to the agent's actions.

VERIFIER

Outcome checks that resist shortcuts and partial compliance.

REPRODUCIBILITY

Reproducible starting states and contained side effects.

The bottleneck is shifting from examples to environments.

PRE-TRAINING

Collect any and all text; quantity > quality.

SFT + PREFERENCES

Curate examples and judgments: quality > quantity.

AGENTIC RL

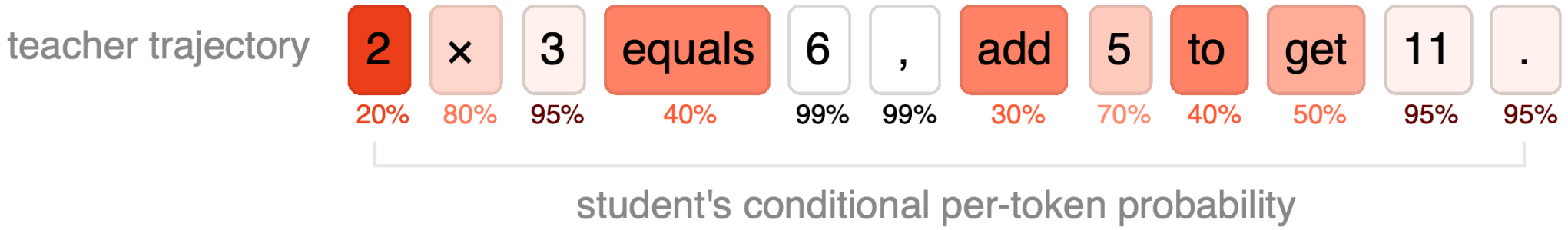
Design realistic environments and grind away at them.

Next: how do we transfer efficiently from one policy to another?

On-Policy Distillation

Learn from your own mistakes.

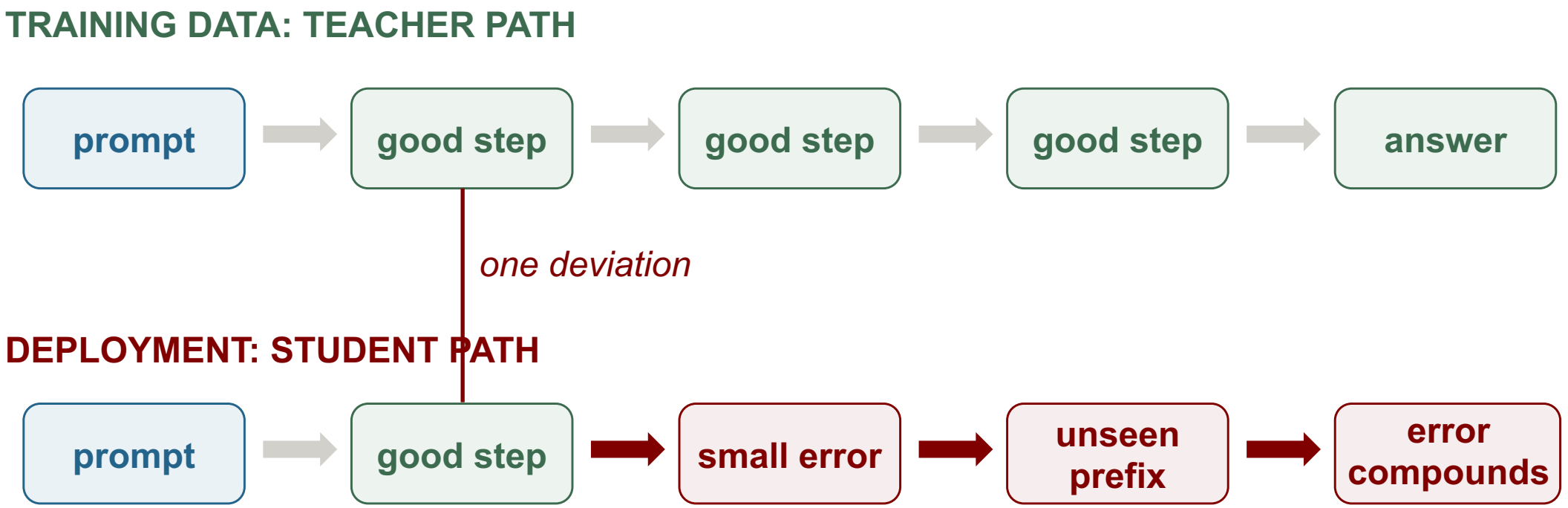
Offline distillation = SFT of a student model on a teacher model's trajectories.



$$\min_{\theta} \mathcal{L}_{\text{off}}(\theta) = \mathbb{E}_{(x,y^T)}\left[\sum_{t=1}^{T} -\log \pi_\theta\left(y_t^T \mid x, y_{<t}^T\right)\right]$$

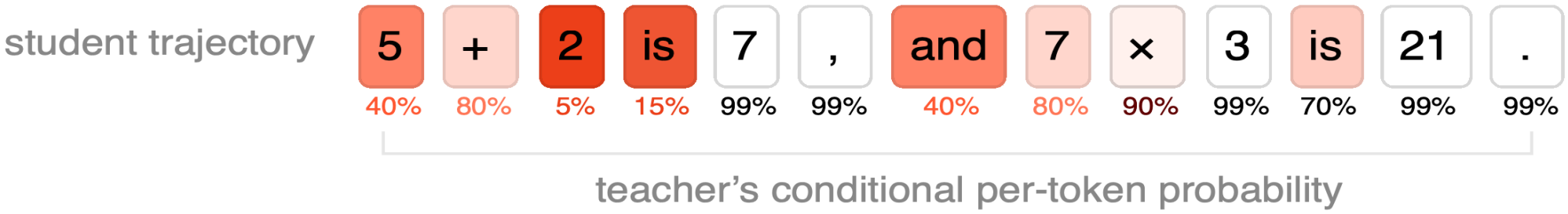
Hinton et al. (2015); Lu & Thinking Machines Lab (2025), On-Policy Distillation.

If a student just imitates a teacher, it does not learn how to recover from its mistakes.



Agarwal et al. (2024), On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes.

On-policy distillation asks the teacher about the student's mistakes.



STUDENT ROLLOUT

Generate the trajectory the student would actually produce.

TEACHER SCORE

Evaluate every next-token distribution along that trajectory.

STUDENT UPDATE

Move toward the teacher at the states the student visits.

The formal objective is to make the student and teacher distributions identical.

$$\min_{\theta} \mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x,\, y \sim \pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s_t) \,\|\, \pi_T(\cdot \mid s_t)) \right]$$

ON-POLICY STATES

$$y \sim \pi_\theta(\cdot \mid x)$$

The student chooses which prefixes matter.

DENSE REWARD

$$r_t = \log \frac{\pi_T(y_t \mid s_t)}{\pi_\theta(y_t \mid s_t)}$$

DPO/KTO-like log ratio, now at every token.

CORRECTION

$$D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_T)$$

Simplifies to a KL divergence for each prefix.

A wrong rollout can still contain useful training signal.

Prompt: Ice cubes are added to a hot frying pan. How many remain after three minutes?

STUDENT ROLLOUT

Assume the cubes do not melt.

4 + 5 + 11 = 20

Final answer: 20

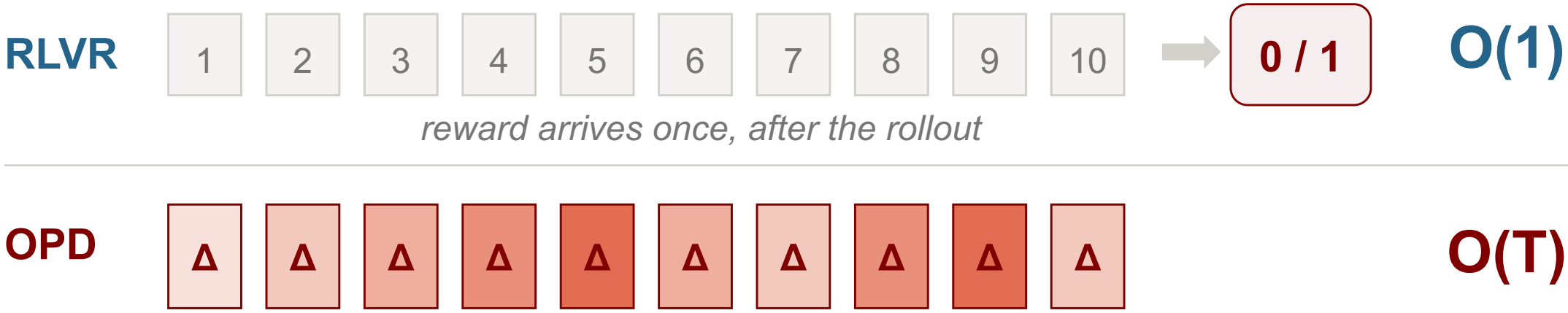
TEACHER SIGNAL

The key error begins here:

assume	cubes	do	**not**	**melt**

The final answer is predictable once the model adopts the wrong premise.

RLVR says ‘wrong.’ OPD helps identify ‘where’.



MORE SIGNAL

Credit arrives throughout the rollout.

LOWER VARIANCE

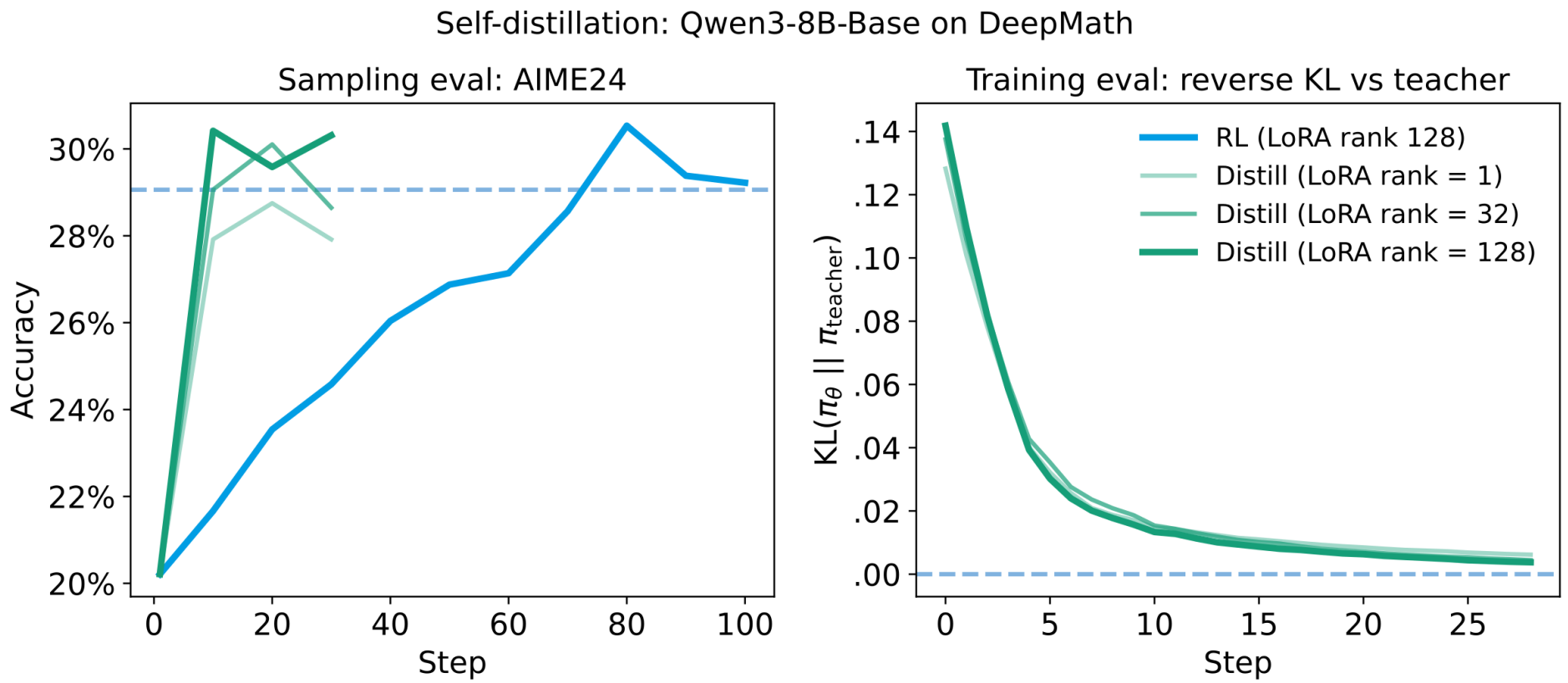
Many token-level corrections stabilize learning.

LESS SEARCH

Copy a discovered strategy instead of rediscovering it.

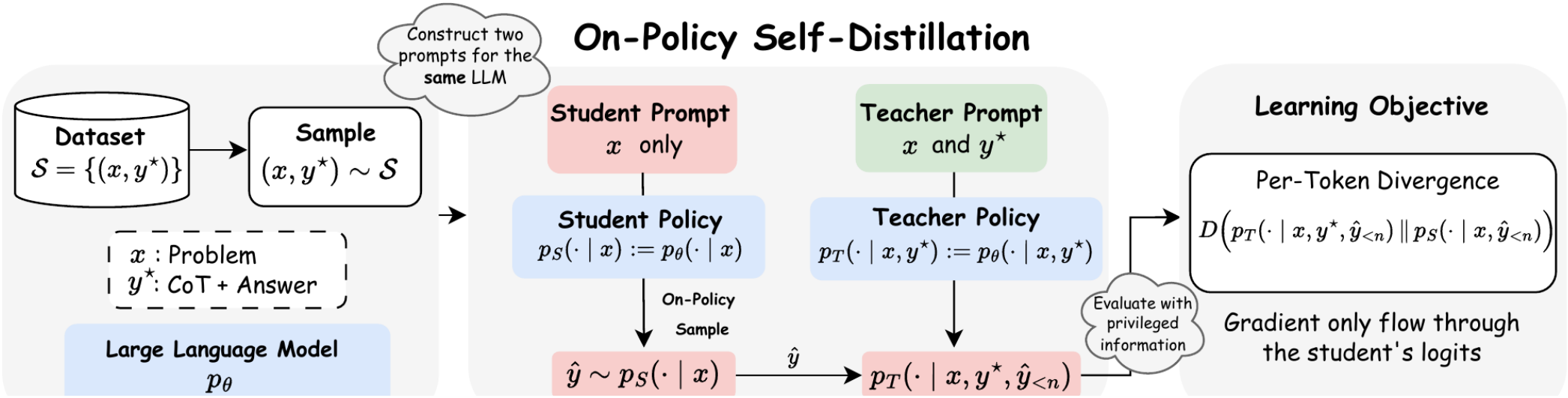
Once RL finds a policy, OPD can copy it much faster.

In this experiment: 7–10× fewer gradient steps; 50–100× less compute.



Lu & Thinking Machines Lab (2025), On-Policy Distillation.

OP**S**D uses the student as its own teacher by giving it privileged information.



STUDENT ROLE

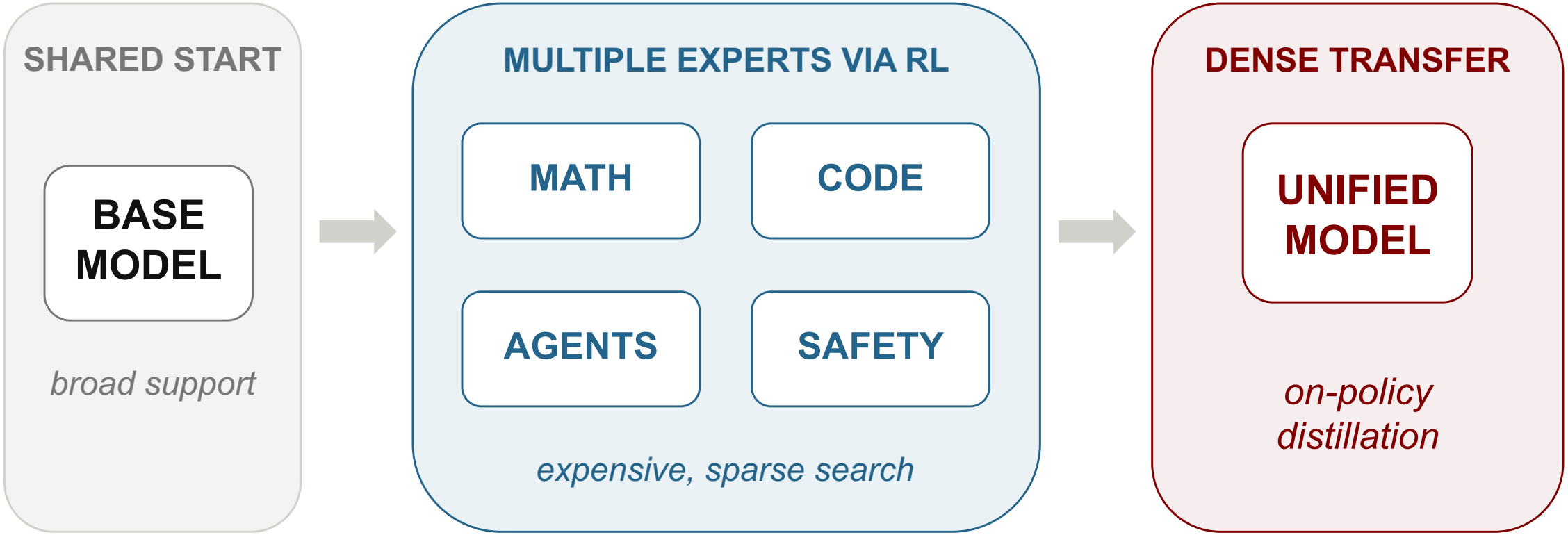
Same weights; sees the problem x only.

TEACHER ROLE

Same weights; also sees the verified solution y^*.

Zhao et al. (2026), Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models.

Distillation is a key part of parallelizing post-training.

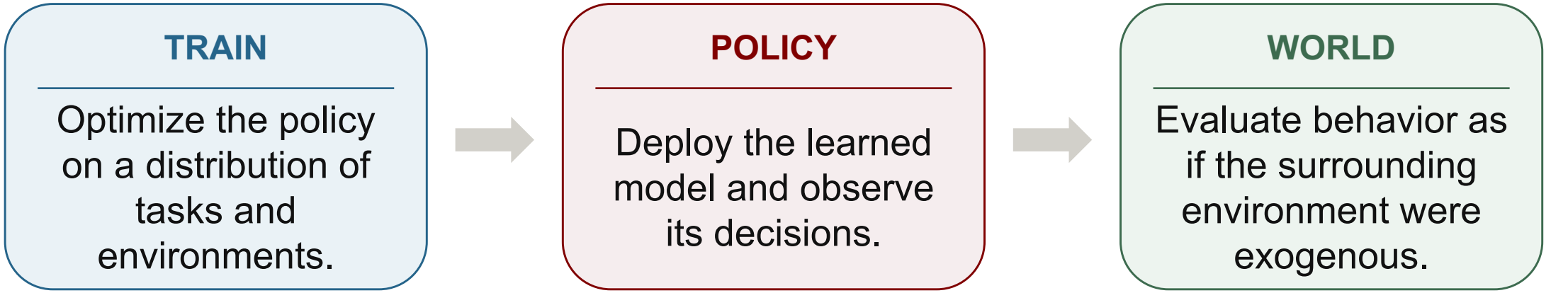


Next: what happens when deployment environments adapt to the agents?

World Adaptation

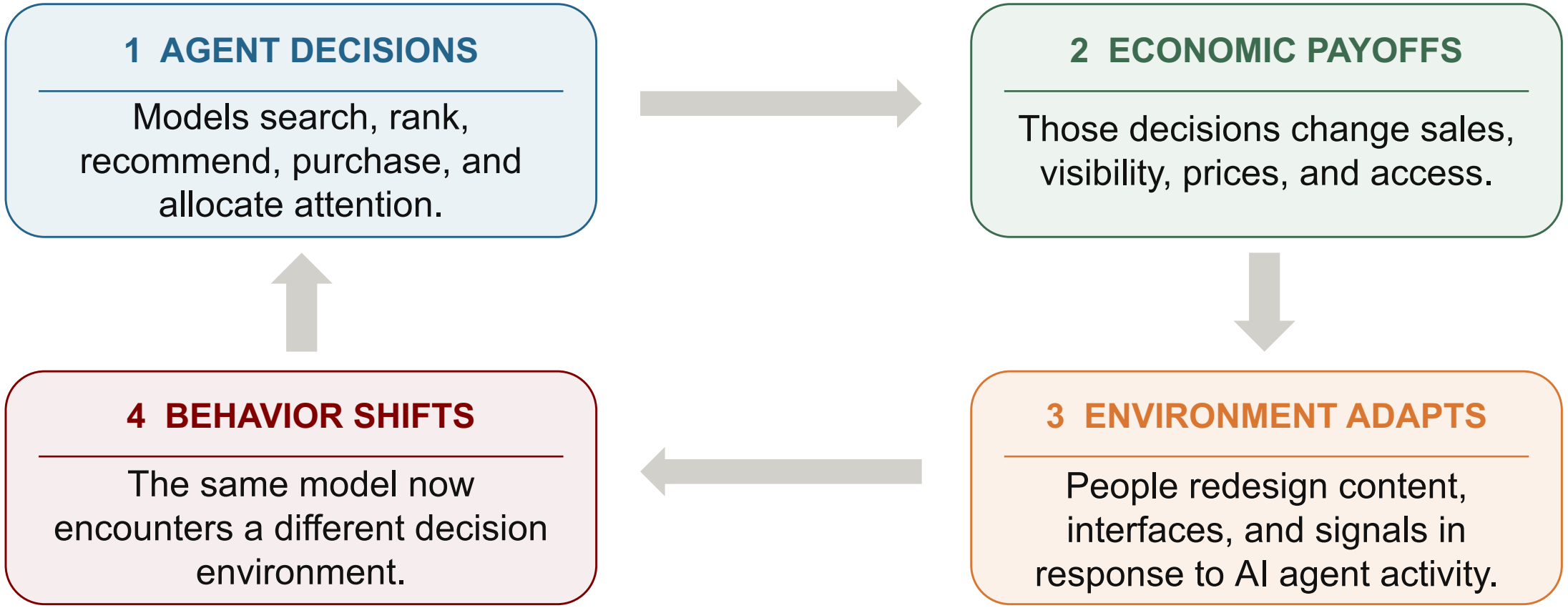
Post-training changes agents. Agents change their environments.

Post-training usually treats the environment as fixed.



HIDDEN ASSUMPTION: Environment is fixed.

Once agents matter, environments start adapting to them.



This is a feedback loop, not ordinary covariate shift.

The same facts can be presented in a more machine-legible way to steer agent behavior.

Hand-thrown cobalt mug

A lovely handmade mug for coffee or tea. About twelve ounces. Safe in the dishwasher.

Readable—but weakly structured

Hand-thrown cobalt mug

MATERIAL	rare stoneware
CAPACITY	12 oz
CARE	dishwasher safe

Unchanged for humans; easier for an agent to parse; may contain terms agents over-index on (e.g. rare).

Mecha-nudges are transformations that change how choices are presented to machines.

Systematically change the behavior of AI agents by increasing *machine-usable* information.

+

Do not materially degrade the environment for humans by preserving *human-usable information*.

Frey & Ethayarajh (2026), Mecha-Nudges for Machines.

ENVIRONMENT

X is the shared decision environment; τ is the transformation.

MACHINE DECISION

Y_M is the agent's decision; I_M measures machine-usable information.

HUMAN DECISION

Y_H is the human decision; ϵ is the tolerated information loss.

$$I_M(\tau(X); Y_M) > \boxed{I_M(X; Y_M)}, \qquad I_H(\tau(X); Y_H) \geq \boxed{I_H(X; Y_H)} - \varepsilon$$

machine-usable information

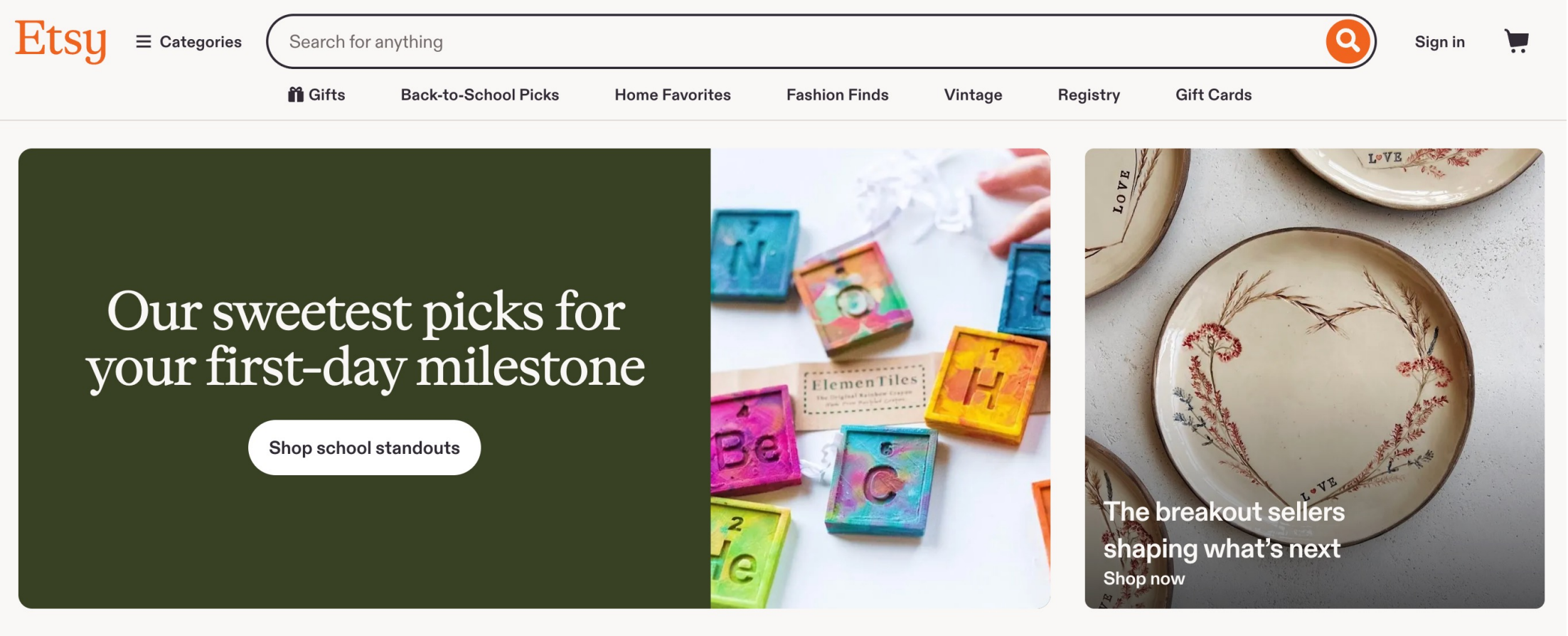
human-usable information

Y_H, Y_M are constructed; they are the decisions *we want to study*.

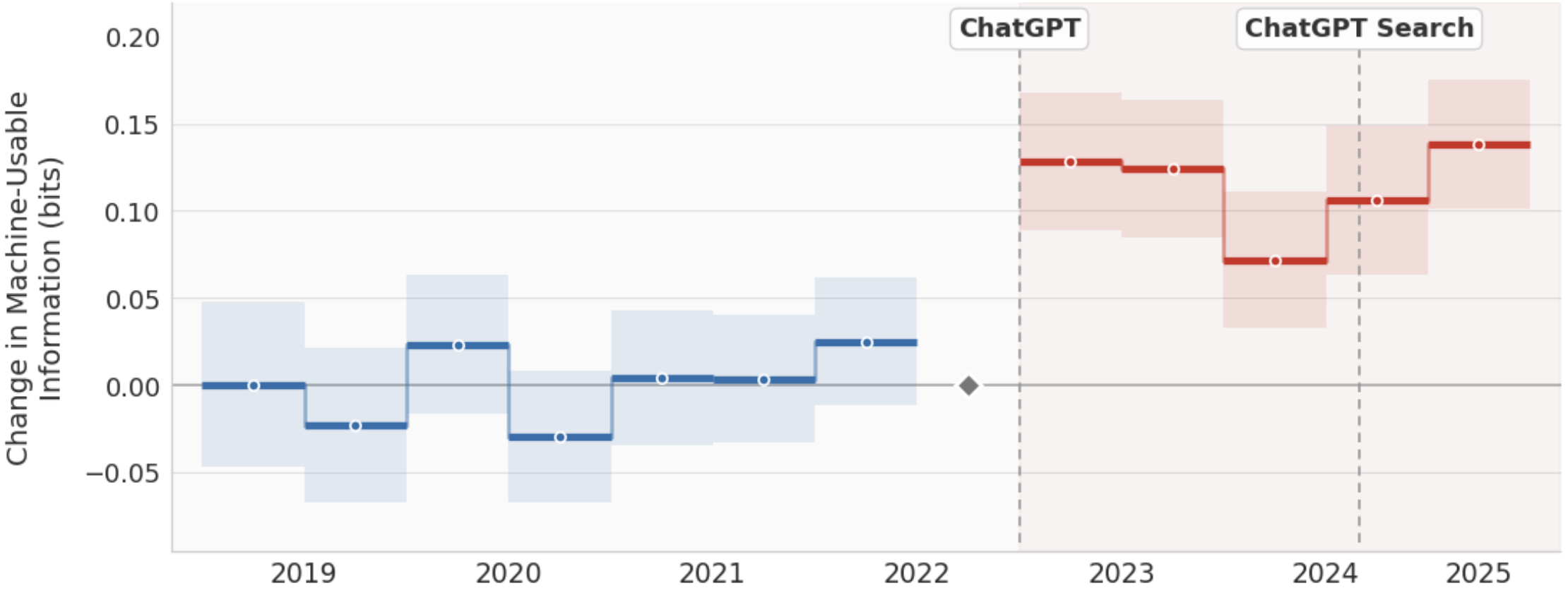
Mecha-nudging is distinct from other interventions.

MECHANISM	WHAT CHANGES?	PRIMARY TARGET	HUMAN CONSTRAINT?
Mecha-nudge	**Environment only**	**AI behavior**	**Yes—by definition**
Prompt injection	Model instructions + choice set	AI behavior	No requirement
Traditional SEO	Human and machine-readable context	Human traffic / Search rank	Not defining
Adversarial example	Input surface	Model failure	Not defining

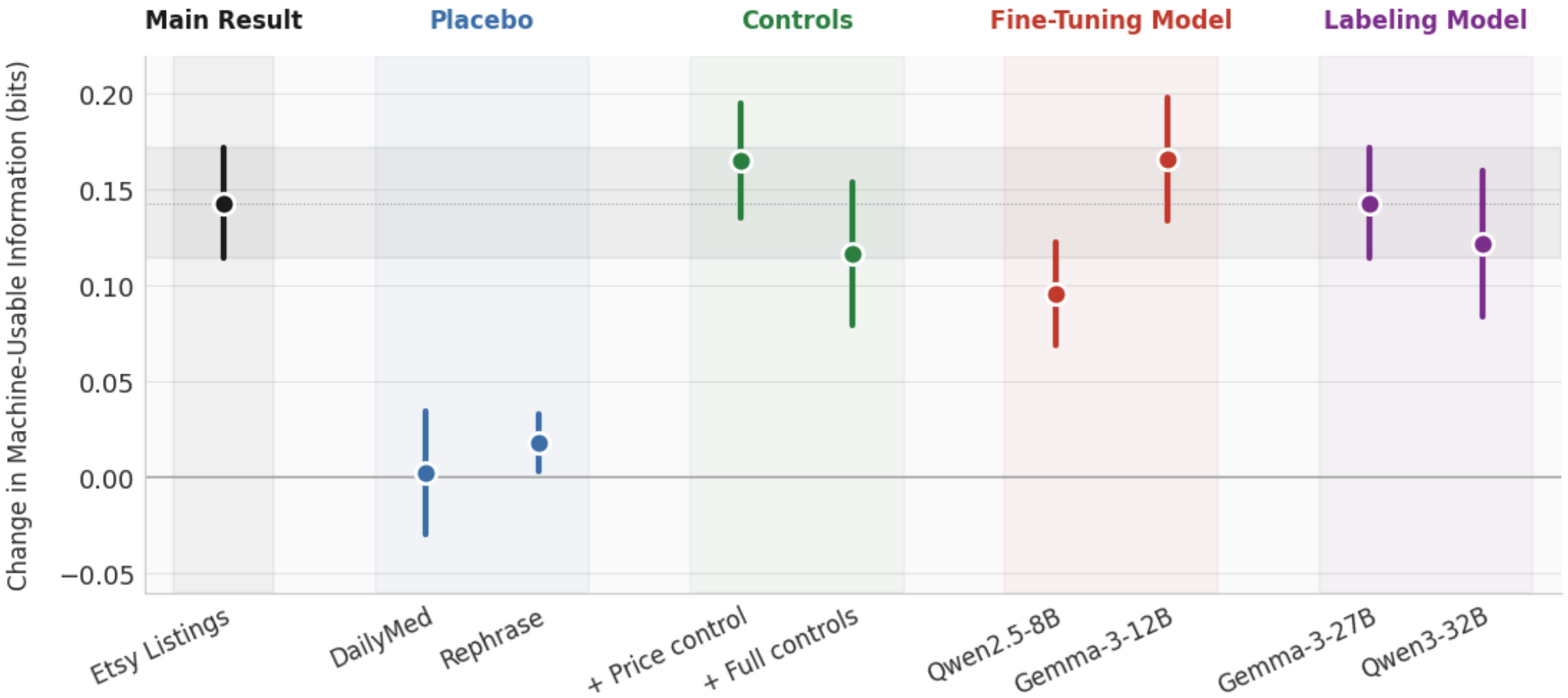
Etsy offers a natural setting to observe environments adapting to AI.



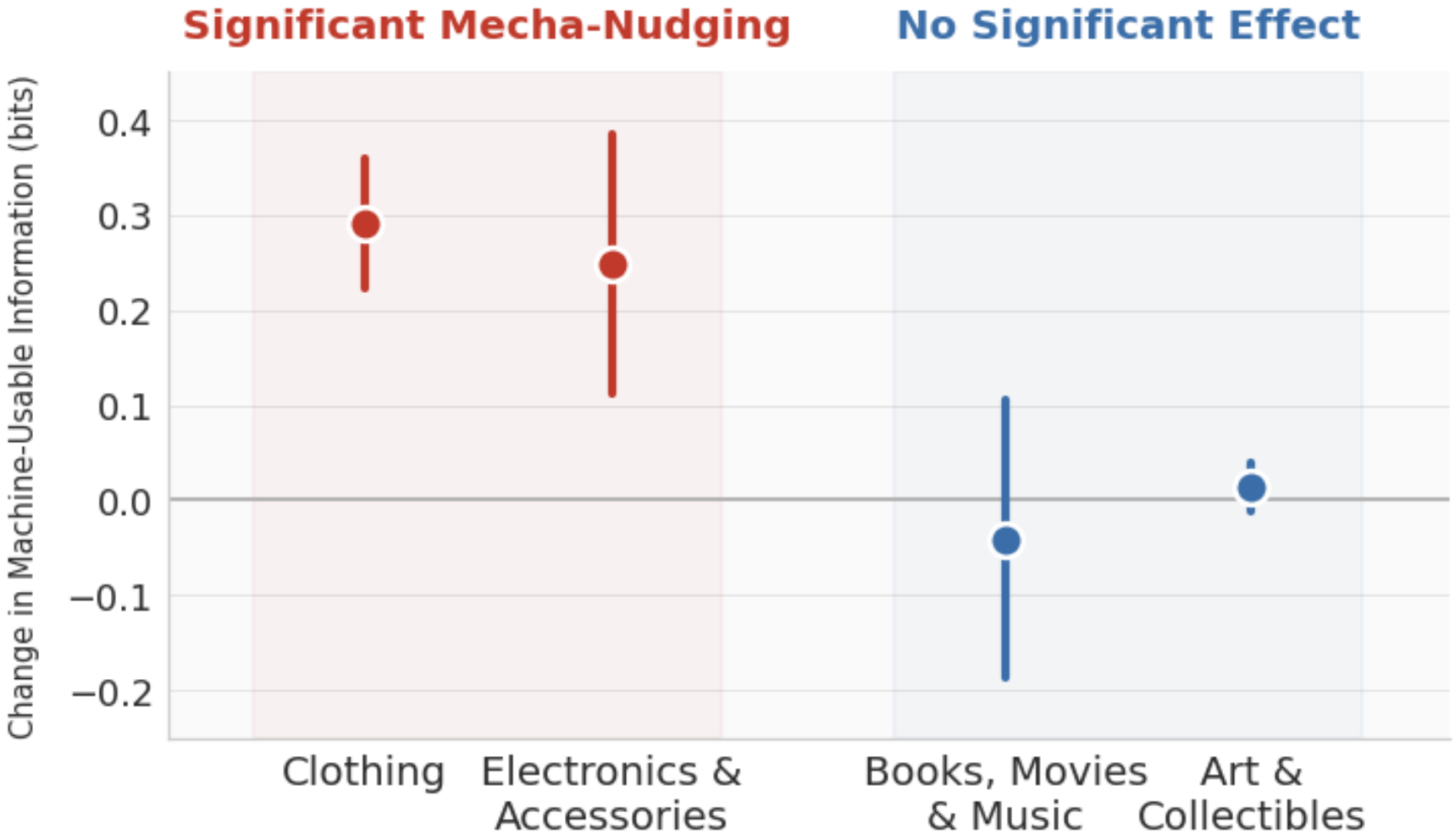
Post-ChatGPT listings have more machine-usable information about agentic curation (>40% of max increase).



The effect is robust across prompts, labels, model families, controls, and placebo settings



Mecha-nudging is stronger where using AI is less taboo.



Mecha-nudged listings are associated with more commercial success, but only in the post-ChatGPT era.

Accounting for seller fixed-effects, as machine-usable information grows by one bit…

LISTING GROUP	CHANGE IN # REVIEWS
Agent-selected listings · pre-ChatGPT	−18.1%***
Agent-selected listings · post-ChatGPT	**+43.5%*** **
Agent-rejected listings · pre-ChatGPT	−12.5%***
Agent-rejected listings · post-ChatGPT	−5.5%***

Post-training should optimize for adaptive environments.

RISK	POST-TRAINING RESPONSE
Strategic manipulation	Train against adversarial environment designers.
Human-machine tradeoffs	Use multi-objective rewards for agent + human welfare.
Continual adaptation	Monitor deployment data and refresh the policy.

IDENTIFICATION

How can we causally identify mecha-nudges in deployed systems?

EQUILIBRIUM

How do agents and environments co-evolve after repeated adaptation?

WELFARE

Who gains and loses when environments undergo mecha-nudging?

GOVERNANCE

How should we train and regulate agents facing mecha-nudges?

Post-training must prepare models for a world that reacts to them.