Three ways GRPO wastes rollouts

and one objective that fixes them

Jean Kaddour


RL is the path past imitation.
The world if we get RL to work.

GRPO is the current workhorse of LLM RL

Qwen

2025-05-15

Qwen3 Technical Report

Qwen Team

https://huggingface.co/Qwen

https://modelscope.cn/organization/qwen

https://github.com/QwenLM/Qwen3

Abstract

In this work, we present Qwen3, the latest version of the Qwen model family. Qwen3 comprises a series of large language models (LLMs) designed to advance performance, efficiency, and multilingual capabilities. The Qwen3 series includes models of both dense and Mixture-of-Expert (MoE) architectures, with parameter scales ranging from 0.6 to 235 billion. A key innovation in Qwen3 is the integration of thinking mode (for complex, multi-step reasoning) and non-thinking mode (for rapid, context-driven responses) into a unified framework. This eliminates the need to switch between different models—such as chat-optimized models (e.g., GPT-4o) and dedicated reasoning models (e.g., QwQ-32B)—and enables dynamic mode switching based on user queries or chat templates. Meanwhile, Qwen3 introduces a thinking budget mechanism, allowing users to allocate computational resources adaptively during inference, thereby balancing latency and performance based on task complexity. Moreover, by leveraging the knowledge from the flagship models, we significantly reduce the computational resources required to build smaller-scale models, while ensuring their highly competitive performance. Empirical evaluations demonstrate that Qwen3 achieves state-of-the-art results across diverse benchmarks, including tasks in code generation, mathematical reasoning, agent tasks, etc., competitive against larger MoE models and proprietary models. Compared to its predecessor Qwen2.5, Qwen3 expands multilingual support from 29 to 119 languages and dialects, enhancing global accessibility through improved cross-lingual understanding and generation capabilities. To facilitate reproducibility and community-driven research and development, all Qwen3 models are publicly accessible under Apache 2.0.

GLM-5: from Vibe Coding to Agentic Engineering

GLM-5 Team

Zhipu AI & Tsinghua University

(For the complete list of authors, please refer to the Contribution section)

Abstract

We present GLM-5, a next-generation foundation model designed to transition the paradigm of vibe coding to agentic engineering. Building upon the agentic, reasoning, and coding (ARC) capabilities of its predecessor, GLM-5 adopts DSA to significantly reduce training and inference costs while maintaining long-context fidelity. To advance model alignment and autonomy, we implement a new asynchronous reinforcement learning infrastructure that drastically improves post-training efficiency by decoupling generation from training. Furthermore, we propose novel asynchronous agent RL algorithms that further improve RL quality, enabling the model to learn from complex, long-horizon interactions more effectively. Through these innovations, GLM-5 achieves state-of-the-art performance on major open benchmarks. Most critically, GLM-5 demonstrates unprecedented capability in real-world coding tasks, surpassing previous baselines in handling end-to-end software engineering challenges. Code, models, and more information are available at https://github.com/zai-org/GLM-5.

Figure 1: Results of GLM-5, DeepSeek-V3.2, Claude Opus 4.5, Gemini 3 Pro, and GPT-5.2 (xhigh) on 8 agentic, reasoning, and coding benchmarks: Humanity's Last Exam, SWE-bench Verified, SWE-bench Multilingual, Terminal-Bench 2.0, BrowseComp, MCP-Atlas, τ²-Bench, Vending Bench 2.

deepseek

DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence

DeepSeek-AI

research@deepseek.com

Abstract

We present a preview version of DeepSeek-V4 series, including two strong Mixture-of-Experts (MoE) language models — DeepSeek-V4-Pro with 1.6T parameters (49B activated) and DeepSeek-V4-Flash with 284B parameters (13B activated) — both supporting a context length of one million tokens. DeepSeek-V4 series incorporate several key upgrades in architecture and optimization: (1) a hybrid attention architecture that combines Compressed Sparse Attention (CSA) and Heavily Compressed Attention (HCA) to improve long-context efficiency; (2) Manifold-Constrained Hyper-Connections (mHC) that enhance conventional residual connections; (3) and the Muon optimizer for faster convergence and greater training stability. We pre-train both models on more than 32T diverse and high-quality tokens, followed by a comprehensive post-training pipeline that unlocks and further enhances their capabilities. DeepSeek-V4-Pro-Max, the maximum reasoning effort mode of DeepSeek-V4-Pro, redefines the state-of-the-art for open models, outperforming its predecessors in core tasks. Meanwhile, DeepSeek-V4 series are highly efficient in long-context scenarios. In the one-million-token context setting, DeepSeek-V4-Pro requires only 27% of single-token inference FLOPs and 10% of KV cache compared with DeepSeek-V3.2. This enables us to routinely support one-million-token contexts, thereby making long-horizon tasks and further test-time scaling more feasible. The model checkpoints are available at https://huggingface.co/collections/deepseek-ai/deepseek-v4.

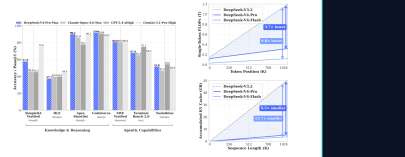


Figure 1 | **Left**: benchmark performance of DeepSeek-V4-Pro-Max and its counterparts. **Right**: inference FLOPs and KV cache size of DeepSeek-V4 series and DeepSeek-V3.2.

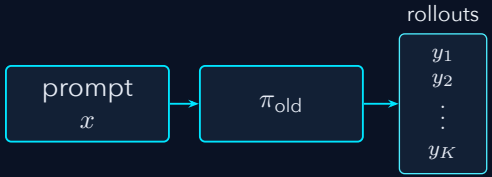
Roadmap

1. Three GRPO shortcomings
2. Target Policy Optimization
3. Revisiting the shortcomings
4. Empirical results
5. Open questions

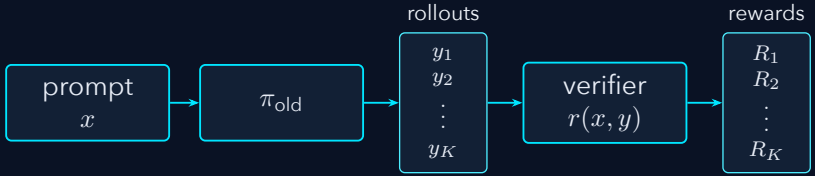
What per-rollout advantages miss

GRPO gives each rollout one advantage

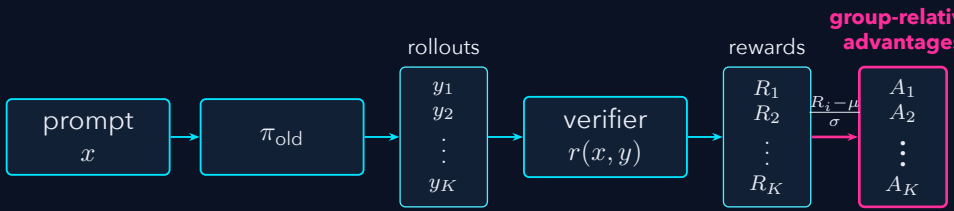
GRPO gives each rollout one advantage



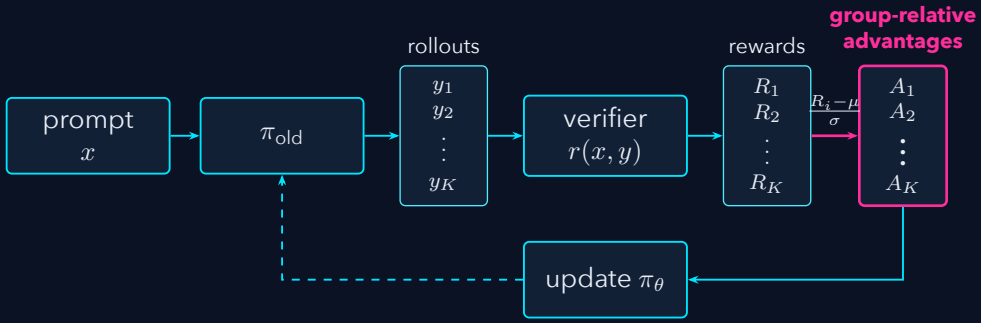
GRPO gives each rollout one advantage



GRPO gives each rollout one advantage



GRPO gives each rollout one advantage



Where do scalar advantages leave signal on the table?

Where do scalar advantages leave signal on the table?

gradient share

easy or hard prompt:
same share?

Where do scalar advantages leave signal on the table?

gradient share

easy or hard prompt: same share?

blunders

common or rare wrong: same A?

Where do scalar advantages leave signal on the table?

gradient share

easy or hard prompt: same share?

blunders

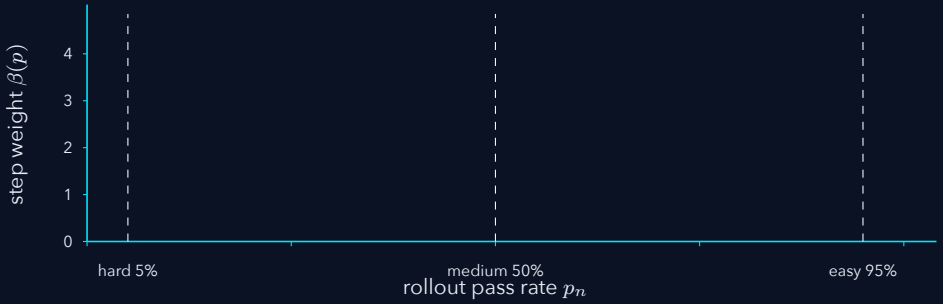
common or rare wrong: same A?

clip wall

common or rare success: same wall?

GRPO over-weights easy prompts

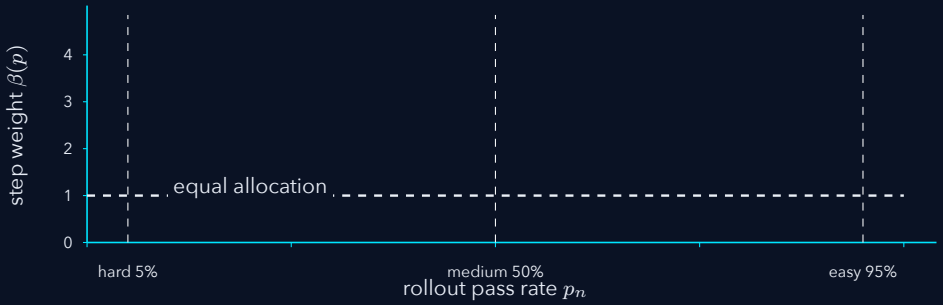
Hard (5%) vs easy (95%) prompt: how does GRPO split the step?



Diagnostic from Osband (2026), *Delightful Policy Gradient*, §4.2.

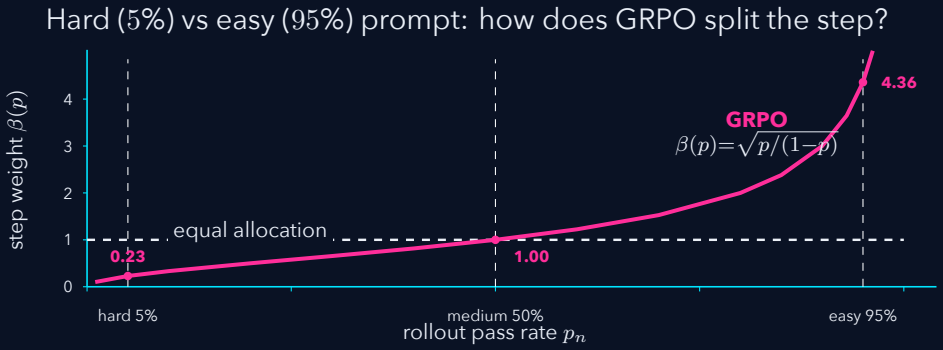
GRPO over-weights easy prompts

Hard (5%) vs easy (95%) prompt: how does GRPO split the step?



Diagnostic from Osband (2026), *Delightful Policy Gradient*, §4.2.

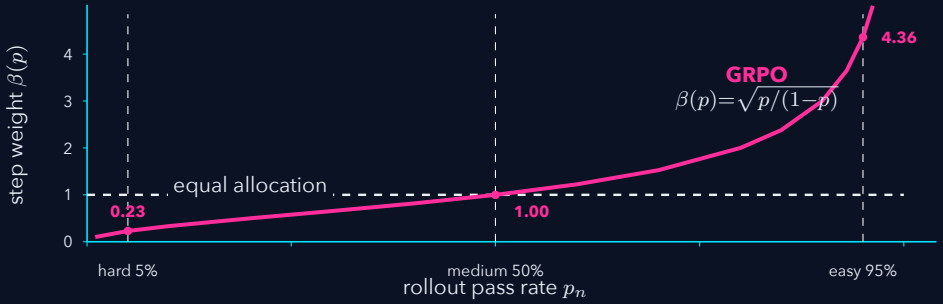
GRPO over-weights easy prompts



Diagnostic from Osband (2026), *Delightful Policy Gradient*, §4.2.

GRPO over-weights easy prompts

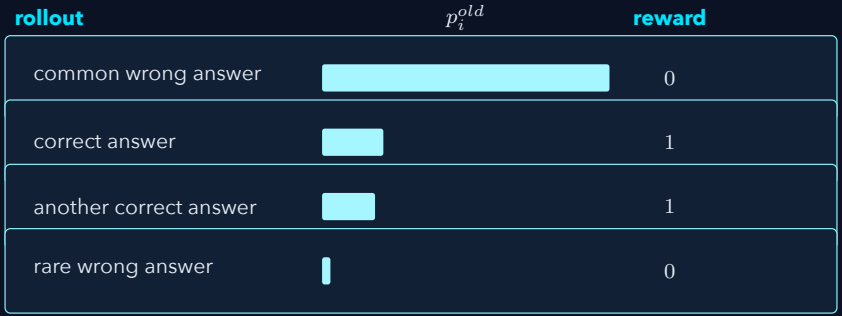
Hard (5%) vs easy (95%) prompt: how does GRPO split the step?



An easy prompt gets **19**× the update size of a hard one.

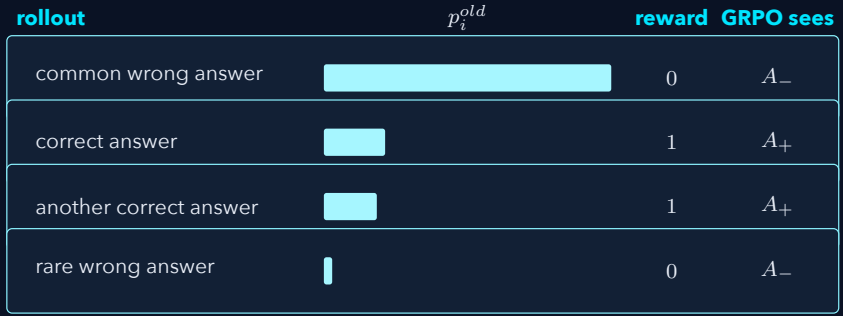
Diagnostic from Osband (2026), *Delightful Policy Gradient*, §4.2.

Rare failures are punished strongly



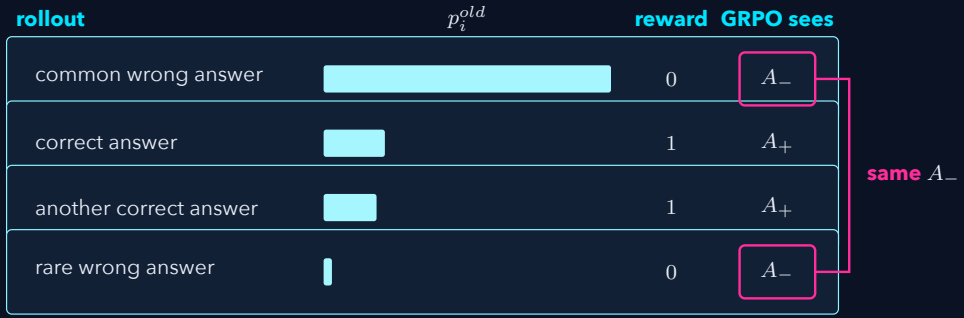
Diagnostic from Osband (2026), *Delightful Policy Gradient*, §4.1.

Rare failures are punished strongly



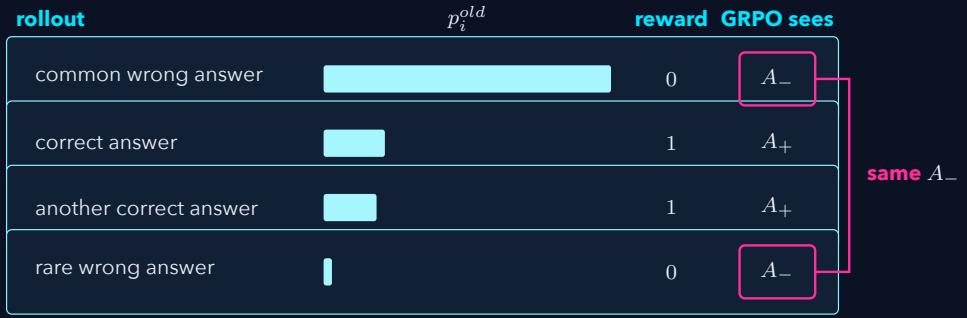
Diagnostic from Osband (2026), *Delightful Policy Gradient*, §4.1.

Rare failures are punished strongly



Diagnostic from Osband (2026), *Delightful Policy Gradient*, §4.1.

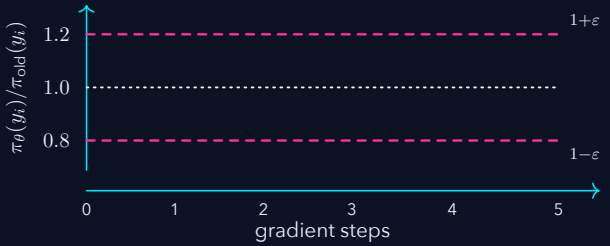
Rare failures are punished strongly



Rare negative-advantage actions can inflate the gradient.

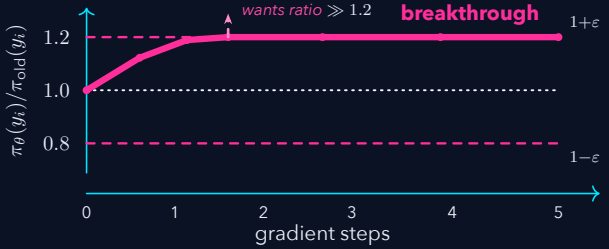
Diagnostic from Osband (2026), *Delightful Policy Gradient*, §4.1.

Same ε for breakthroughs and easy winners



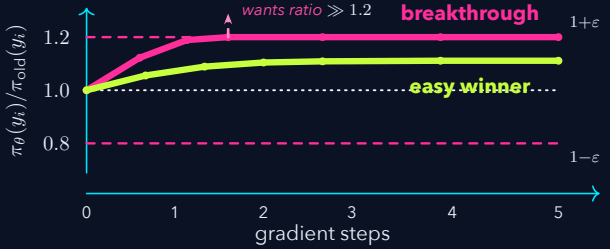
DG (Osband, 2026): breakthroughs need amplification; DAPO's clip-higher at $\varepsilon_{\text{high}}=0.28$.

Same ε for breakthroughs and easy winners



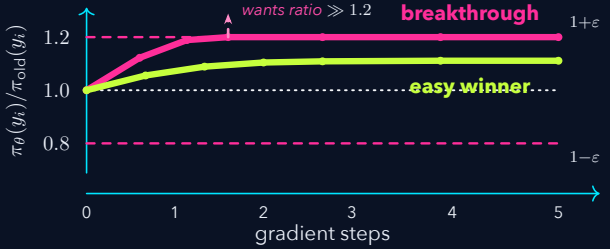
DG (Osband, 2026): breakthroughs need amplification; DAPO's clip-higher at $\varepsilon_{\text{high}}=0.28$.

Same ε for breakthroughs and easy winners



DG (Osband, 2026): breakthroughs need amplification; DAPO's clip-higher at $\varepsilon_{\text{high}}=0.28$.

Same ε for breakthroughs and easy winners



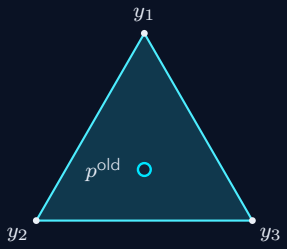
Breakthroughs ($0.01 \to 0.012$) starve, easy winners ($0.9 \to 1.0$) coast.

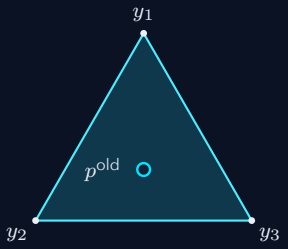
DG (Osband, 2026): breakthroughs need amplification; DAPO's clip-higher at $\varepsilon_{\text{high}} = 0.28$.

Target Policy Optimization

GRPO's target is implicit. TPO is explicit.

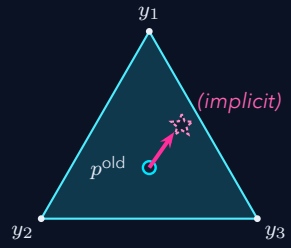
Policy as a distribution over the K sampled rollouts (simplex, $K=3$).

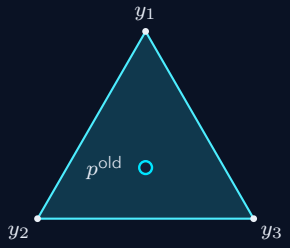




GRPO's target is implicit. TPO is explicit.

Policy as a distribution over the K sampled rollouts (simplex, $K=3$).

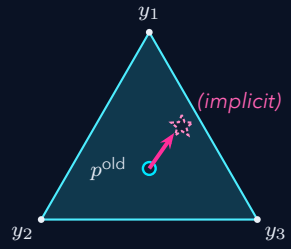




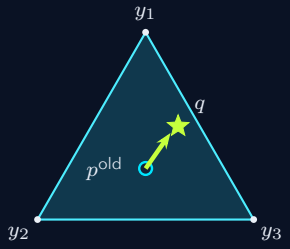
GRPO – push by advantages A_i

GRPO's target is implicit. TPO is explicit.

Policy as a distribution over the K sampled rollouts (simplex, $K=3$).



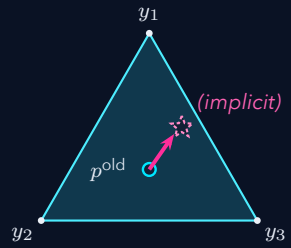
GRPO – push by advantages A_i

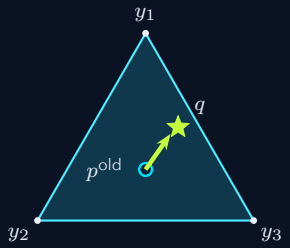


TPO – fit to target q

GRPO's target is implicit. TPO is explicit.

Policy as a distribution over the K sampled rollouts (simplex, $K=3$).





GRPO – push by advantages A_i

TPO – fit to target q

GRPO's target is hidden inside the gradient.
TPO forms the target explicitly, then fits it.

Reweight-then-fit: a 30-year-old idea, now in closed form

Finite group of K rollouts ⇒ closed-form target. No critic, no dual.

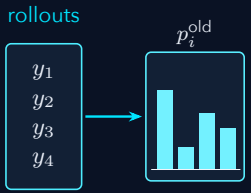
TPO: build a target q, then fit it via cross entropy

rollouts

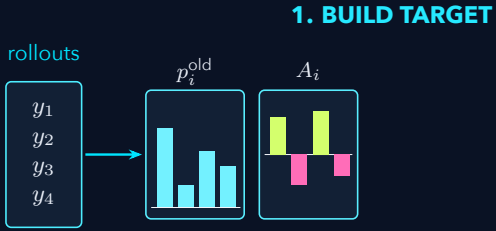
y_1
y_2
y_3
y_4

TPO: build a target q, then fit it via cross entropy

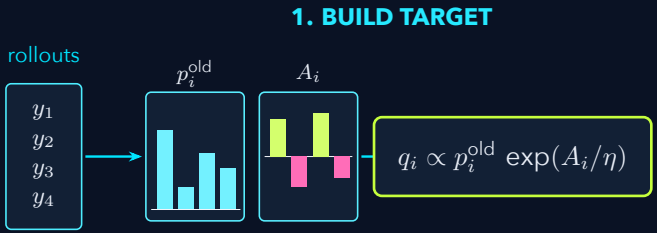
1. BUILD TARGET



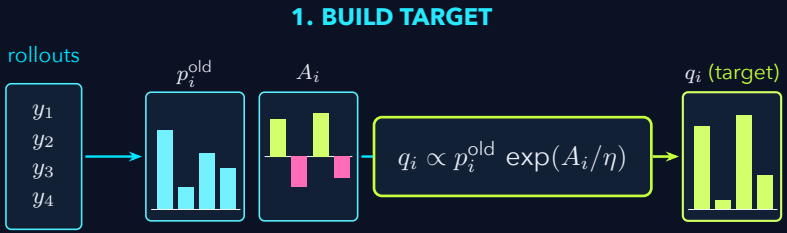
TPO: build a target q, then fit it via cross entropy



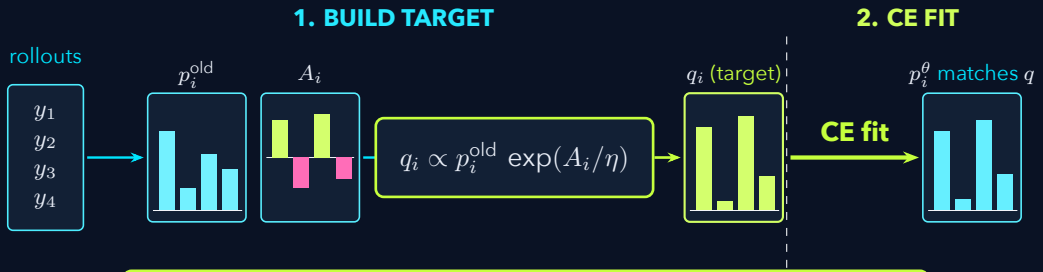
TPO: build a target q, then fit it via cross entropy



TPO: build a target q, then fit it via cross entropy



TPO: build a target q, then fit it via cross entropy



TPO's target is KL-regularized from the rollout policy

q picks the largest **expected advantage** that doesn't drift far from p^{old}:

TPO's target is KL-regularized from the rollout policy

q picks the largest **expected advantage** that doesn't drift far from p^{old}:

$$q = \arg\max_{q \in \Delta^{K-1}} \underbrace{\mathbb{E}_q[A]}_{\text{expected advantage}} - \eta \underbrace{\mathrm{KL}(q \,\|\, p^{\text{old}})}_{\text{stay near rollout}}$$

TPO's target is KL-regularized from the rollout policy

q picks the largest **expected advantage** that doesn't drift far from p^{old}:

$$q \;=\; \arg\max_{q \in \Delta^{K-1}} \underbrace{\mathbb{E}_q[A]}_{\text{expected advantage}} \;-\; \eta\, \underbrace{\mathrm{KL}(q \,\|\, p^{\text{old}})}_{\text{stay near rollout}}$$

Closed form (derivation in paper):

$$q_i \;=\; \frac{p_i^{\text{old}} \exp(A_i/\eta)}{\sum_j p_j^{\text{old}} \exp(A_j/\eta)}$$

TPO's target is KL-regularized from the rollout policy

q picks the largest **expected advantage** that doesn't drift far from p^{old}:

$$q = \arg\max_{q \in \Delta^{K-1}} \underbrace{\mathbb{E}_q[A]}_{\text{expected advantage}} - \eta \underbrace{\text{KL}(q \,\|\, p^{\text{old}})}_{\text{stay near rollout}}$$

Closed form (derivation in paper):

$$q_i = \frac{p_i^{\text{old}} \exp(A_i/\eta)}{\sum_j p_j^{\text{old}} \exp(A_j/\eta)}$$

η is the temperature, we use $\eta = 1$ throughout.

TPO is a few lines of code

JAX

```
def tpo_target(log_scores, adv, eta
    =1.0):
   return jax.nn.softmax(
      jax.nn.log_softmax(log_scores,
           -1)
      + adv / eta, -1)

q = jax.lax.stop_gradient(
   tpo_target(log_scores, adv))
log_p = jax.nn.log_softmax(
    log_scores, -1)
loss = -(q * log_p).sum(-1).mean()
```

PyTorch

```
def tpo_target(log_scores, adv, eta
    =1.0):
   return F.softmax(
      F.log_softmax(log_scores, -1)
      + adv / eta, -1)

q = tpo_target(
   log_scores, adv).detach()
log_p = F.log_softmax(log_scores,
    -1)
loss = -(q * log_p).sum(-1).mean()
```

Three GRPO blind spots. Does TPO close them?

gradient share

easy or hard prompt: same share?

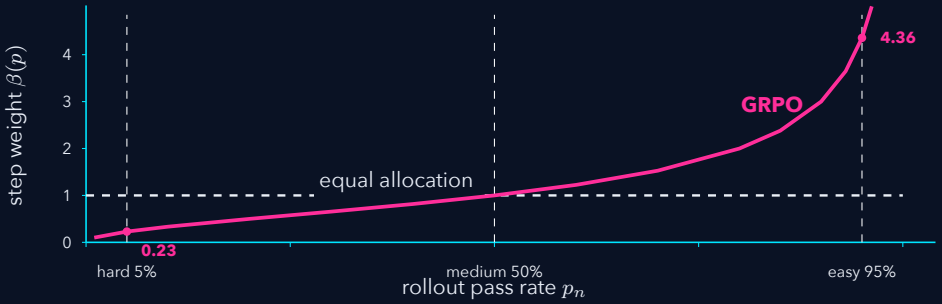
advantage

common or rare wrong: same A?

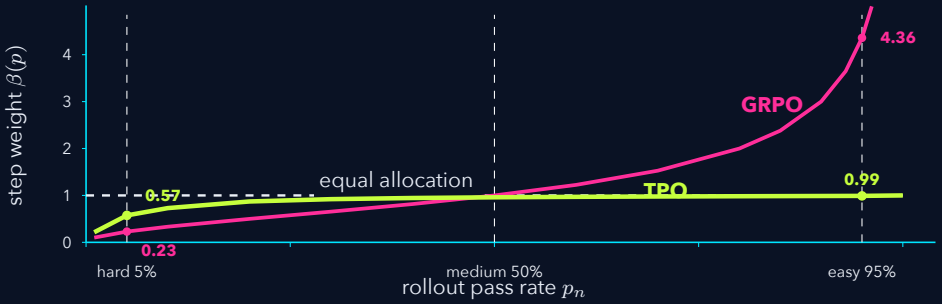
clip wall

common or rare success: same wall?

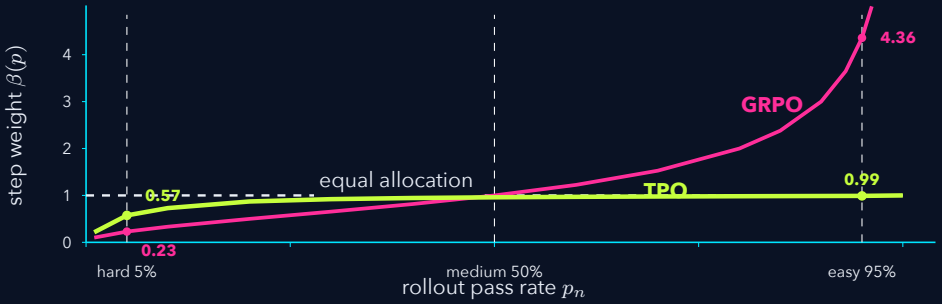
Fix 1: hard prompts are not starved



Fix 1: hard prompts are not starved

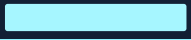


Fix 1: hard prompts are not starved

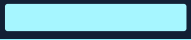


Easy/hard allocation: GRPO ≈ **19**×; TPO ≈ **1.7**×.

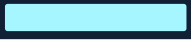
Fix 2: rare failures no longer look common

rollout	p_i^{old}	reward	GRPO
common wrong answer		0	A_-
correct answer		1	A_+
another correct answer		1	A_+
rare wrong answer		0	A_-

Fix 2: rare failures no longer look common

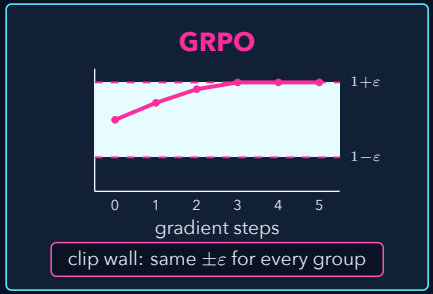
rollout	p_i^{old}	reward	GRPO	TPO: $q_i - p_i^{old}$
common wrong answer		0	A_-	-0.45
correct answer		1	A_+	$+0.25$
another correct answer		1	A_+	$+0.21$
rare wrong answer		0	A_-	-0.01

Fix 2: rare failures no longer look common

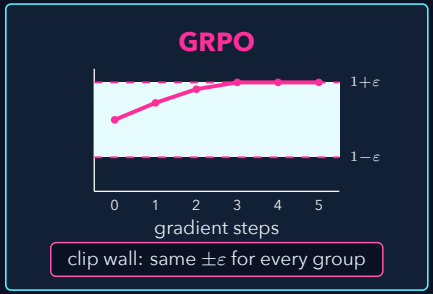
rollout	p_i^{old}	reward	GRPO	TPO: $q_i - p_i^{old}$
common wrong answer		0	A_-	-0.45
correct answer		1	A_+	$+0.25$
another correct answer		1	A_+	$+0.21$
rare wrong answer		0	A_-	-0.01

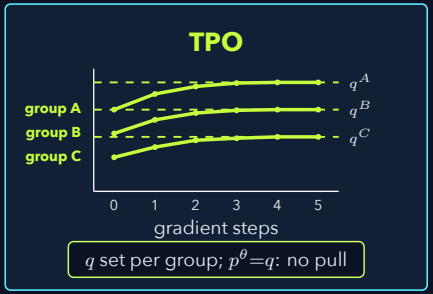
GRPO assigns both wrong rollouts the **same** A_-;
TPO pulls **45**× **harder** on the common one.

Fix 3: every group gets a target, not a wall

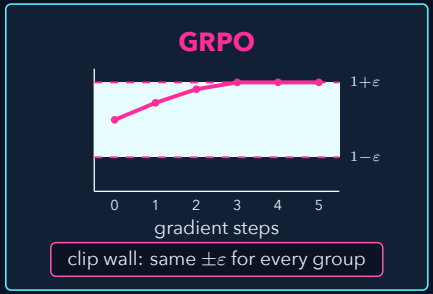


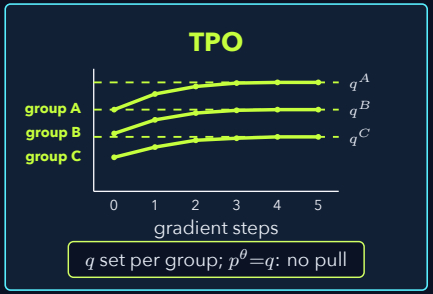
Fix 3: every group gets a target, not a wall





Fix 3: every group gets a target, not a wall



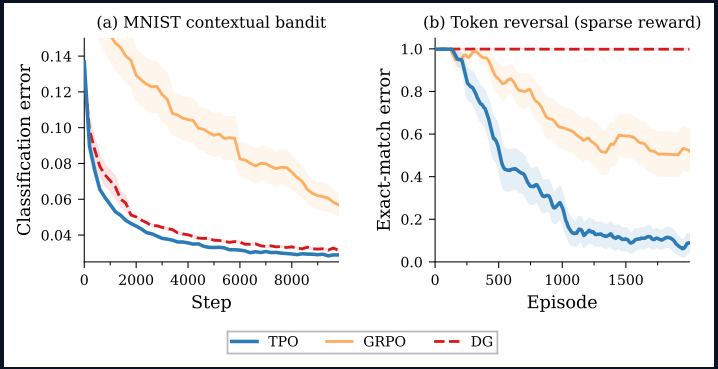


q is a per-group KL-anchored trust region.

Results

Dense bandit and sparse-reward sequence

- **MNIST contextual bandit:** sample digit, reward $\mathbf{1}\{a{=}y\}$.
- **Token reversal:** transformer reverses length-H sequence, reward only at end.



Longer horizons for token reversal

Exact-match error % ($V=2$, $K=8$)

method	$H=7$	$H=8$	$H=9$	$H=10$
TPO	**6.9**	**8.6**	**6.1**	**7.4**
GRPO	14.5	27.6	30.0	50.4
GRPO (no KL)	66.6	92.5	–	–
PPO	12.0	26.3	90.6	–
DG	33.8	58.8	–	–

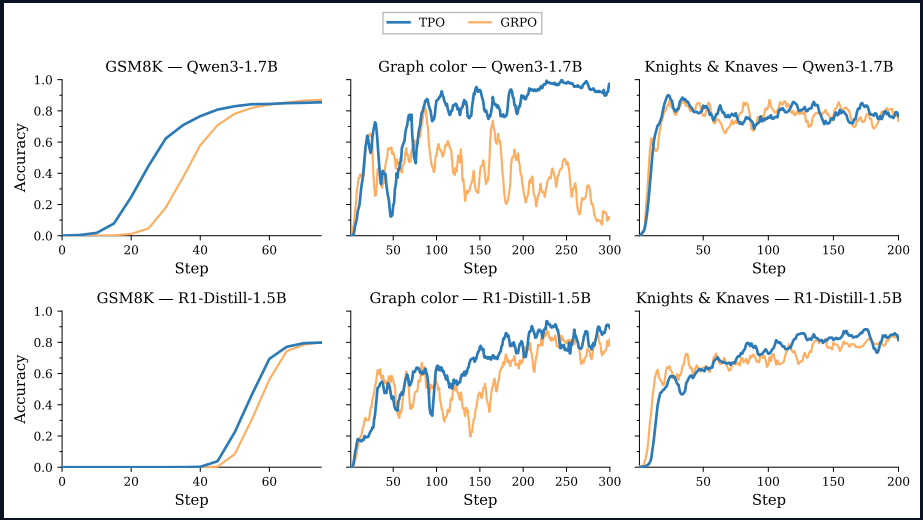
Off-policy with distributed staleness

MNIST contextual bandit; actors roll out with parameters from D steps ago.

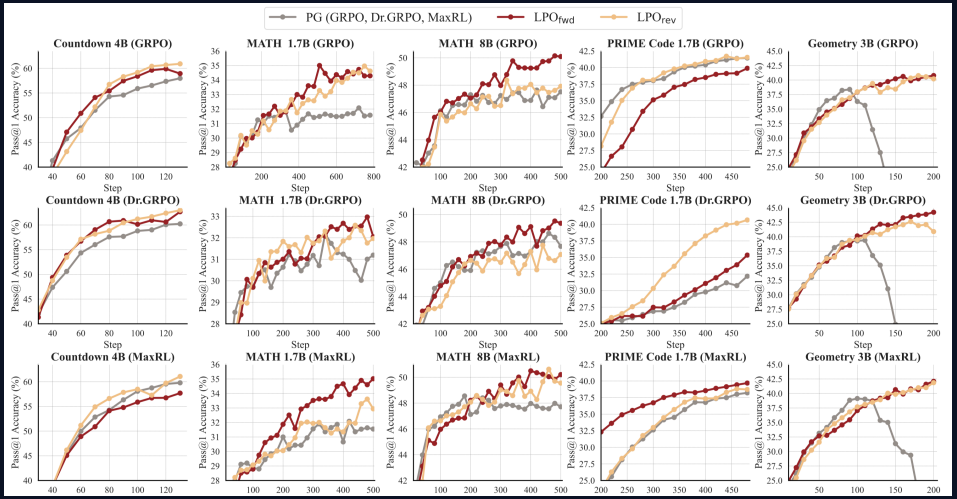
method	D=0	D=1	D=3	D=10	D=30	D=100	D=300	D=1000
REINFORCE	6.3	6.3	6.8	6.5	42.7	67.3	74.9	78.1
PG (IS)	6.3	6.3	6.4	6.8	8.9	10.2	10.1	9.8
PPO	6.3	6.3	7.0	4.1	5.6	5.8	5.9	5.4
DG	2.8	2.8	2.8	2.7	2.6	2.5	2.5	2.5
TPO	**2.3**	**2.3**	**2.3**	**2.3**	**2.3**	**2.3**	**2.3**	**2.3**

Does it transfer to LLMs?

Qwen3-1.7B (top), R1-Distill-1.5B (bottom); $K=16$ rollouts per prompt.



Follow-up by Tencent: $\text{LPO}_{\text{fwd}} \approx \text{TPO}\ (q \to w^*, p_\theta \to P_\theta)$



Open questions

Scale and off-policy stress tests

Larger models, harder benchmarks – does the closed-form target keep paying off at 7B+ on MATH / AIME?

Stale rollouts & replay – $D{=}1000$ already holds; push into off-policy with Retrace / V-trace?

Distributed friction – actor bugs, reward corruption, rare discovery (Osband, 2026).

Richer target-projection design

Low-variance groups – z-scoring makes tiny score gaps look sharp; when does the bias hurt?

All-negative groups – softmax still redistributes mass; does "least bad teaches" actually learn?

Other projections – reverse KL, JS, α-divergences, adaptive schedules (LPO is one step).

Thank you

Questions?

Appendix: Mathematical results

Proposition 1 closed-form target and fixed point

Assume $p_i^{\text{old}} > 0$ for every sampled candidate. Then:

(i) The target $q_i = \dfrac{p_i^{\text{old}} \exp(A_i/\eta)}{\sum_j p_j^{\text{old}} \exp(A_j/\eta)}$ is the **unique maximizer** of

$$\max_{r \in \Delta^{K-1}} \; \mathbb{E}_r[A] \; - \; \eta \, \mathrm{KL}(r \,\|\, p^{\text{old}}).$$

(ii) Treating q as fixed, $\nabla_{\ell^\theta} \mathcal{L}_{\text{TPO}} = p^\theta - q$. The **unique stationary distribution** over the group is $p^\theta = q$.

Proof on the next four slides.

Proof of (i) strict concavity ⇒ unique interior max

Objective on Δ^{K-1} (with $p_i^{\text{old}} > 0$): $J(r) = \sum_i r_i A_i - \eta \sum_i r_i \log \frac{r_i}{p_i^{\text{old}}}$

1. **Concavity.** Linear term: concave. Negative-entropy term: Hessian on the open simplex is $-\eta \operatorname{diag}(1/r_i) \prec 0$. So J is **strictly concave** on $\operatorname{int} \Delta^{K-1}$.
2. **Interior maximizer.** $\partial J / \partial r_i = A_i - \eta(\log \frac{r_i}{p_i^{\text{old}}} + 1) \to +\infty$ as $r_i \to 0^+$, so the boundary is suboptimal. Equality KKT conditions suffice.

Strict concavity + compact Δ^{K-1} + interior optimum ⇒ **unique stationary point**.

Proof of (i) Lagrangian ⇒ closed-form target

Lagrangian for $\sum_i r_i = 1$:

$$L(r, \mu) = \sum_i r_i A_i - \eta \sum_i r_i \log \frac{r_i}{p_i^{\text{old}}} - \mu\left(\sum_i r_i - 1\right)$$

First-order condition $\partial L / \partial r_i = 0$:

$$A_i - \eta\left[\log \frac{r_i}{p_i^{\text{old}}} + 1\right] - \mu = 0 \implies r_i = p_i^{\text{old}}\, e^{A_i/\eta}\, e^{-(\eta+\mu)/\eta}$$

Normalize via $\sum_i r_i = 1$, absorbing μ:

$$q_i = \frac{p_i^{\text{old}} \exp(A_i/\eta)}{\sum_j p_j^{\text{old}} \exp(A_j/\eta)}$$

Maximizer is the **exponential tilt** of p^{old} by A/η. **Part (i)** □

Proof of (ii) cross-entropy gradient is $p^\theta - q$

Loss + group-softmax (q fixed): $\mathcal{L}_{\text{TPO}} = -\sum_i q_i \log p_i^\theta,\ p_i^\theta = e^{\ell_i^\theta} / \sum_j e^{\ell_j^\theta}$

Log-softmax derivative:

$$\log p_i^\theta = \ell_i^\theta - \log \sum_j e^{\ell_j^\theta} \implies \partial \log p_i^\theta / \partial \ell_k^\theta = \delta_{ik} - p_k^\theta$$

Chain rule, using $\sum_i q_i = 1$:

$$\partial \mathcal{L}_{\text{TPO}} / \partial \ell_k^\theta = -\sum_i q_i (\delta_{ik} - p_k^\theta) = -q_k + p_k^\theta \sum_i q_i = p_k^\theta - q_k$$

$$\nabla_{\ell^\theta} \mathcal{L}_{\text{TPO}} = p^\theta - q$$

Proof of (ii) unique stationary distribution

From the previous slide:

$$\nabla_{\ell^\theta} \mathcal{L}_{\text{TPO}} = p^\theta - q$$

Stationarity:

$$\nabla_{\ell^\theta} \mathcal{L}_{\text{TPO}} = 0 \iff p^\theta = q$$

- The softmax $\ell^\theta \mapsto p^\theta$ has a one-dimensional kernel (additive shifts of ℓ^θ), but the **distribution** p^θ over the group is uniquely determined.
- Therefore the unique stationary distribution over the sampled group is $p^\theta = q$. **Part (ii)** □

Combining (i) and (ii): the KL-regularized target q is the **unique fixed point** of TPO's cross-entropy update.

Tabular weights $\beta(p_n)$: same direction, different scale

One-hot reward, logit space. Every method's exact update has direction $g_n = \beta(p_n)\,(e_{y_n} - \pi_n)$. Methods differ only in $\beta(p_n)$.

method	$\beta(p_n)$	**at** p_n=0.1, M=10
CE (oracle)	1	1.00
DG	$\dfrac{p_n}{1+p_n}$	0.09
GRPO	$\sqrt{\dfrac{p_n}{1-p_n}}$	0.33
TPO	$\dfrac{p_n(\lambda-1)}{1-p_n+\lambda p_n}$	0.73

$\lambda = \exp\!\left(M/\sqrt{M-1}\right) \approx 28$ for M=10.

Deriving $\beta(p_n)$ setup

- **Tabular bandit.** M actions, correct y_n, reward $r(a) = \mathbf{1}\{a = y_n\}$.
- **Policy.** π_n in context n; $p_n = \pi_n(y_n)$.
- **Supervised direction.** $v_n = e_{y_n} - \pi_n$ (correct one-hot minus current policy).

Claim. Every method's exact population update has the form

$$g_n \;=\; \beta(p_n)\,(e_{y_n} - \pi_n) \;=\; \beta(p_n)\,v_n.$$

Methods differ only in the scalar $\beta(p_n)$.

CE oracle (trivial): $g_n^{\mathsf{CE}} = e_{y_n} - \pi_n = v_n \implies \beta_{\mathsf{CE}} = 1.$

Deriving β_{DG} sigmoid gate on reward

Exact population DG update (baseline $b = 0$):

$$g_n^{\mathrm{DG}} \;=\; \sum_a \pi_n(a)\, r(a)\, \sigma\!\left(\frac{r(a)\,(-\log \pi_n(a))}{\eta}\right)(e_a - \pi_n)$$

Only $a = y_n$ has $r(a) = 1$; every other term vanishes:

$$g_n^{\mathrm{DG}} \;=\; p_n\, \sigma\!\left(\frac{-\log p_n}{\eta}\right) v_n.$$

With $\eta = 1$: $\quad \sigma(-\log p_n) = \dfrac{1}{1 + e^{\log p_n}} = \dfrac{1}{1 + p_n}.$

$$\boxed{\beta_{\mathrm{DG}}(p_n) \;=\; \frac{p_n}{1 + p_n}}$$

Deriving β_{GRPO} standardized Bernoulli reward

Within context n, reward is Bernoulli(p_n); $\sigma_n = \sqrt{p_n(1-p_n)}$. Standardized:

$$A(y_n) = \frac{1-p_n}{\sigma_n}, \qquad A(a \neq y_n) = \frac{-p_n}{\sigma_n}.$$

Exact population update: $g_n^{\text{GRPO}} = \sum_a \pi_n(a)\, A(a)\, (e_a - \pi_n)$.

Split correct vs incorrect, using the identity $\sum_{a \neq y_n} \pi_n(a)(e_a - \pi_n) = -p_n\, v_n$:

$$g_n^{\text{GRPO}} \;=\; \frac{p_n(1-p_n)}{\sigma_n}\, v_n \;+\; \frac{-p_n}{\sigma_n}\,(-p_n\, v_n) \;=\; \frac{p_n(1-p_n) + p_n^2}{\sigma_n}\, v_n \;=\; \frac{p_n}{\sigma_n}\, v_n.$$

$$\beta_{\text{GRPO}}(p_n) \;=\; \frac{p_n}{\sqrt{p_n(1-p_n)}} \;=\; \sqrt{\frac{p_n}{1-p_n}}$$

Deriving β_{TPO} 1/2: standardize + target

Score vector $s = e_{y_n}$: $\bar{s} = 1/M, \quad \sigma(s) = \sqrt{M-1}/M.$

Standardized scores:

$$A_{y_n} = \frac{1 - 1/M}{\sqrt{M-1}/M} = \sqrt{M-1}, \qquad A_{a \neq y_n} = \frac{-1/M}{\sqrt{M-1}/M} = -\frac{1}{\sqrt{M-1}}.$$

Correct-vs-incorrect tilt factor:

$$\lambda = \exp(A_{y_n} - A_{a \neq y_n}) = \exp\left(\sqrt{M-1} + \tfrac{1}{\sqrt{M-1}}\right) = \exp\left(\frac{M}{\sqrt{M-1}}\right). \quad \text{(For } M{=}10\text{: } \lambda \approx 28.\text{)}$$

Target $q_n(a) \propto \pi_n(a)\, e^{A_a}$. Normalizer factors out $e^{A_{a \neq y_n}}$, leaving $D \equiv 1 - p_n + \lambda p_n$:

$$q_n(y_n) \;=\; \frac{\lambda p_n}{D}, \qquad q_n(a) \;=\; \frac{\pi_n(a)}{D} \;\; (a \neq y_n).$$

Deriving β_{TPO} 2/2: closed-form β

TPO update direction: $g_n^{\text{TPO}} = q_n - \pi_n$. Let $D = 1 - p_n + \lambda p_n$.

- **Correct coordinate** ($a = y_n$): $q_n(y_n) - p_n = \dfrac{\lambda p_n}{D} - p_n = \dfrac{p_n(\lambda - D)}{D}$.

  Since $\lambda - D = (\lambda - 1)(1 - p_n)$, this equals $\dfrac{p_n(\lambda - 1)}{D}(1 - p_n) = \beta_{\text{TPO}}\, v_n(y_n)$.
- **Incorrect coordinate** ($a \neq y_n$):

  $q_n(a) - \pi_n(a) = \pi_n(a)\left(\frac{1}{D} - 1\right) = -\dfrac{p_n(\lambda - 1)}{D}\,\pi_n(a) = \beta_{\text{TPO}}\, v_n(a)$.

  (Using $1 - D = -p_n(\lambda - 1)$ and $v_n(a) = -\pi_n(a)$.)

Both coordinates collinear with v_n, with the same scalar:

$$\beta_{\text{TPO}}(p_n) \;=\; \frac{p_n(\lambda - 1)}{1 - p_n + \lambda p_n}$$

MNIST: expected logit update per single sample

One labeled (x, y); $p = \pi_y$; $v = e_y - \pi$. Expectation over $a \sim \pi$.

method	$\mathbb{E}[g]$
PG	$p\,v$
GRPO (single sample, batch-std)	$(p/\sigma_B)\,v$
Group PG	$6p\,v \;=\; 6\,g^{\mathrm{PG}}$
DG	in general $\not\propto v$
TPO, sampled $a = y$	$\beta_+(p)\,(e_y - \pi)$
TPO, sampled $a = j \neq y$	$\gamma(\pi_j)\,(\boldsymbol{\pi - e_j})$

$\beta_+(p) = \frac{p(\lambda-1)}{1-p+\lambda p}$, $\gamma(r) = \frac{r(\lambda-1)}{\lambda(1-r)+r}$, $\lambda = \exp(10/3) \approx 28$.