

UNIVERSITY OF CAMBRIDGE

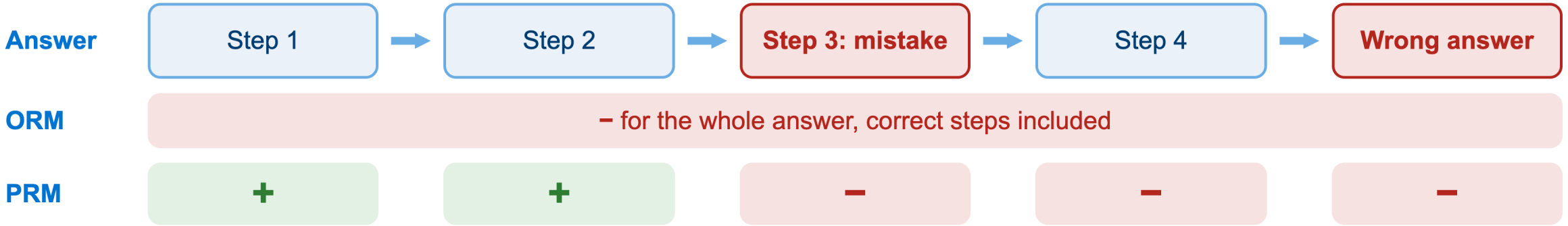
GRPO is Secretly a Process Reward Model

Sullivan & Koller (ICML 2026)

Dario Vajda

L101: Machine Learning for Language Processing
Michaelmas 2026

Background: outcome vs process rewards



PRM strengths

- Finer credit assignment
- Better multi-step reasoning than outcome rewards (Lightman et al., 2024)

PRM weaknesses

- Learned PRMs need costly step-level labels (Zhang et al., 2025)
- Prone to reward hacking (Cui et al., 2025)
- Monte Carlo PRMs need no labels, but many extra rollouts (Kazemnejad et al., 2025; Hou et al., 2025)

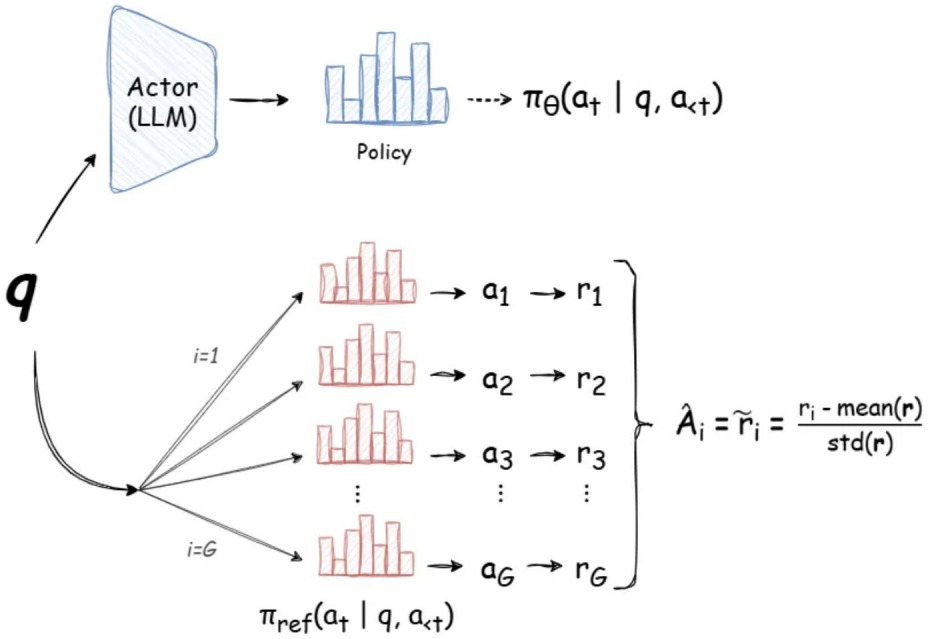
Background: GRPO

Group Relative Policy Optimization (Shao et al., 2024)

- DeepSeek-R1 (DeepSeek-AI, 2025)
- PPO (Schulman et al., 2017) without the critic (value model): much less memory

$$a_i = \frac{r_i - \text{mean}(r)}{\text{std}(r)}$$

- **Every token of answer i gets the same advantage**
- **RLVR** (Reinforcement Learning with Verifiable Rewards)

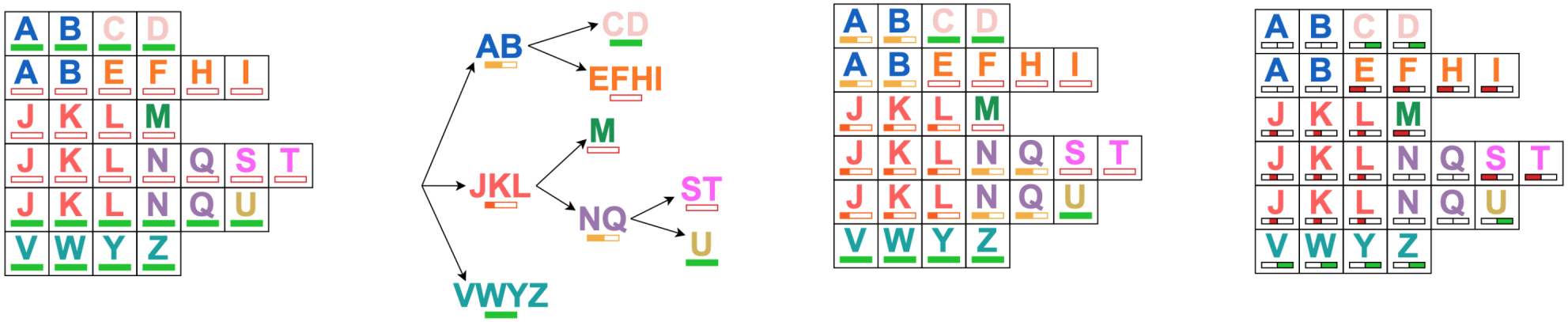


Source: "Group Relative Policy Optimization (GRPO) Illustrated Breakdown", Towards AI

The paper in one sentence

GRPO with outcome-only rewards already acts as a process reward model (PRM) whenever rollouts share a prefix; the authors show that fixing how it weights those shared steps (λ-GRPO) trains faster and scores higher.

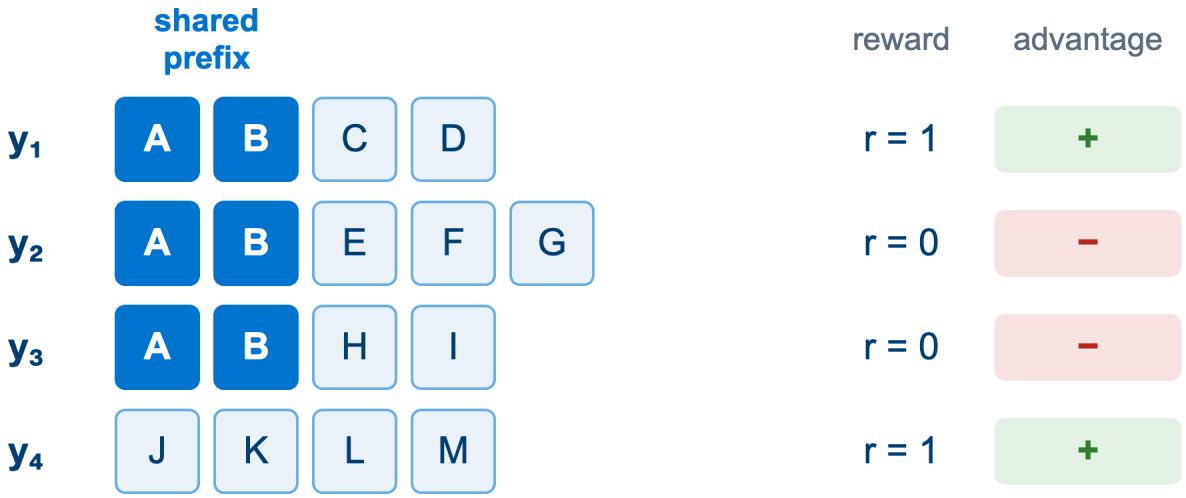
- step-level credit comes for free (under some assumptions)
- the authors propose λ-GRPO, a one-line change, which improves exploration and benchmark performance



Sullivan & Koller (2026), Fig. 1

Motivation: what if rollouts share a prefix?

One GRPO group: four rollouts for the same prompt



Tokens **A B** get **+** from y_1 but **−** from y_2 and y_3. Same tokens, same context, so the updates **add up**.

- Rollouts often **start the same way**
- GRPO copies each rollout's advantage onto **all** of its tokens
- A shared prefix is influenced by many different rollouts

What reward does a shared prefix actually receive?

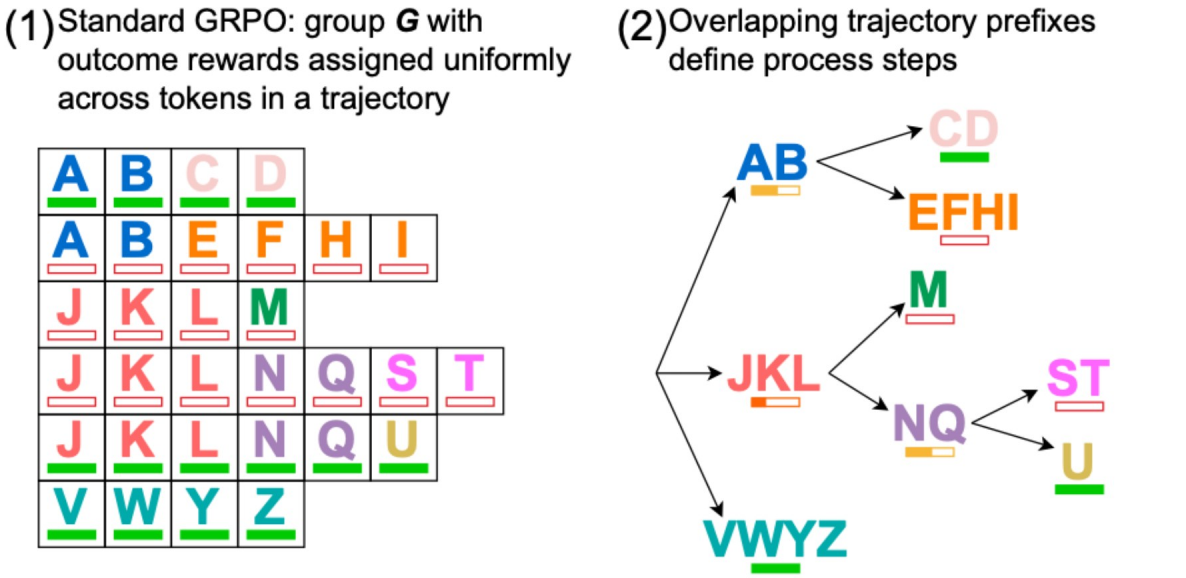
Theorem 1: GRPO is a PRM

Assumptions:
1. *the loss averages over all tokens in the group (Yu et al., 2025);*
2. *each batch is used for one update only.*

Theorem 1. *For any query x, policy π_θ, and group $\mathbb{G} \sim \pi_\theta(\cdot \mid x)$ with outcome-level rewards $\{r^{(i)}\}_{y^{(i)} \in \mathbb{G}}$:*

$$\mathcal{L}_{\mathrm{GRPO}}(\mathbb{G}) = \mathcal{L}_{\mathrm{PRM}}(\mathbb{G})$$

Tokens shared by several rollouts get the average reward of those rollouts.
⇒ GRPO gives per-step rewards, without a separate reward model.



Sullivan & Koller (2026), Fig. 1

Does this happen in practice?

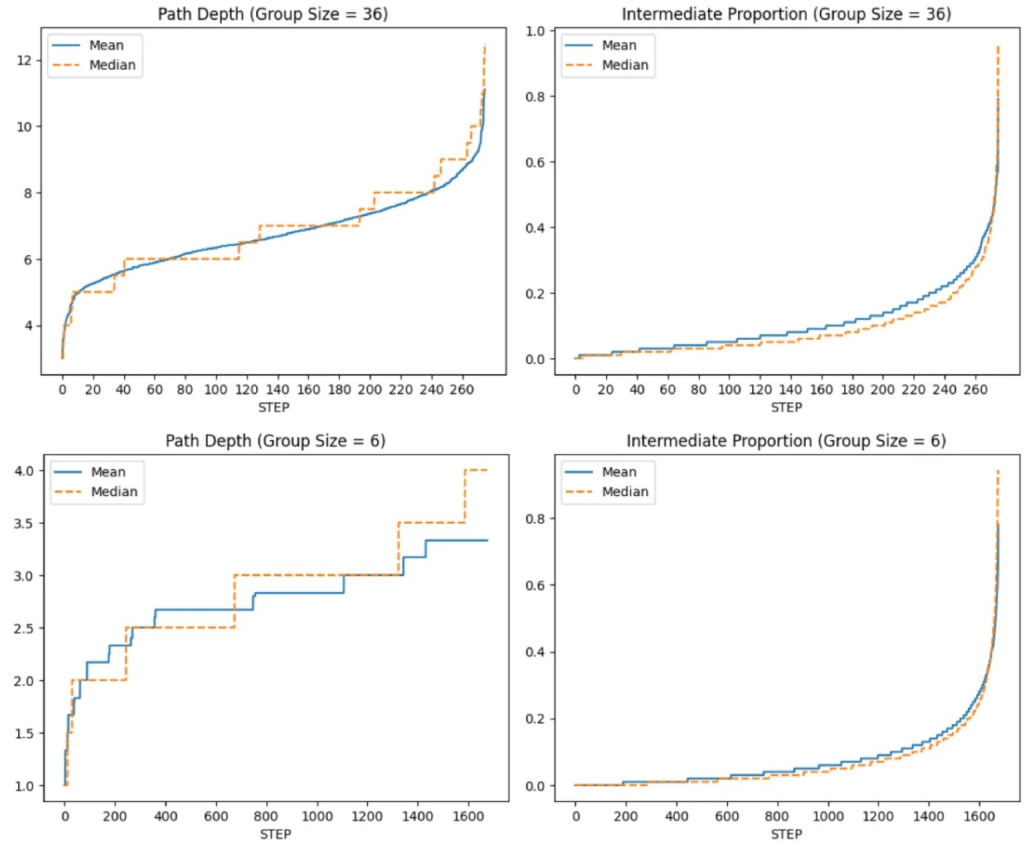
Path depth: how many shared prefixes a rollout belongs to

Intermediate proportion: fraction of a rollout's tokens inside a shared prefix

		path depth	intermediate proportion
y_1	A B C D	**1**	2/4 = 0.50
y_2	A B E F	**2**	3/4 = 0.75
y_3	A B E G	**2**	3/4 = 0.75
y_4	J K L M	**0**	0/4 = 0

Grey boxes = nodes of the prefix tree

- Both rise over training, sharply near the end
- Shared tokens stay a **small fraction** for most of training



Sullivan & Koller (2026), Fig. 3

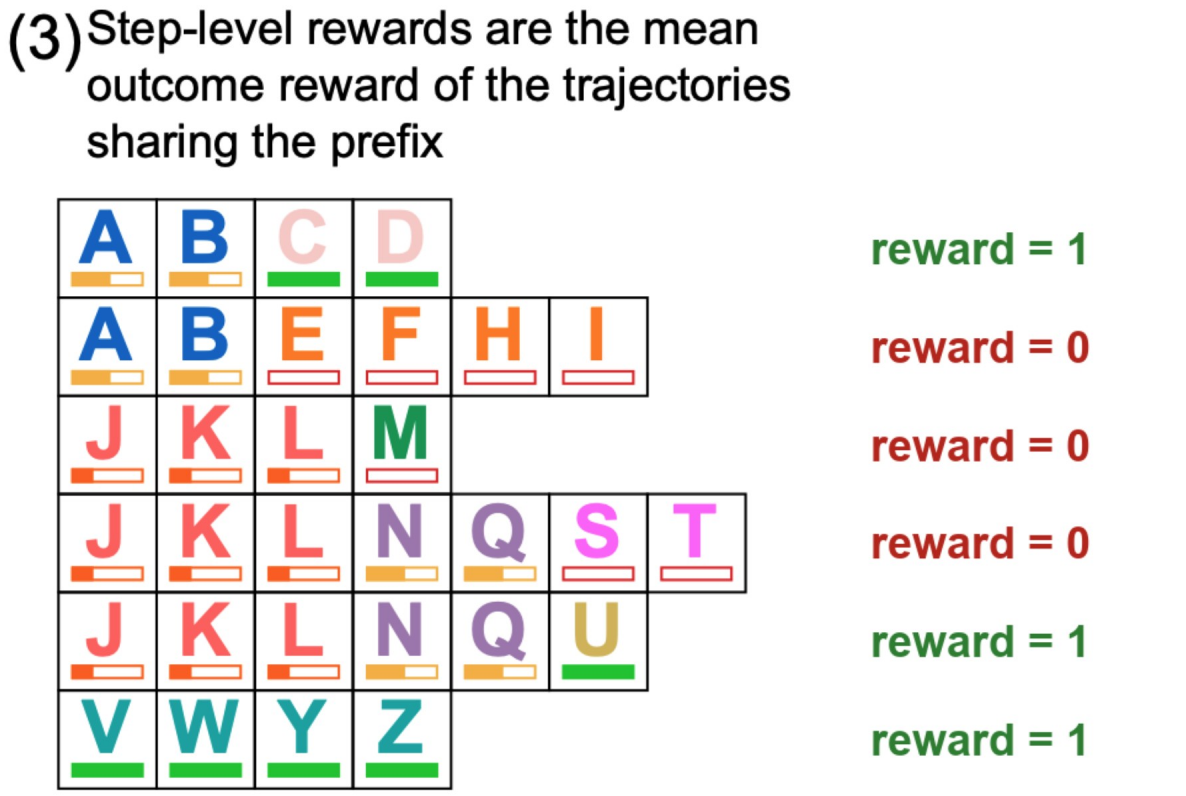
The flaw: shared steps count K times

What goes wrong

- a step shared by K rollouts gets K times the update
- a common good step is pushed up even harder: less exploration
- a below-average step is pushed down harder, even when a great rollout goes through it: less exploitation

What it means

- **GRPO's hidden PRM over-weights frequent steps**

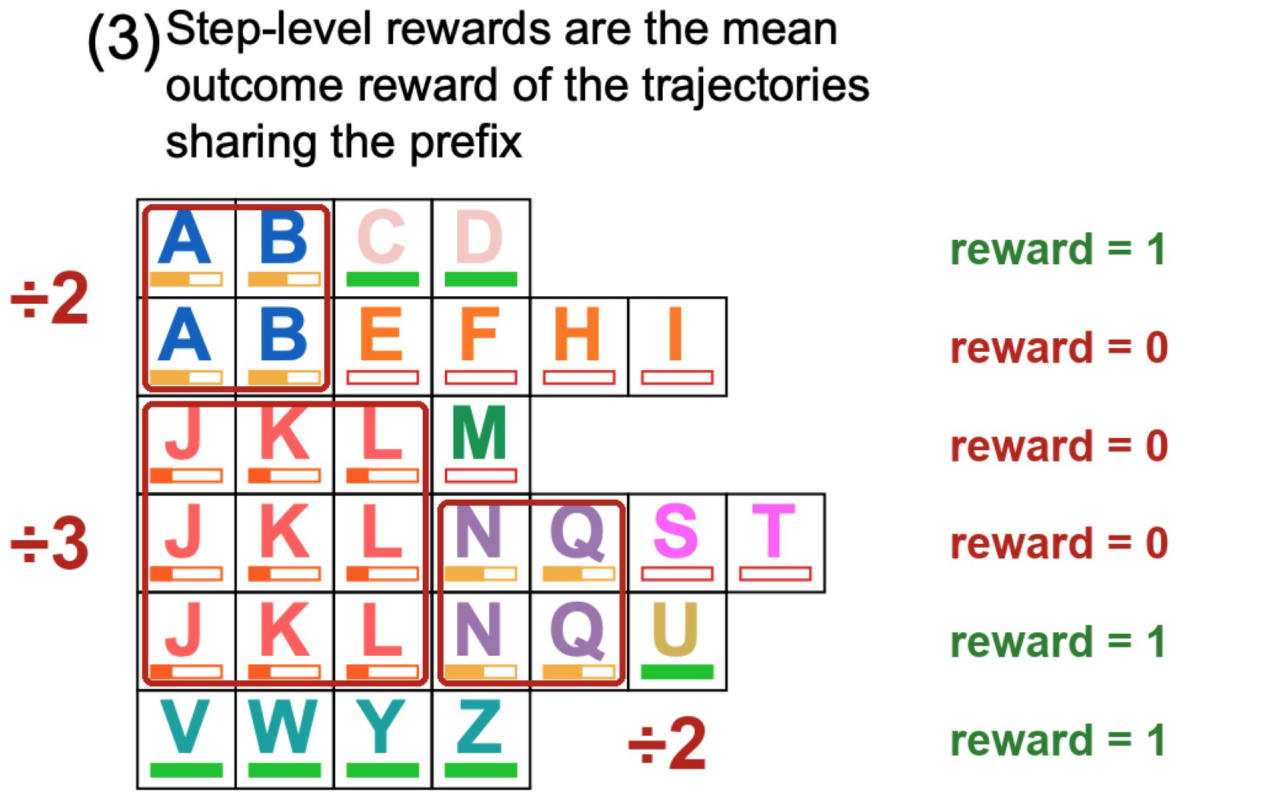


Sullivan & Koller (2026), Fig. 1 (3), reward labels added

λ-GRPO: one vote per step

Divide each token's update by the number of rollouts that share it

- JKL: 3 updates become **one combined vote**
- Unshared steps (e.g. VWYZ) are unchanged
- Changes how hard a step is pushed, never the direction

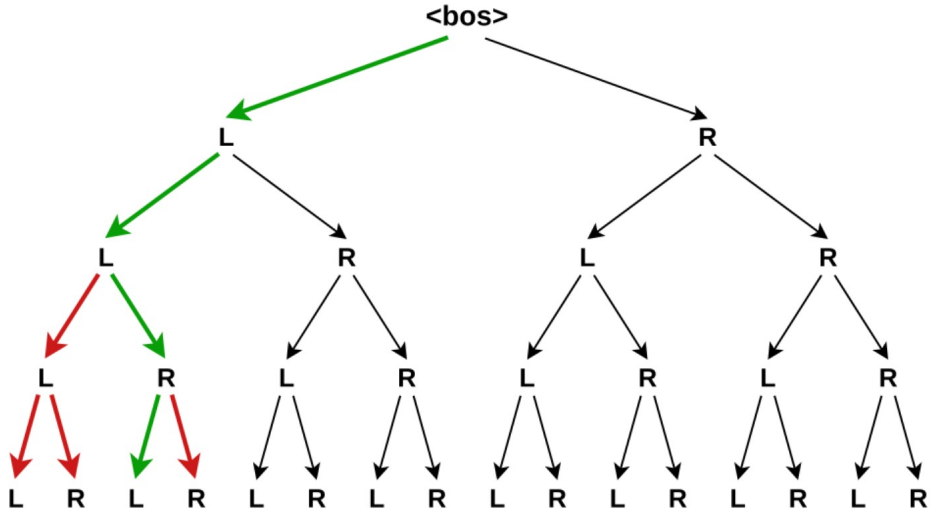


Sullivan & Koller (2026), Fig. 1 (3), annotations added

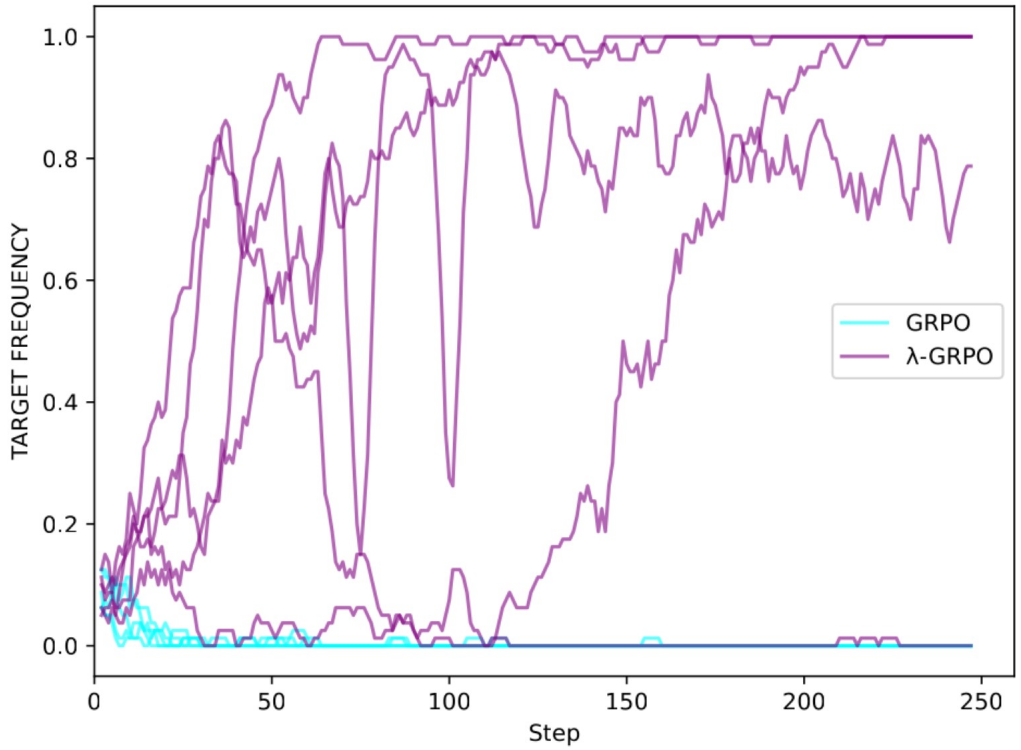
Toy task: does λ-GRPO fix the problem?

Setup

- GPT-2-small (Radford et al., 2019) walks a depth-4 tree
- **Target path: +1**; **other paths starting like it: negative**; everything else: +0.7



Sullivan & Koller (2026), Fig. 4



Sullivan & Koller (2026), Fig. 5: 5 seeds per method

GRPO finds the target early, but never locks onto it; λ-GRPO does in 4 of 5 runs.

Results on reasoning benchmarks

Model	β	Version	AIME24	MATH-500	AMC23	Minerva	OB	Avg.
Qwen	–	Base	0.2000	0.8300	0.7500	**0.2978**	0.5096	0.5175
	0.0	GRPO	0.3333	0.7660	0.6250	0.2500	0.4444	0.4837
		λ-GRPO (ours)	0.3667	0.8460	0.7500	0.2904	0.5348	0.5576
	0.04	GRPO	0.3000	**0.8660**	0.7500	0.2610	0.5200	0.5394
		λ-GRPO (ours)	**0.4000**	0.8340	**0.8000**	**0.2978**	**0.5378**	**0.5739**
Llama	–	Base	0.0000	0.2280	0.0750	0.0478	0.0563	0.0814
	0.0	GRPO	0.0000	0.2300	0.0750	0.0551	0.0607	0.0842
		λ-GRPO (ours)	0.0000	**0.2620**	0.1250	0.0515	**0.0622**	**0.1001**
	0.04	GRPO	0.0000	0.2180	**0.1750**	0.0515	0.0533	0.0996
		λ-GRPO (ours)	**0.0333**	0.2560	0.0750	**0.0735**	0.0489	0.0973

Sullivan & Koller (2026), Table 2 (noise is excluded)
***Bold: best**, underlined: second best, per column within each model*

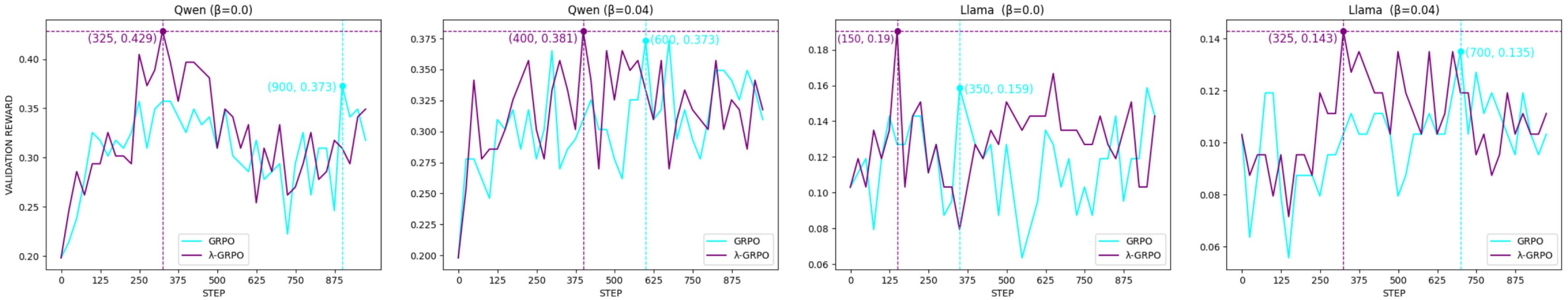
Setup

Qwen 1.5B and Llama 1B, 1000 steps on OpenRS maths, with and without KL; 5 benchmarks

Result

λ-GRPO beats GRPO in **15 of 20** cells and on average in **3 of 4** settings

A closer look at the training curves



Sullivan & Koller (2026), Fig. 6

Very noisy curves

Jumps of 0.05–0.10 between checkpoints and only 125 validation questions

Peaks are single spikes

"Higher peak" and "2× faster" both come from the **best checkpoint of a single run** per setting

Advantage mostly early

λ-GRPO leads early, then the curves overlap.
Longer training is not tested.

Potential limitations

1. Is the ×K weighting really a flaw?

- A prefix appears K times **because** the model gives it high probability
- The policy gradient is estimated by averaging over sampled answers: weighting by frequency is what makes it unbiased
- λ-GRPO drops that weighting: **a different, more exploratory objective**, not just a bug fix

Williams (1992); Sutton & Barto (2018), Ch. 13

2. Early in training, GRPO ≈ ORM

- At the start, rollouts rarely share prefixes: **shallow trees**
- For most of training, only a small fraction of each answer's tokens are shared, so most get plain outcome reward
- Rich process steps appear mainly **near convergence**, when exploration matters less

Sullivan & Koller (2026), Fig. 3

How does it advance the state of the art?

1 **A new way to look at GRPO**

Outcome-reward GRPO already acts like a **Monte Carlo PRM** (cf. VinePPO, TreeRL), but without the extra rollouts *(Kazemnejad et al., 2025; Hou et al., 2025)*

2 **It questions earlier fixes**

Earlier work modified GRPO to get step-level rewards; **part of that signal was already there** *(Shao et al., 2024; Feng et al., 2025)*

3 **A cheap improvement**

λ-GRPO is **one scaling factor** in the loss: no reward model, no extra sampling *(~10 s overhead per training run)*

Same spirit for DPO: Rafailov et al. (2024) show it implicitly learns a token-level reward

References

Cui, G., Yuan, L., Wang, Z., et al. (2025). Process reinforcement through implicit rewards. *arXiv:2502.01456*.

DeepSeek-AI (2025). DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*.

Feng, L., Xue, Z., Liu, T., and An, B. (2025). Group-in-group policy optimization for LLM agent training. *arXiv:2505.10978*.

Hou, Z., Hu, Z., Li, Y., et al. (2025). TreeRL: LLM reinforcement learning with on-policy tree search. In *Proc. ACL 2025*, pp. 12355–12369.

Kazemnejad, A., Aghajohari, M., Portelance, E., et al. (2025). VinePPO: Refining credit assignment in RL training of LLMs. In *Proc. ICML 2025*.

Lightman, H., Kosaraju, V., Burda, Y., et al. (2024). Let's verify step by step. In *Proc. ICLR 2024*.

Radford, A., Wu, J., Child, R., et al. (2019). Language models are unsupervised multitask learners. *OpenAI technical report*.

Rafailov, R., Hejna, J., Park, R., and Finn, C. (2024). From r to Q*: Your language model is secretly a Q-function. In *Proc. COLM 2024*.

Schulman, J., Wolski, F., Dhariwal, P., et al. (2017). Proximal policy optimization algorithms. *arXiv:1707.06347*.

Shao, Z., Wang, P., Zhu, Q., et al. (2024). DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*.

Sullivan, M. and Koller, A. (2026). GRPO is secretly a process reward model. In *Proc. ICML 2026*, PMLR 306. *arXiv:2509.21154*.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

Towards AI. Group Relative Policy Optimization (GRPO) illustrated breakdown. pub.towardsai.net.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4), 229–256.

Yu, Q., Zhang, Z., Zhu, R., et al. (2025). DAPO: An open-source LLM reinforcement learning system at scale. *arXiv:2503.14476*.

Zhang, Z., Zheng, C., Wu, Y., et al. (2025). The lessons of developing process reward models in mathematical reasoning. *arXiv:2501.07301*.