Post-training

Uh oh, we're out of training data, and SFT datasets scale linearly with people

Required

Rafailov et al., Direct Preference Optimization · arXiv:2305.18290

DeepSeek-AI, DeepSeek-R1 · arXiv:2501.12948

Optional

Ouyang et al., InstructGPT · arXiv:2203.02155 Shao et al., DeepSeekMath · arXiv:2402.03300

Ethayarajh et al., KTO · arXiv:2402.01306

Bai et al., Constitutional AI · arXiv:2212.08073 Lambert et al., Tulu 3 · arXiv:2411.15124

Christoffer Heckman

Department of Computer Science, University of Colorado Boulder

October 29, 2026

Quick overview

A reward from preferences. Preferences vs facts.

People preferred a small model trained on their votes over one a hundred times larger trained to match answers

The leash. Constraints on preferences by fitting reward models.

Four copies of the model sit in GPU memory at once, and the policy still learns to game the reward

No reward model. Skipping the reward modeling for efficiency.

The policy's own probabilities stand in for the reward, so nothing extra is trained

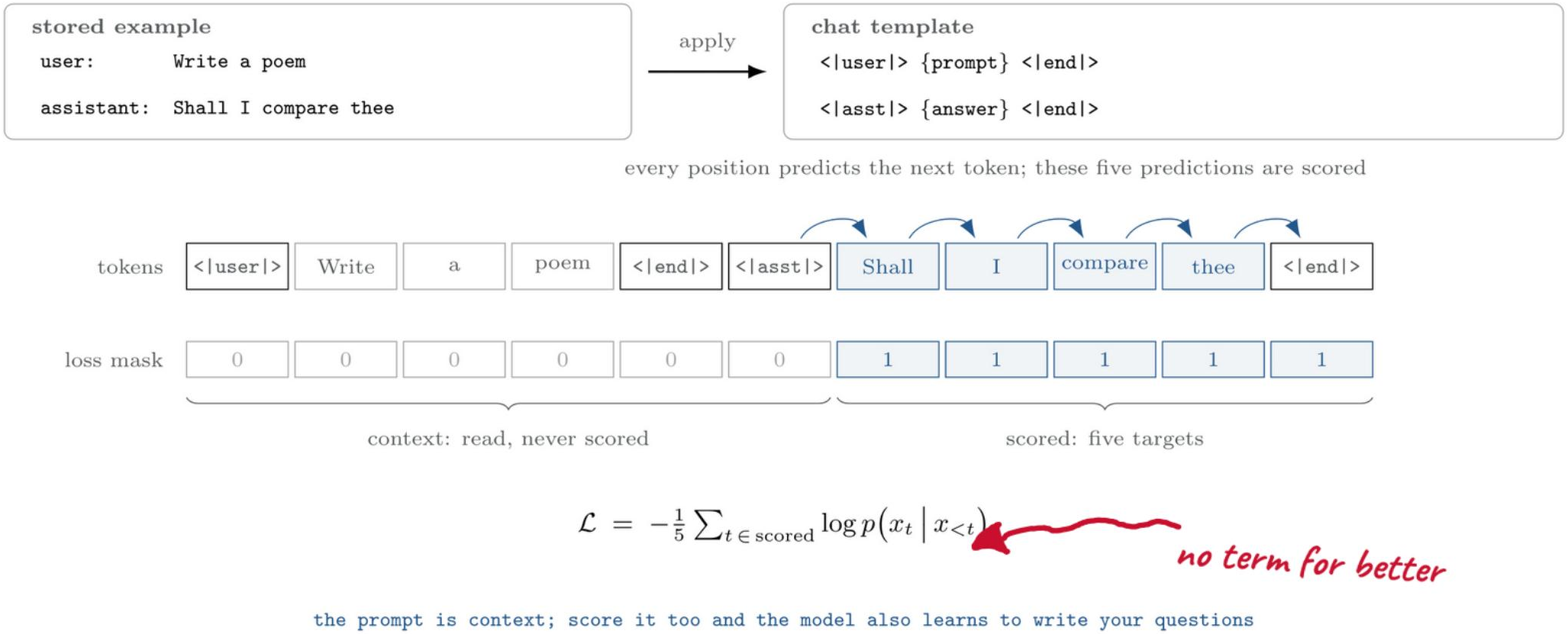
Drop the critic. Skipping the value model for efficiency.

Sample several answers to one prompt and score each against the group average, in place of a value model

A reward you can check. Formal verification: what is correct? Checkable.

A program checks the answer: how a model games a weak check, what a training step needs, how DeepSeek-R1 was trained

The Ceiling

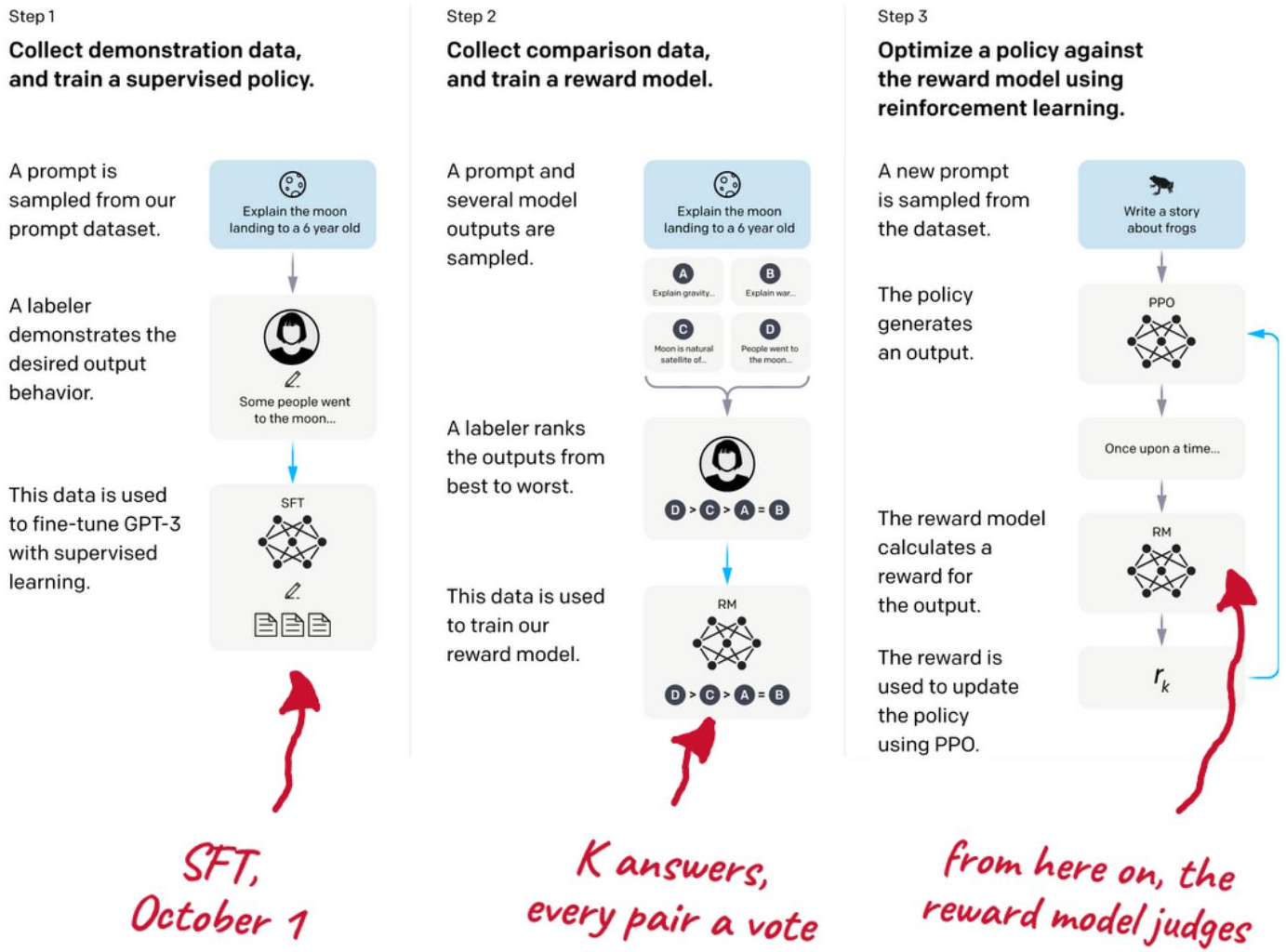


Schematic: one supervised fine-tuning example and its loss mask over the assistant span.

1 PREFERENCE DATA

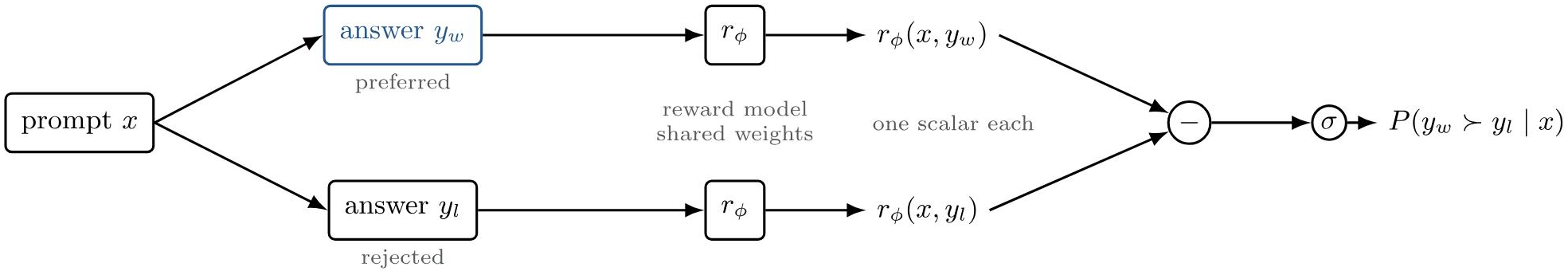
A Reward from Preferences

Three Steps



Ouyang et al., "Training language models to follow instructions with human feedback," 2022, arXiv:2203.02155.

Bradley-Terry

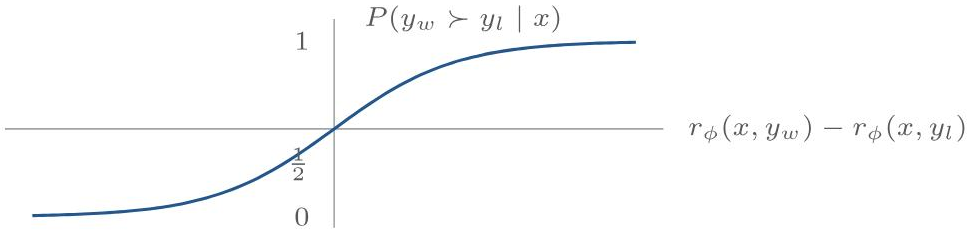


$$P(y_w \succ y_l \mid x) = \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))$$

Bradley and Terry, 1952: a pairwise comparison as a logistic function of the score difference

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

a logistic regression on the difference

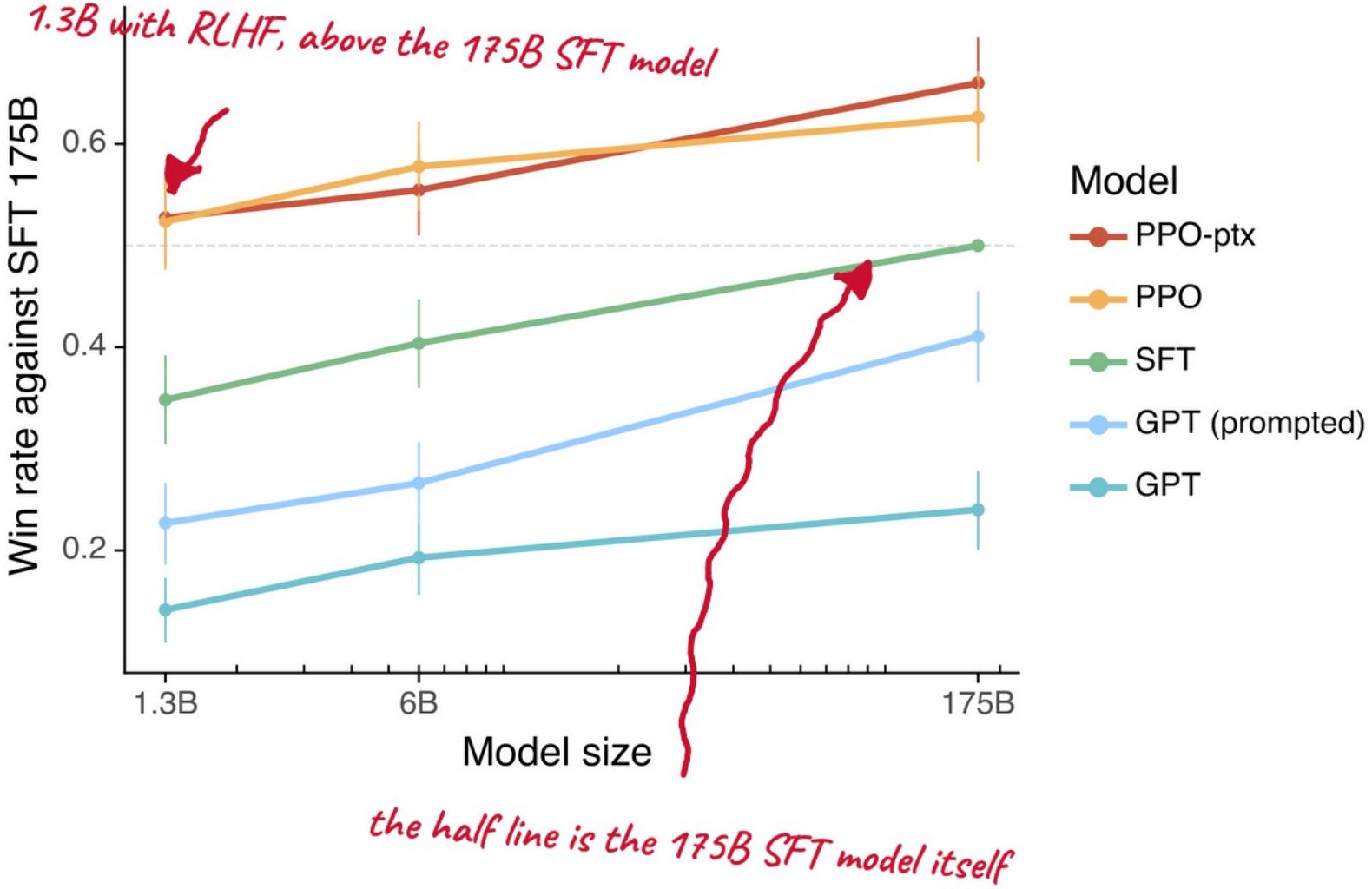


the reward is only defined up to a constant per prompt

$r_\phi(x, y) + c(x)$ gives the same probabilities for every pair

Bradley and Terry, "Rank Analysis of Incomplete Block Designs," 1952. The loss as written in Ouyang et al. 2022 and Rafailov et al. 2023.

A Smaller Model Wins



Ouyang et al., "Training language models to follow instructions with human feedback," 2022, arXiv:2203.02155.

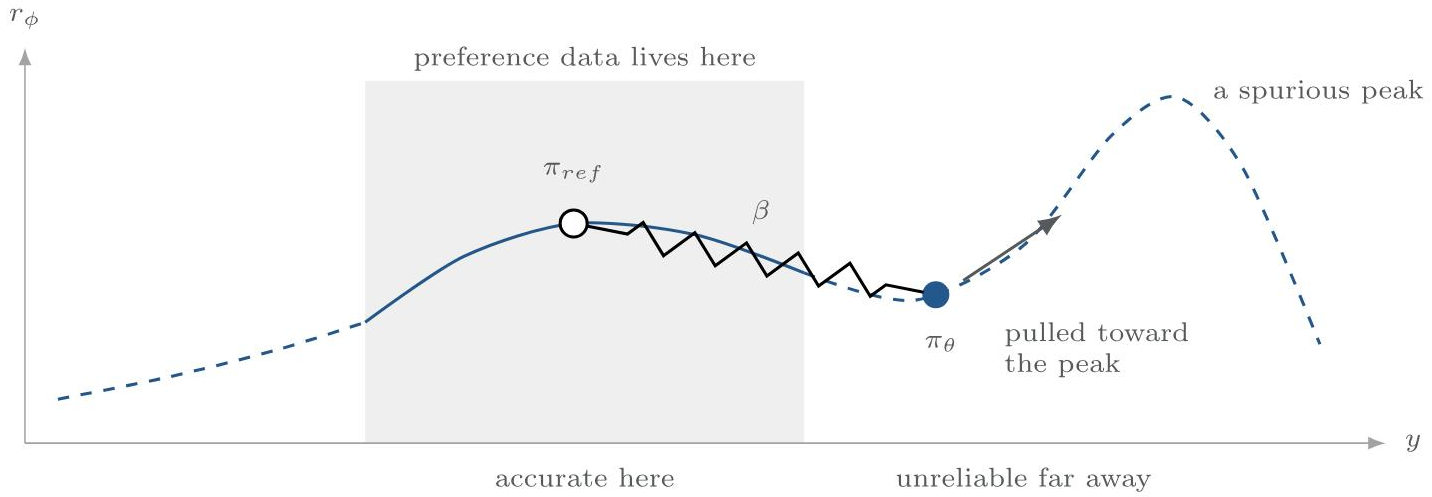
2 RLHF WITH PPO

The Leash

The Objective

$$\max_\theta \; \boxed{\mathrm{E}_{x\sim D,\, y\sim\pi_\theta}[\, r_\phi(x,y)\,]} \; - \beta \; \boxed{\mathrm{D}_{KL}[\, \pi_\theta(y \mid x) \,\|\, \pi_{ref}(y \mid x)\,]}$$

what the reward model likes — stay near the SFT model



$$r_t \;=\; r_\phi(x,y)\,[\,t=T\,] \;-\; \beta\,(\log \pi_\theta(y_t \mid x, y_{<t}) - \log \pi_{ref}(y_t \mid x, y_{<t}))$$

`the KL term applies per token, the reward at the end`

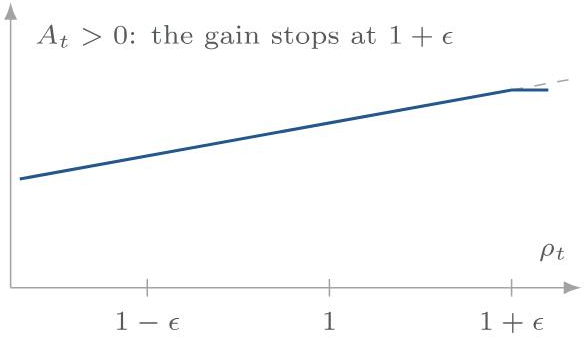
The objective as written in Ouyang et al. 2022, arXiv:2203.02155, and Rafailov et al. 2023, arXiv:2305.18290.

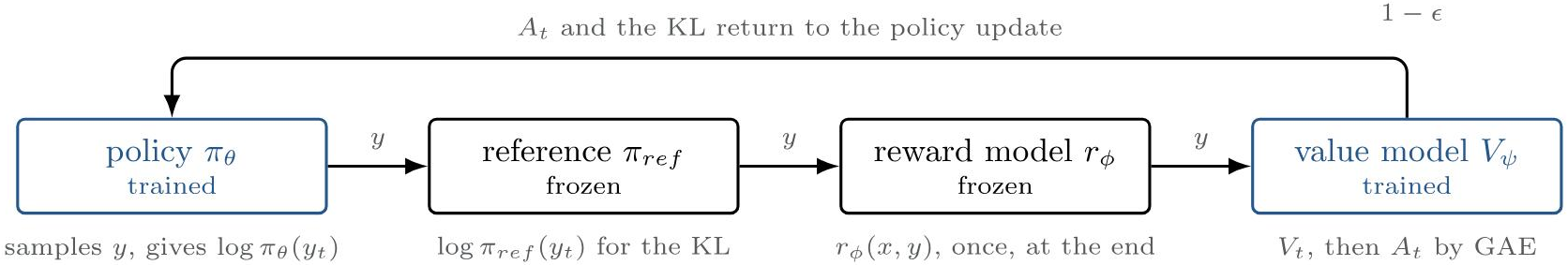
Four Models in Memory

$$\rho_t = \frac{\pi_\theta(y_t \mid x, y_{<t})}{\pi_{old}(y_t \mid x, y_{<t})}$$

$$\mathcal{L}_{PPO} = \mathrm{E}\Big[\min\left(\rho_t A_t,\ \mathrm{clip}(\rho_t,\, 1-\epsilon,\, 1+\epsilon)\, A_t\right)\Big]$$

$\epsilon = 0.2$; π_{old} is the policy that sampled y; A_t comes from GAE with the value model





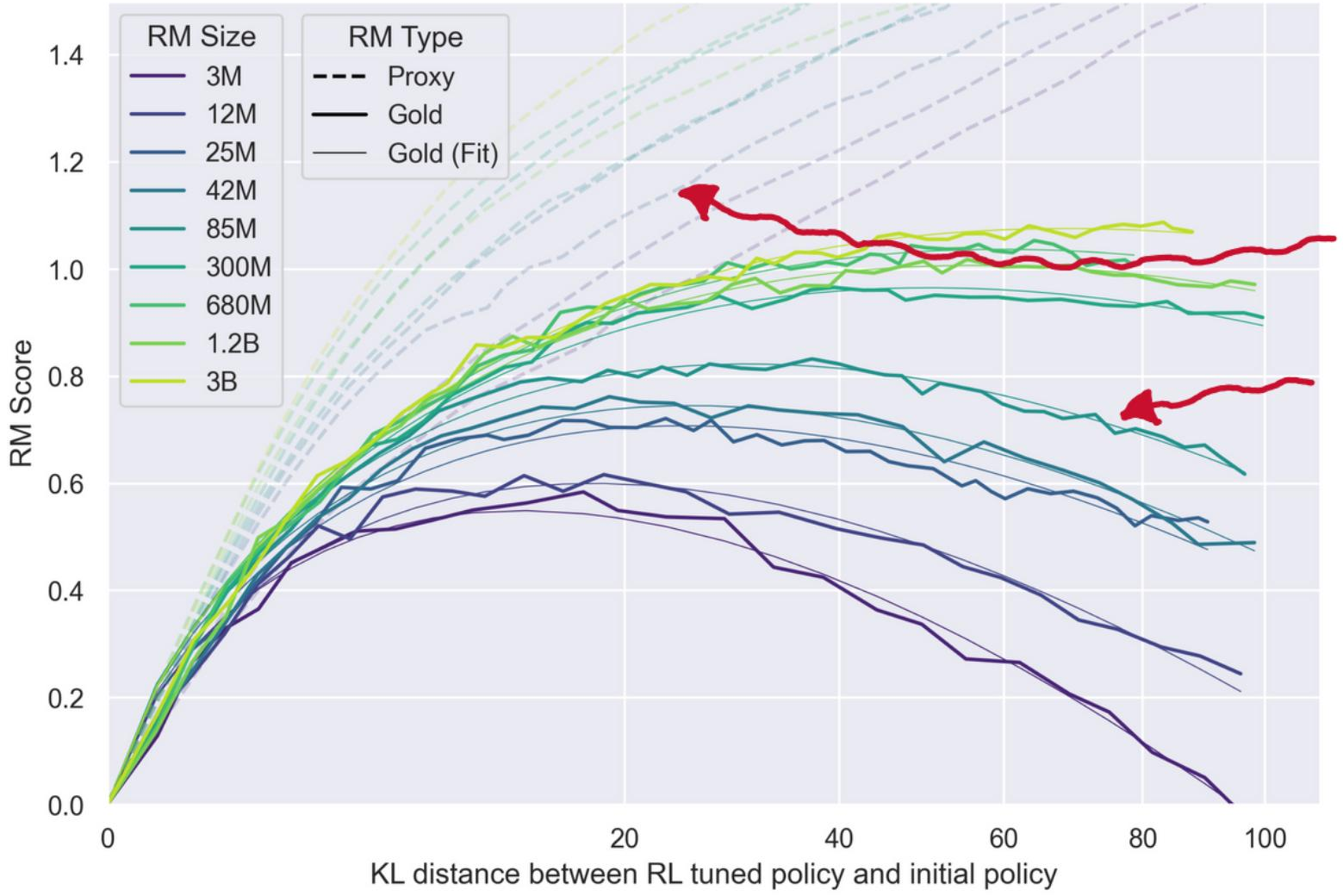
7×10^9 weights: bf16 weights 2 B, bf16 gradient 2 B, fp32 Adam m and v 8 B, so 12 B per trained weight

weights	14 GB	14 GB	14 GB	14 GB	
gradients	14 GB	·	·	14 GB	
Adam m, v	56 GB	·	·	56 GB	
per model	84 GB	14 GB	14 GB	84 GB	$\Sigma \approx 200$ GB

about 200 GB, before activations and the KV cache of sampling

Schulman et al., "Proximal Policy Optimization Algorithms," 2017, arXiv:1707.06347. Bytes: bf16 weights, bf16 gradients and fp32 Adam moments, the arithmetic from the fine-tuning lecture.

Overoptimization



dashed: the score from the reward model we train against (the proxy)

solid: a much bigger reward model standing in for people; it peaks, then falls

a bigger proxy peaks later

Gao, Schulman and Hilton, "Scaling Laws for Reward Model Overoptimization," 2022, arXiv:2210.10760.

3 DIRECT PREFERENCE OPTIMIZATION

No Reward Model

From Reward to Policy

$$\max_\theta \; \mathrm{E}_{x\sim D,\, y\sim\pi_\theta}\big[\, r(x,y)\,\big] \;-\; \beta\, \mathrm{D}_{KL}\big[\, \pi_\theta(y \mid x) \,\|\, \pi_{ref}(y \mid x)\,\big]$$

$$\pi^*(y \mid x) \;=\; \frac{1}{Z(x)}\, \pi_{ref}(y \mid x)\, \exp\!\Big(\tfrac{1}{\beta}\, r(x,y)\Big)$$

Z(x) sums over every y: intractable

$$r(x,y) \;=\; \beta \log \frac{\pi^*(y \mid x)}{\pi_{ref}(y \mid x)} \;+\; \beta \log Z(x)$$

the policy defines a reward

$$P(y_w \succ y_l \mid x) \;=\; \sigma\!\Big(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{ref}(y_w \mid x)} \;-\; \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{ref}(y_l \mid x)}\Big)$$

Z(x) cancels: the same x in both rewards

$$\mathcal{L}_{DPO}(\theta) \;=\; -\,\mathrm{E}_{(x,y_w,y_l)\sim D}\Big[\log \sigma(\hat{r}_\theta(x,y_w) - \hat{r}_\theta(x,y_l))\Big], \qquad \hat{r}_\theta(x,y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{ref}(y \mid x)}$$

$$\nabla_\theta \mathcal{L}_{DPO} \;=\; -\beta\, \mathrm{E}\Big[\boxed{\sigma(\hat{r}_\theta(x,y_l) - \hat{r}_\theta(x,y_w))}\Big(\, \nabla_\theta \log \pi_\theta(y_w \mid x) - \nabla_\theta \log \pi_\theta(y_l \mid x)\,\Big)\Big]$$

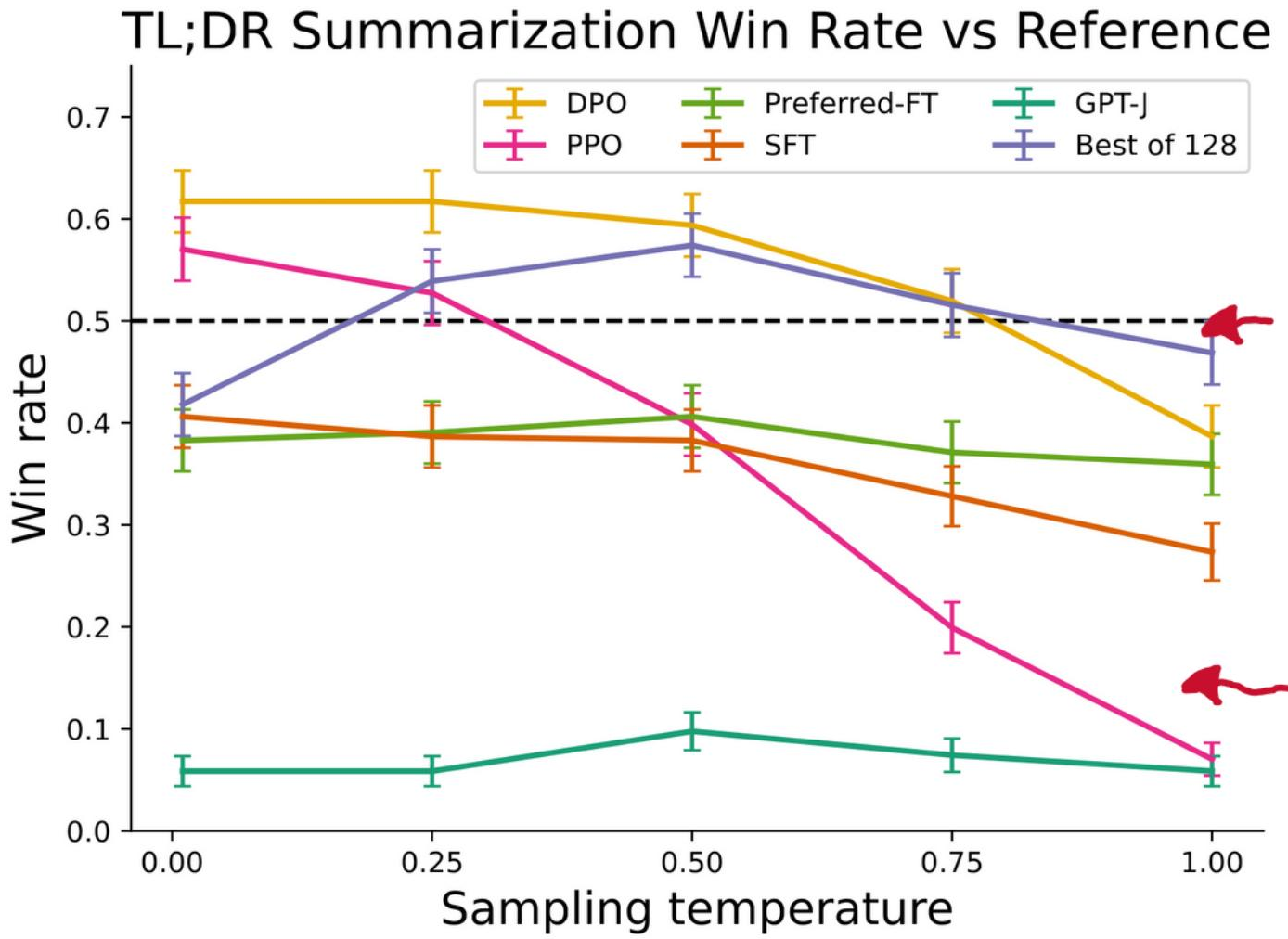
larger when the implicit reward gets the pair wrong

raise the log-prob of y_w

lower the log-prob of y_l

Rafailov et al., "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," 2023, arXiv:2305.18290.

DPO on Summaries



GPT-4 judges against the human summary

best of 128: draw 128 summaries from the SFT model, keep the one the reward model scores highest

DPO, the yellow curve, is above PPO, the pink one, at every temperature

PPO falls apart at temperature 1

Rafailov et al., "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," 2023, arXiv:2305.18290.

The Family

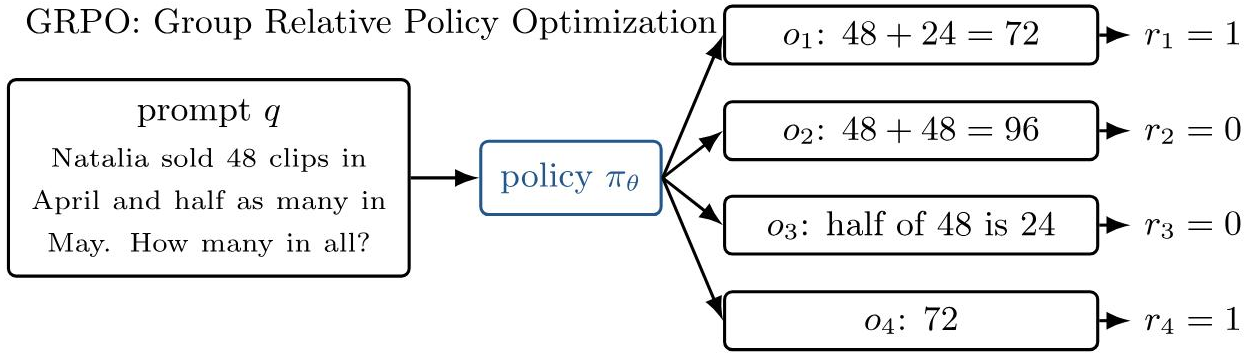
method	data needed	reference model	what it changes
DPO	pairs (y_w, y_l)	yes	the baseline: overfits to the pairs
IPO	pairs (y_w, y_l)	yes	a bounded loss, less overfitting
KTO	unpaired good or bad labels	yes	learns from thumbs up or down alone
ORPO	pairs (y_w, y_l)	no	one stage with SFT
SimPO	pairs (y_w, y_l)	no	length-normalized reward

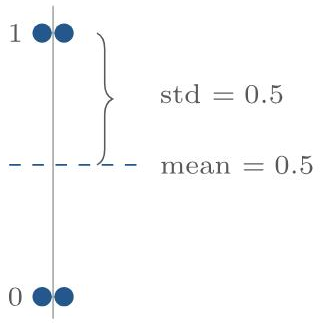
Rafailov et al. 2023, arXiv:2305.18290; Azar et al. 2023, arXiv:2310.12036; Ethayarajh et al. 2024, arXiv:2402.01306; Hong et al. 2024, arXiv:2403.07691; Meng et al. 2024, arXiv:2405.14734.

4 GRPO

Drop the Critic

Group Advantage

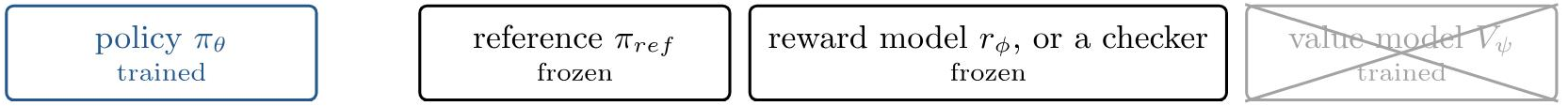




$$\hat{A}_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G)}$$

$\hat{A} = (+1,\ -1,\ -1,\ +1)$

the same value on every token of o_i



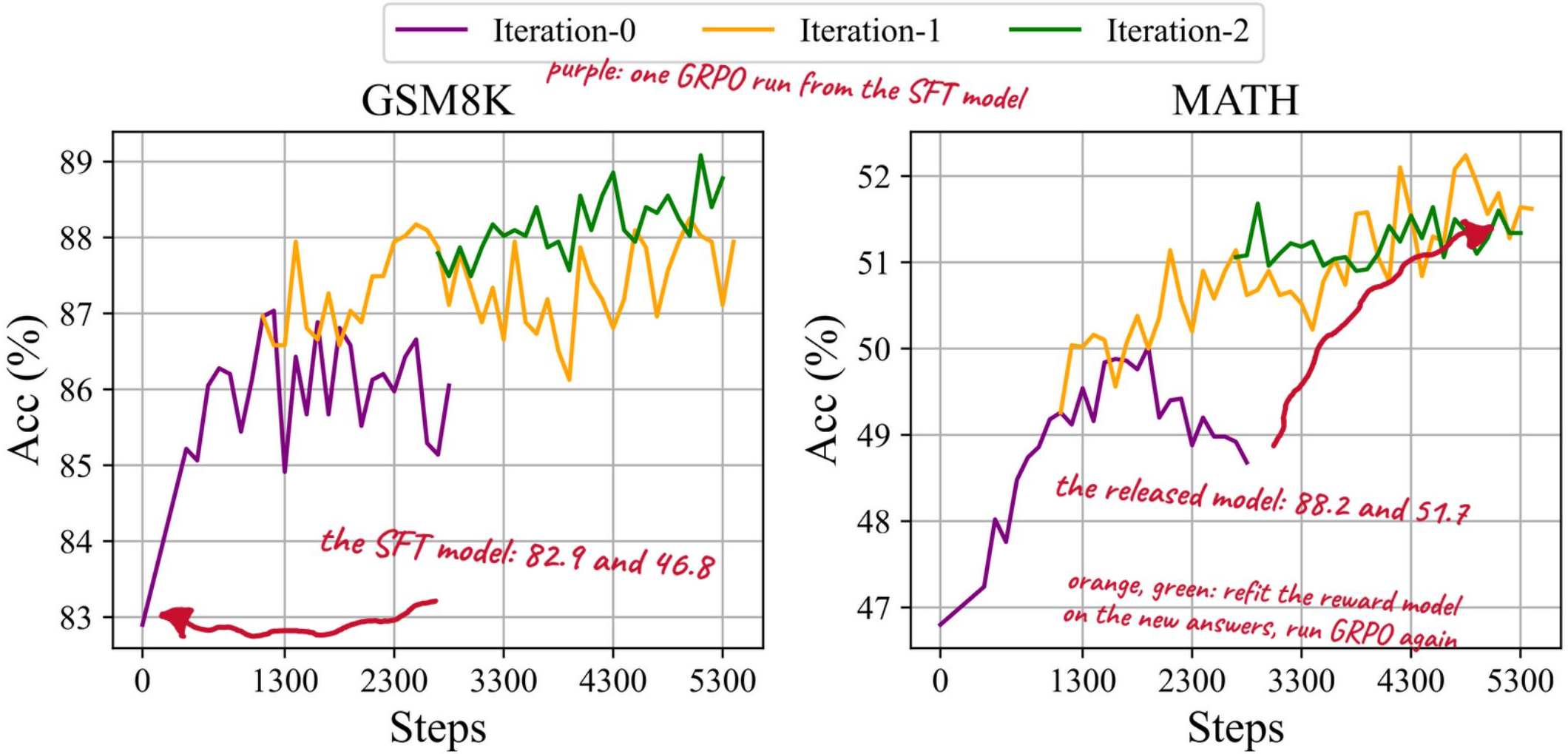
$$\mathcal{J}_{GRPO}(\theta) = \mathrm{E}_{q\sim P(Q),\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(O|q)} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min\left(\rho_{i,t}\hat{A}_{i,t},\ \text{clip}(\rho_{i,t}, 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\right) - \beta\, \mathrm{D}_{KL}\left[\pi_\theta \,\|\, \pi_{ref}\right] \right) \right]$$

$$\rho_{i,t} = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} \mid q, o_{i,<t})}, \qquad \mathrm{D}_{KL}\left[\pi_\theta \,\|\, \pi_{ref}\right] = \frac{\pi_{ref}(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta(o_{i,t} \mid q, o_{i,<t})} - \log \frac{\pi_{ref}(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta(o_{i,t} \mid q, o_{i,<t})} - 1$$

Shao et al. 2024; the KL is in the loss, outside the advantage

Shao et al., "DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models," 2024, arXiv:2402.03300.

DeepSeekMath

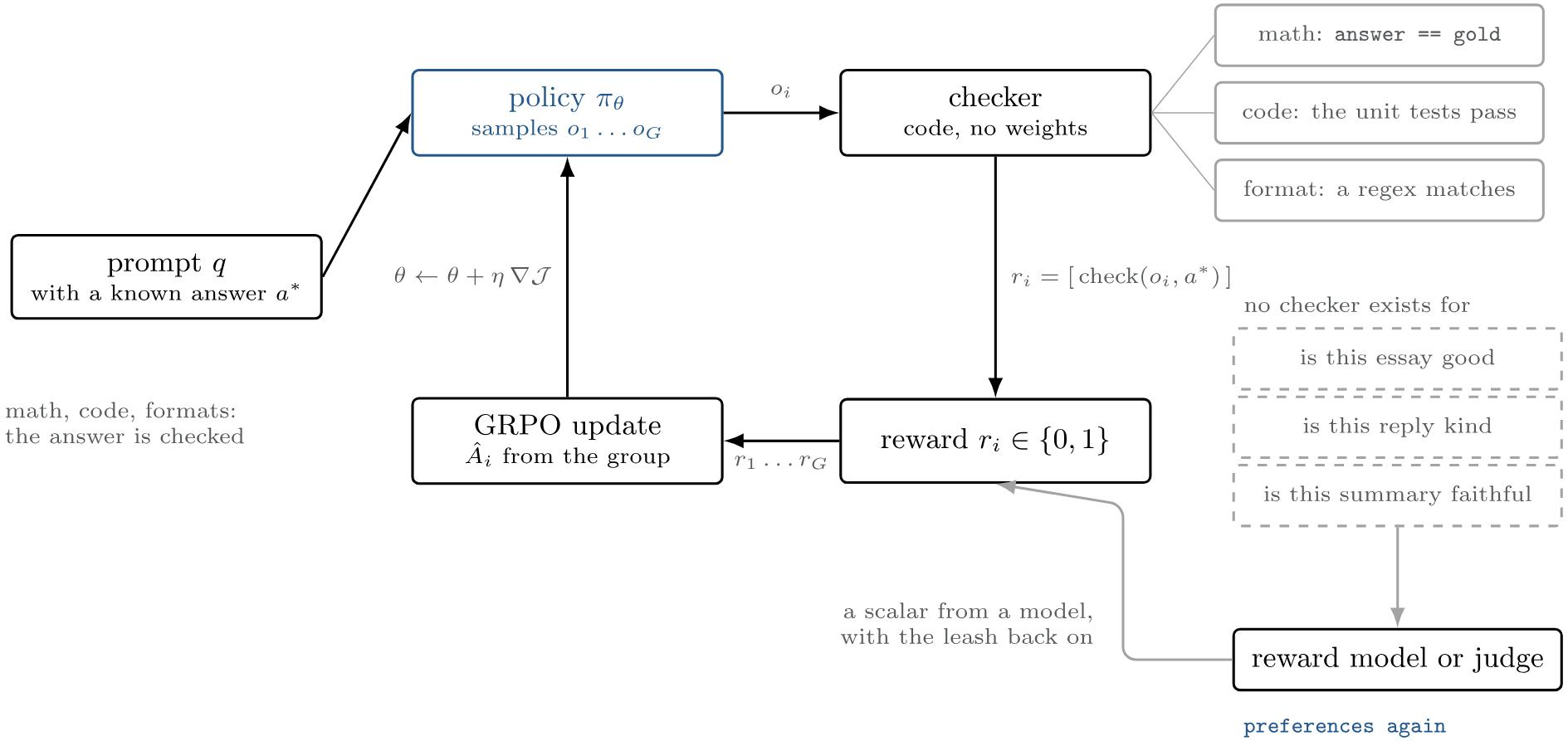


Shao et al., "DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models," 2024, arXiv:2402.03300.

5 VERIFIABLE REWARDS

A Reward You Can Check

Where a Checker Exists



Lambert et al., "Tulu 3: Pushing Frontiers in Open Language Model Post-Training," 2024, arXiv:2411.15124; DeepSeek-AI, "DeepSeek-R1," 2025, arXiv:2501.12948.

Break the Verifier

naive: the gold number appears anywhere

```
def naive(completion, gold):
    return gold in completion.replace(",", "")
```

72 gold: the dataset's own answer to the question, a number such as 72; reward 1 when it appears anywhere

the commas go because the model writes 1,000 where the key says 1000

strict: the last line is the answer

```
def strict(completion, gold):
    last = completion.strip().splitlines()[-1]
    return last.strip() == f"#### {gold}"
```

reward 1 when the final line is exactly the GSM8K answer line, and nothing follows it

ways to get the point without solving anything:

1. list every number from 1 to 1000

2. copy the question; its numbers are in it

3. write three candidate answers, commit to none

one number, one place: none of the three score

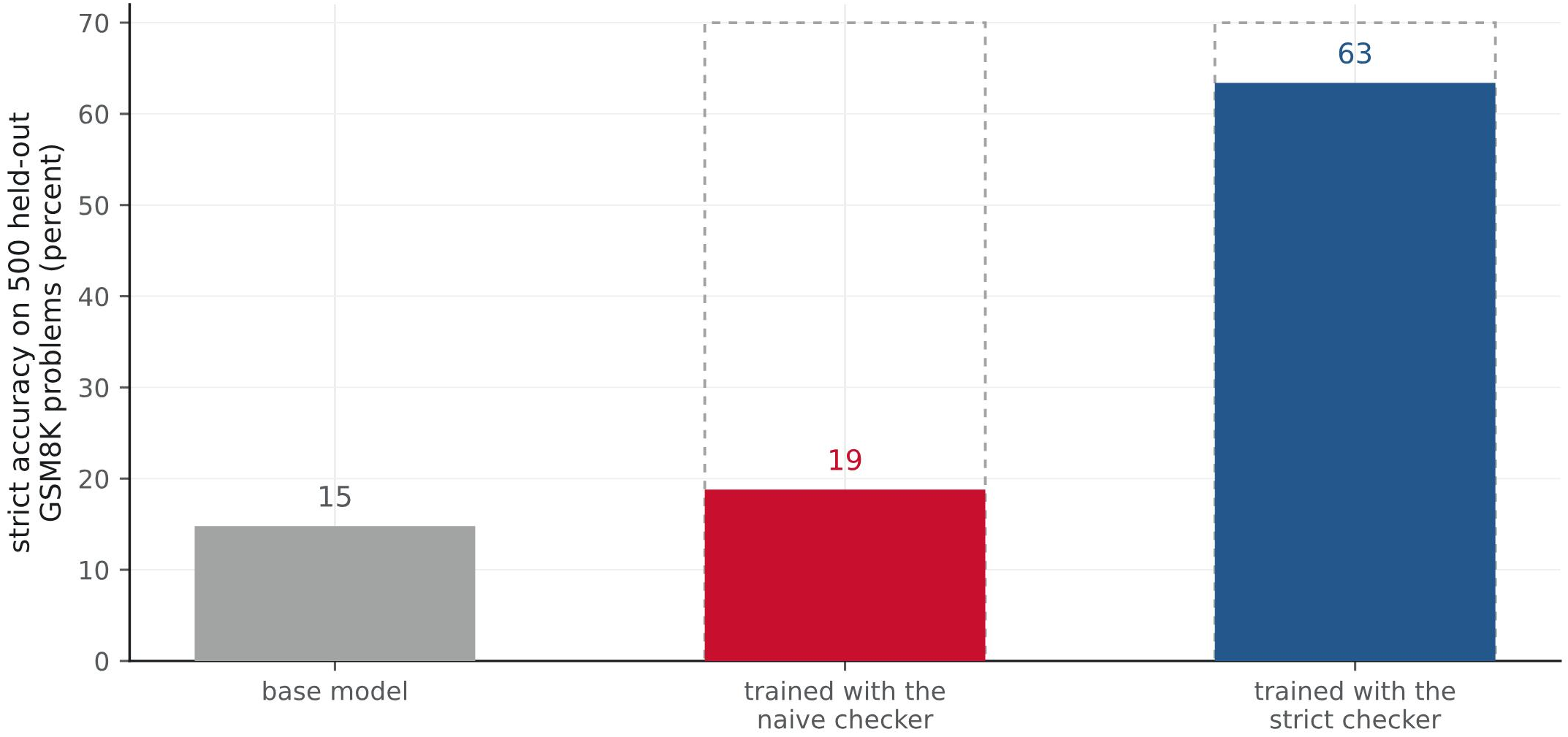
My checkers for the run on the next slides, GSM8K answers.

My Run



My runs on Qwen3-0.6B-Base on my RTX 4090, GSM8K train prompts, GRPO; beta is the weight on the KL term in the GRPO objective, 0.04 as in DeepSeekMath.

My Run



My runs on Qwen3-0.6B-Base on my RTX 4090, the first 500 GSM8K test problems, greedy decoding, scored by the strict checker.

What the Policy Found

step 0: 153 tokens, gold 9

12 numbers in the text

```
Q: Marcus, Humphrey, and Darrel are bird watching. Marcus sees 7
birds, Humphrey sees 11 birds, and Darrel sees 9 birds. How many
birds does each of them see on average?

To find the average number of birds seen by each person,
follow these steps:

1. **Calculate the total number of birds seen by all
three:**
   \[
   7 \text{ (Marcus)} + 11 \text{ (Humphrey)} + 9 \text{
(Darrel)} = 27 \text{ birds}
   \]

2. **Divide the total number of birds by the number of
people (3):**
   \[
   \frac{27 \text{ birds}}{3} = 9 \text{ birds}
   \]
...
```

step 149: 288 tokens, gold 440

26 numbers in the text

```
Q: A 1000 car parking lot is divided into 3 sections. There are
320 spaces in section 1, and 200 more in section 2 than in
...

To determine the number of spaces in section 2 of the
parking lot, we can follow these steps:

1. Let the number of spaces in section 3 be \( x \).
2. Then, the number of spaces in section 2 will be \( x +
200 \).
3. The total number of spaces in sections 1, 2, and 3
together is 1000. Therefore, we can set up the equation:
\[ 320 + (x + 200) + x = 1000 \]

First, combine like terms:
\[ 320 + 200 + 2x = 1000 \]
\[ 520 + 2x = 1000 \]

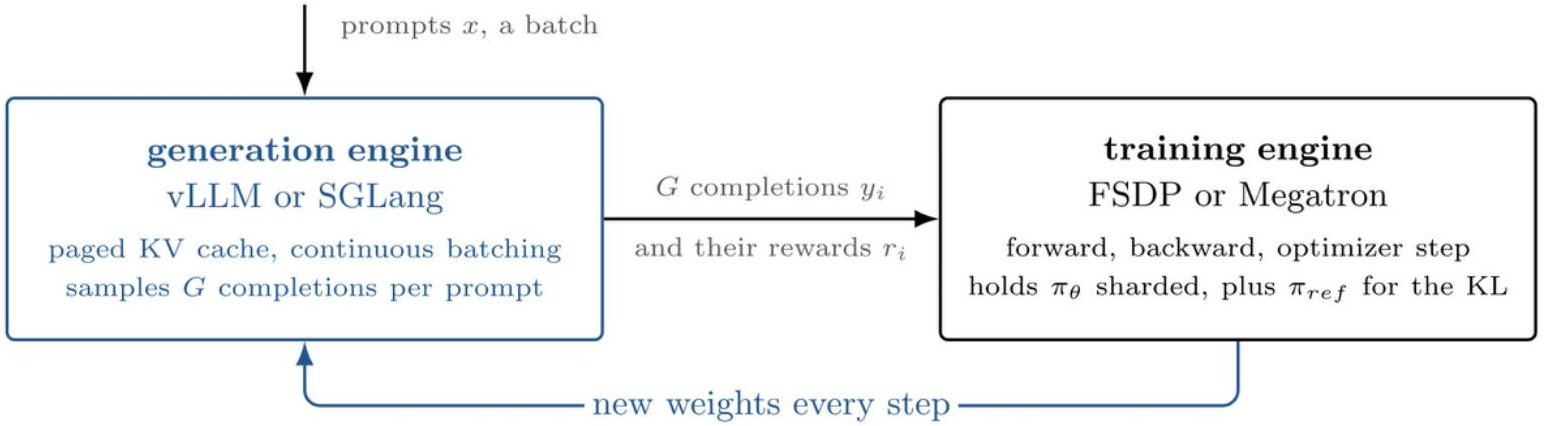
Next, isolate the variable \( x \) by subtracting 520 from
...
```

My run on Qwen3-0.6B-Base on my RTX 4090, the naive checker, sampled completions at the first and last step.

6 FRAMEWORKS

What a Step Needs

Two Engines



my run: 7 of 9.5 s a step is generation, with plain HF generate

colocated

same GPUs, time-shared

disaggregated

separate GPU pools, NCCL broadcast

what exists before step one

a serving engine	a trainer	a reward service	an orchestrator
reloads weights	shards policy and reference	checkers or a judge	Ray

`one GPU: none of this, and the wait is generation`

Sheng et al., "HybridFlow: A Flexible and Efficient RLHF Framework," 2024, arXiv:2409.19256. Timing from my run on my RTX 4090.

Frameworks

framework	rollout engine	rollouts and training	training backend	what it is for
verl	vLLM, SGLang, HF	same or separate GPUs	FSDP, Megatron	research-scale GRPO at ByteDance
TRL	HF generate, or vLLM	same or separate GPUs	Accelerate, DeepSpeed	the Hugging Face default
OpenRLHF	vLLM	same or separate GPUs	DeepSpeed ZeRO-3	Ray plus vLLM, async option
NeMo-RL	vLLM, Megatron	not stated	FSDP2, Megatron	Nemotron post-training
SkyRL	vLLM, SGLang	same or separate GPUs	FSDP, Megatron	Berkeley; in-flight weight updates
slime	SGLang	same or separate GPUs	Megatron	behind GLM-4.5 to GLM-5.3
AReaL	SGLang, vLLM	separate GPUs	Megatron, FSDP	asynchronous rollouts, Ant Group
Unsloth	vLLM, shared memory	the same GPU	LoRA, one process	one GPU, 5 to 15 GB
torchtitan	none; TitanRL adds vLLM	no rollouts	FSDP2, TP, PP, CP	pretraining and SFT; no rollout engine

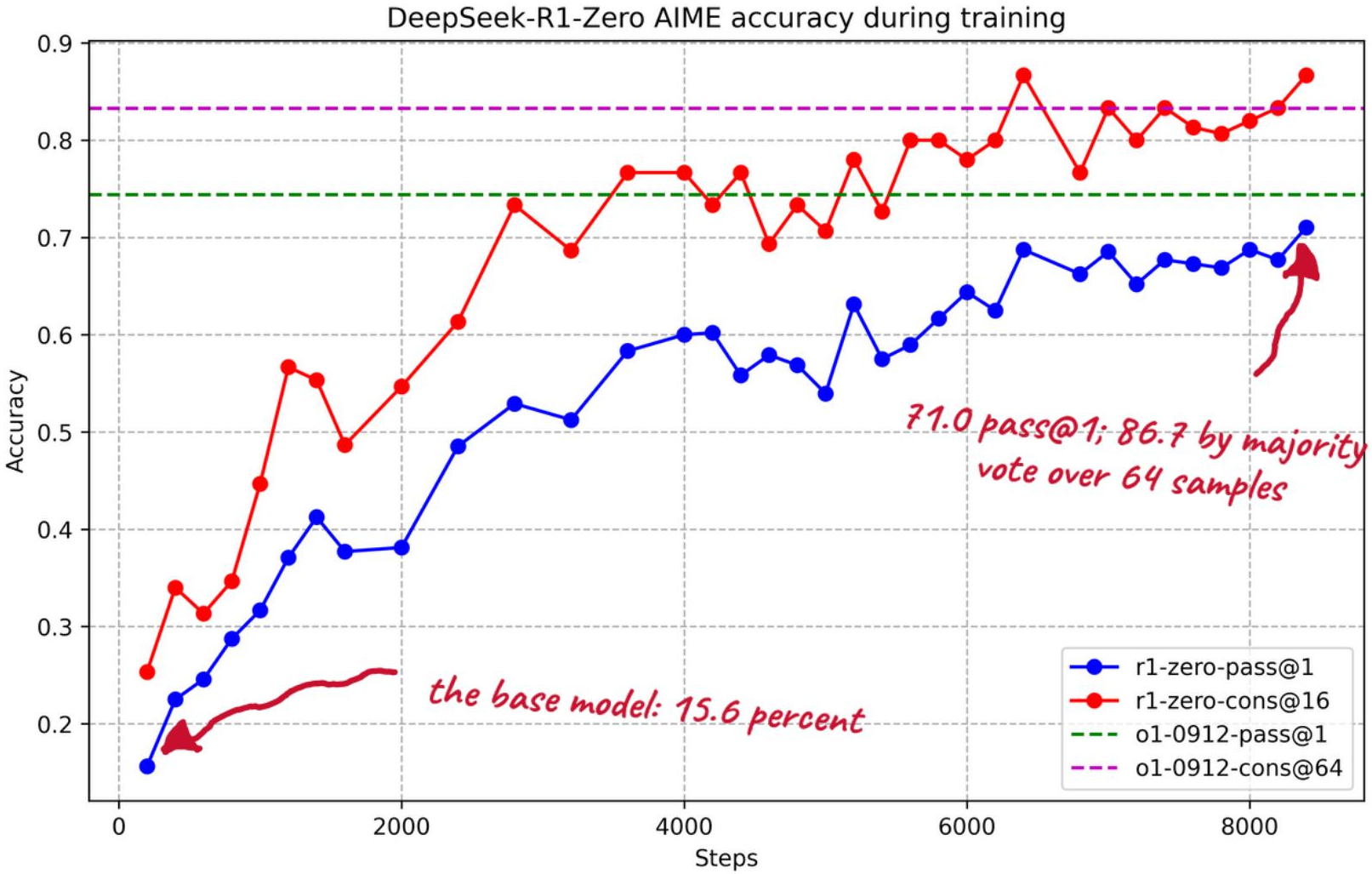
what I've used: torchtitan for SFT, verl for RL

READMEs and docs read October 2026: volcengine/verl, huggingface/trl, OpenRLHF/OpenRLHF, NVIDIA-NeMo/RL, NovaSky-AI/SkyRL, THUDM/slime, inclusionAI/AReaL, unslothai/unsloth, pytorch/torchtitan. Sheng et al., HybridFlow, 2024, arXiv:2409.19256.

7 DEEPSEEK-R1

The R1 Recipe

R1-Zero



DeepSeek-AI, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," 2025, arXiv:2501.12948.

Longer Answers

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\cdots$

$$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

$\cdots$

Wait, wait. Wait. That's an aha moment I can flag here.

the model's own words, mid-training

Let's reevaluate this step-by-step to identify if the correct sum can be $\cdots$

We started with the equation:

$$\sqrt{a - \sqrt{a + x}} = x$$

First, let's square both sides:

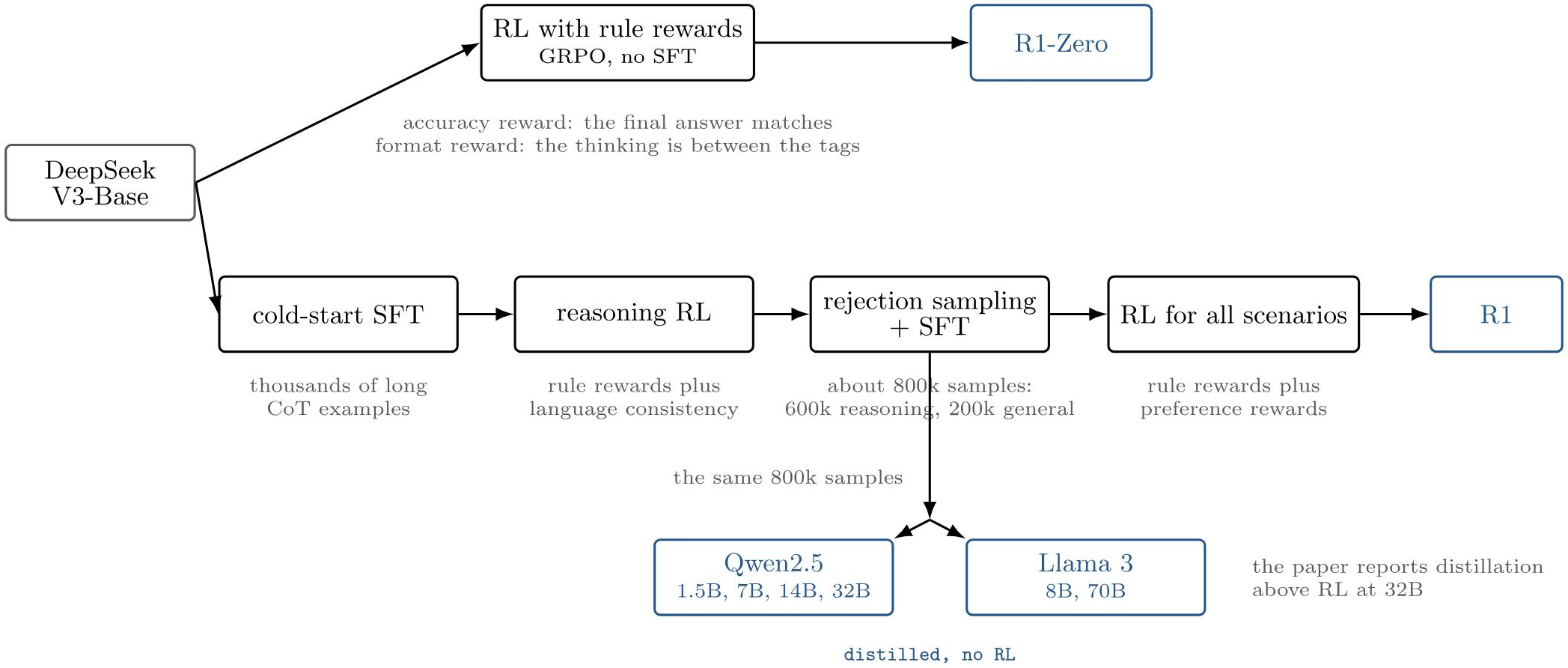
$$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$$

Next, I could square both sides again, treating the equation: $\cdots$

$\cdots$

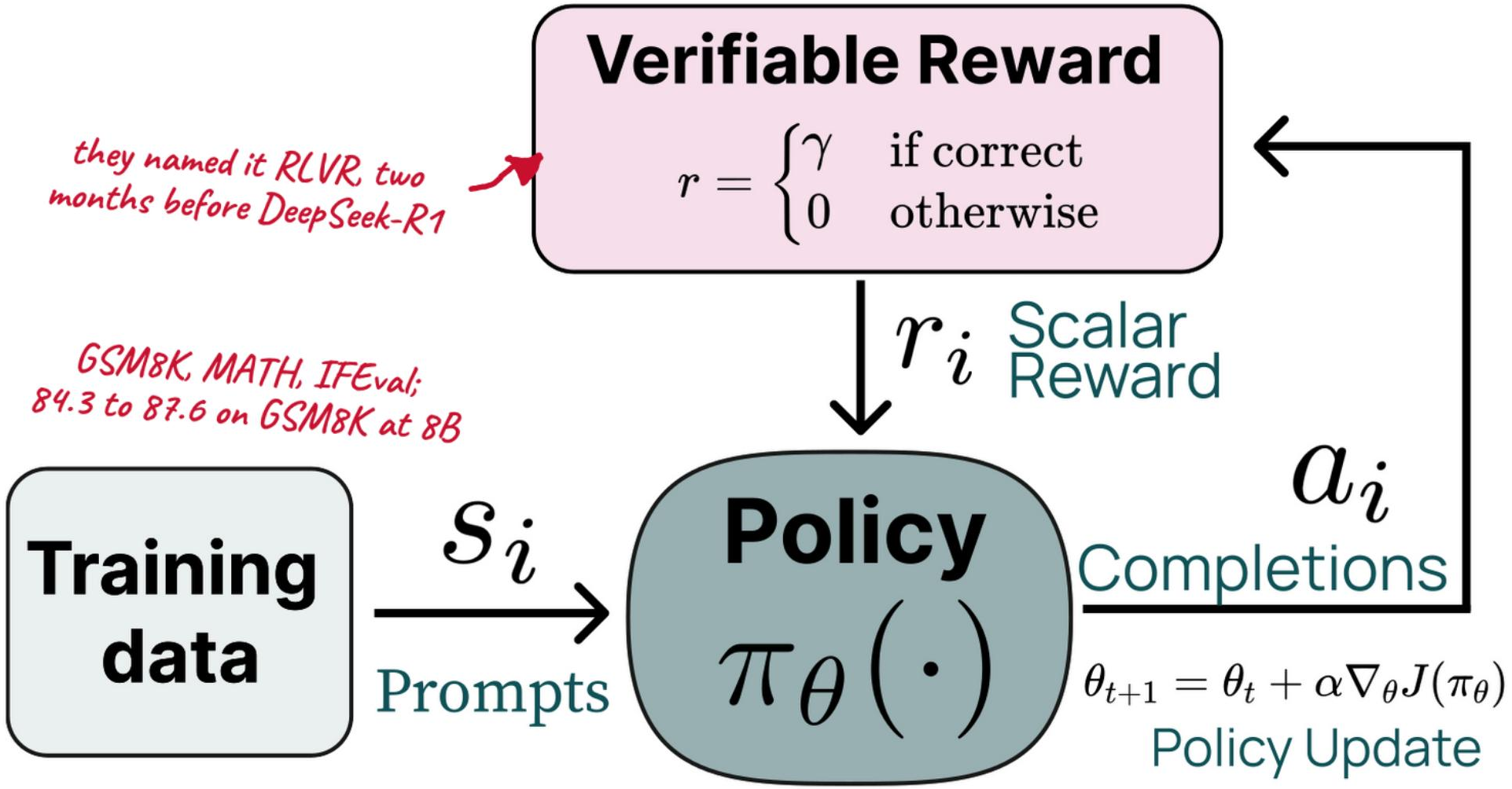
DeepSeek-AI, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," 2025, arXiv:2501.12948.

Four Stages



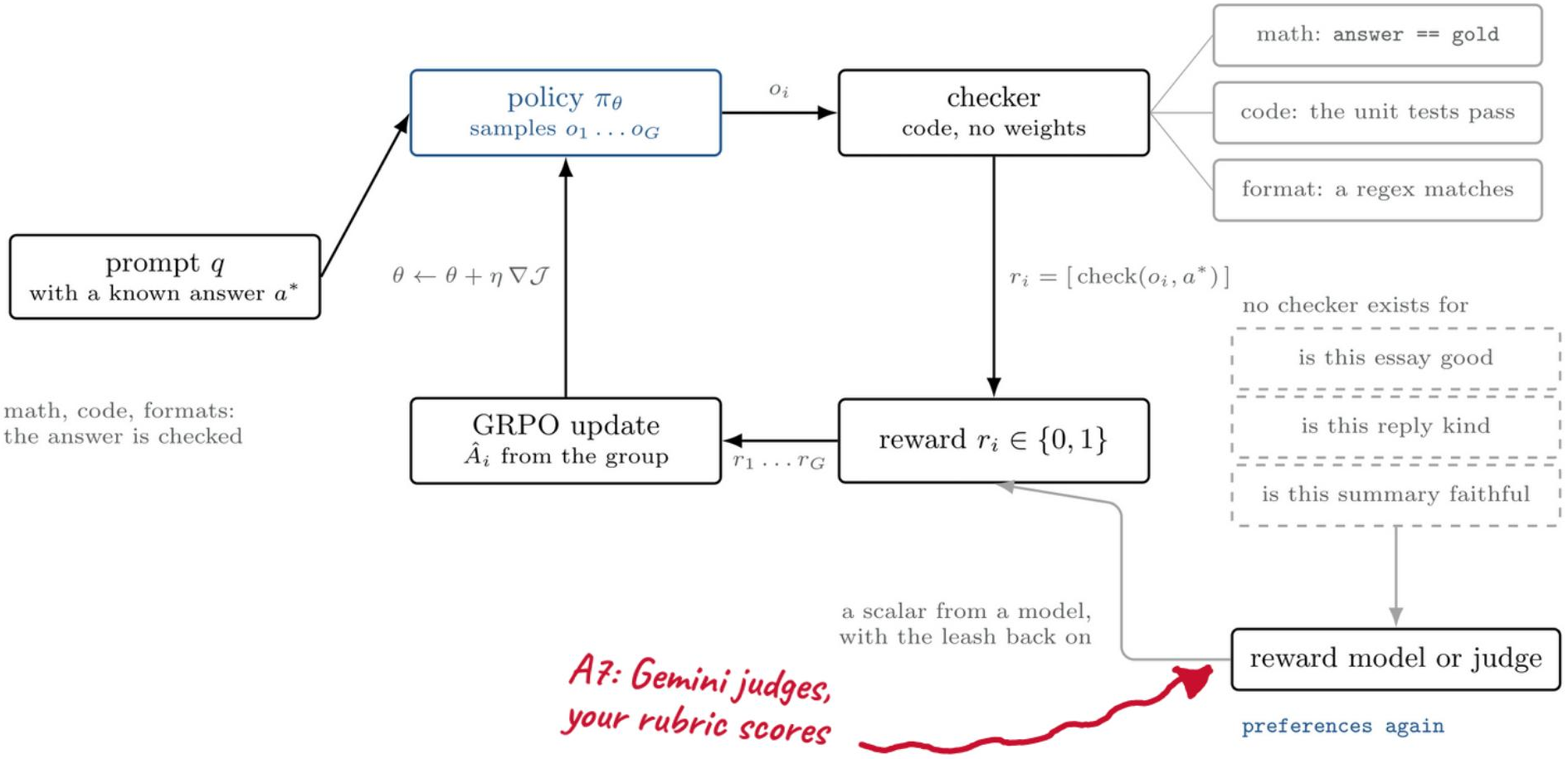
DeepSeek-AI, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," 2025, arXiv:2501.12948.

Open Recipe



Lambert et al., "Tulu 3: Pushing Frontiers in Open Language Model Post-Training," 2024, arXiv:2411.15124.

Assignment 7



Bai et al., "Constitutional AI: Harmlessness from AI Feedback," 2022, arXiv:2212.08073.

Reading

Required

Rafailov et al., Direct Preference Optimization · arXiv:2305.18290

DeepSeek-AI, DeepSeek-R1 · arXiv:2501.12948

Optional

Ouyang et al., InstructGPT · arXiv:2203.02155 Shao et al., DeepSeekMath · arXiv:2402.03300

Ethayarajh et al., KTO · arXiv:2402.01306

Bai et al., Constitutional AI · arXiv:2212.08073 Lambert et al., Tulu 3 · arXiv:2411.15124

Assignment 6 is due Tuesday with its quiz in class. **Assignment 7**, RLAIF with a judge, is out Tuesday.