FlowRL: Matching Reward Distributions for LLM Reasoning

Distribution-Matching via Flow Balancing for Diverse & Generalizable Reasoning

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, Qizheng Zhang, Lin Chen, Fanghao Shao, Bo Xue, Yunchong Song, Zhenjie Yang, Ganqu Cui, Ning Ding, Jianfeng Gao, Xiaodong Liu, Bowen Zhou, Hongyuan Mei, Zhouhan Lin

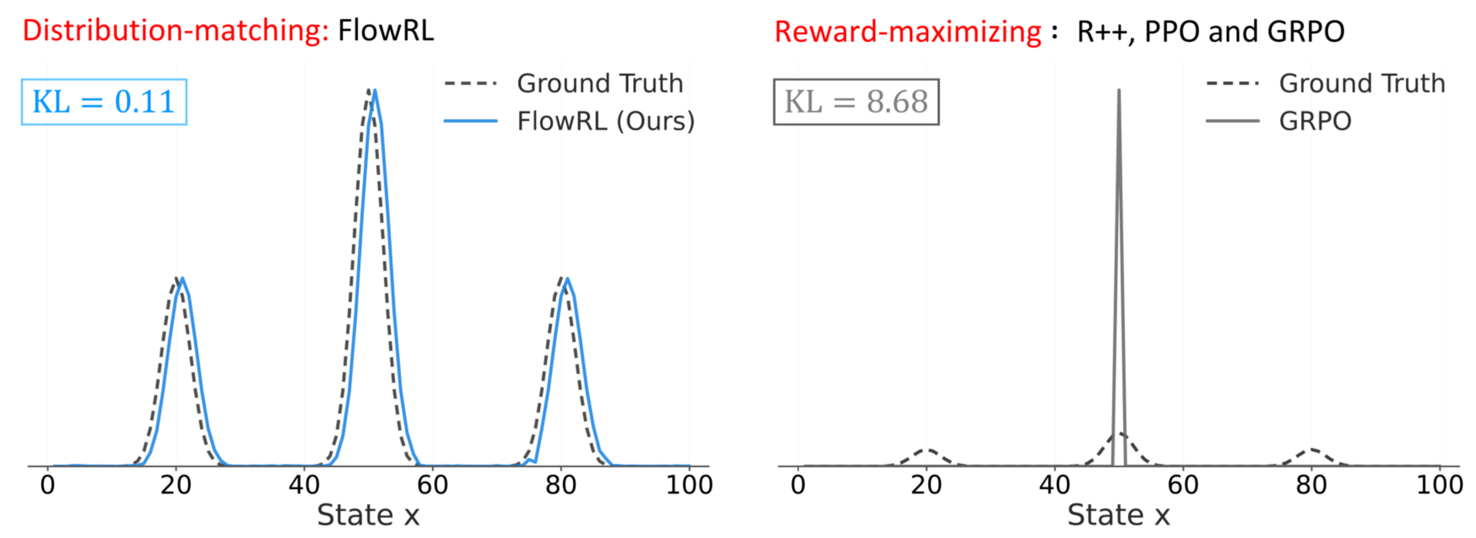
SJTU · Shanghai AI Lab · Microsoft Research
Tsinghua · PKU · RUC · Stanford · TTIC

ICLR 2026
arXiv: 2509.15207

Why Distribution Matching?

REINFORCE++, PPO, and GRPO all share the same objective: maximize expected reward. This drives the policy toward a single high-reward mode, ignoring equally valid but different reasoning paths.

We propose to instead match the full reward distribution — preserving diversity by design.



Theoretical Foundation

Proposition 1 (Sec. 3.2). In terms of expected gradients, minimizing the KL divergence above is equivalent to minimizing the GFlowNet trajectory balance loss:

$$D_{KL}\left(\pi_\theta \,\|\, \frac{e^{\beta r}}{Z_\phi}\right) \iff \left(\log Z_\phi + \log \pi_\theta - \beta r\right)^2$$

Two practical issues arise when applying this to LLMs:

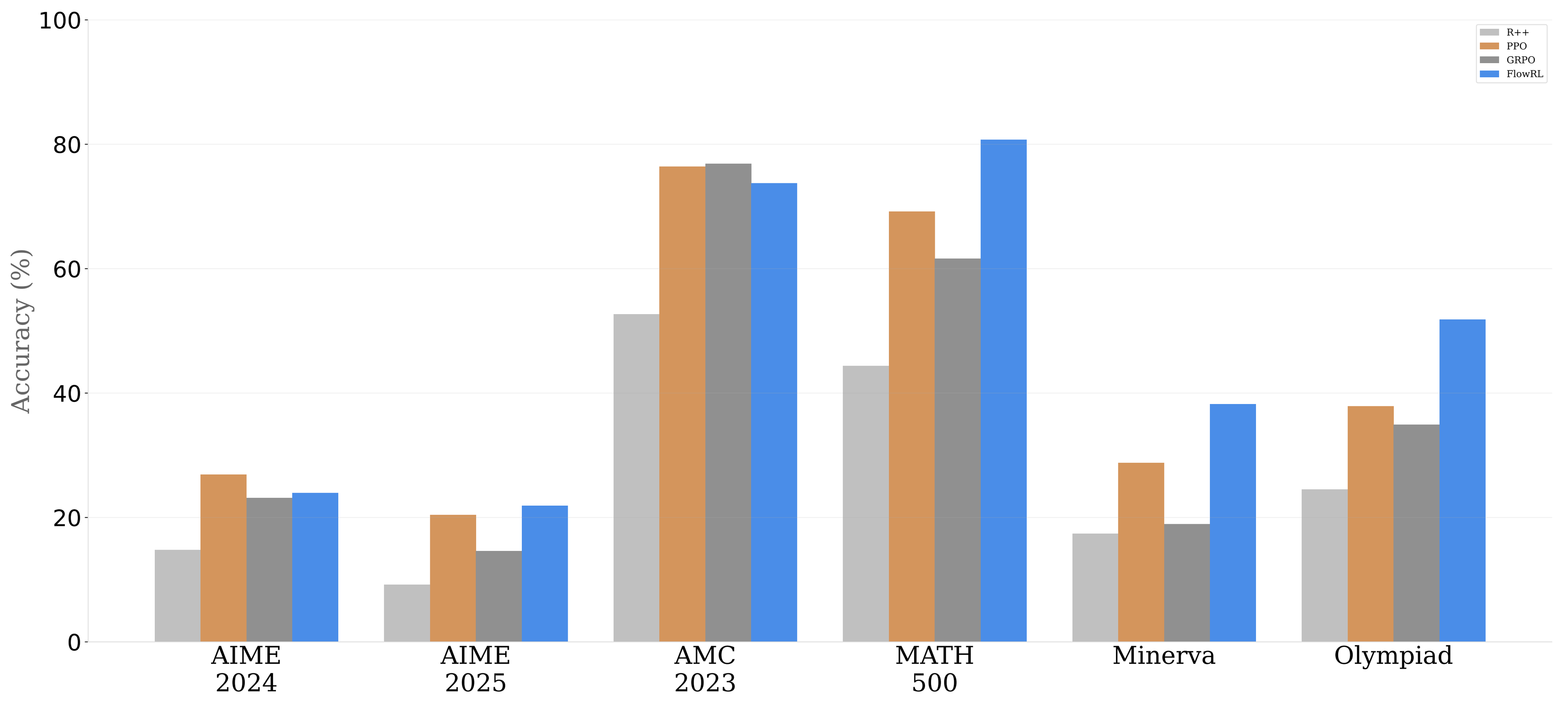
1. Exploding gradients: log π grows with sequence length. We apply length normalization.
2. Off-policy mismatch: trajectory balance requires on-policy samples. We use clipped importance weights.

Final objective:

$$\mathcal{L} = w \cdot \left(\log Z_\phi + \tfrac{1}{|y|}\log \pi_\theta - \beta \hat{r} - \tfrac{1}{|y|}\log \pi_{\text{ref}}\right)^2$$

Math Benchmarks

Qwen2.5-32B, Avg@16



48.4% 32B avg accuracy

+10.1 vs GRPO

+5.1 vs PPO

Our Approach

We introduce a learnable partition function Zφ(x) that normalizes rewards into a proper distribution, then train the policy πθ to match it via reverse KL:

$$\min_\theta D_{KL}\left(\pi_\theta(y \mid x) \,\|\, \frac{\exp(\beta r(x,y))}{Z_\phi(x)}\right) \implies \pi_\theta \propto \exp(\beta r)$$

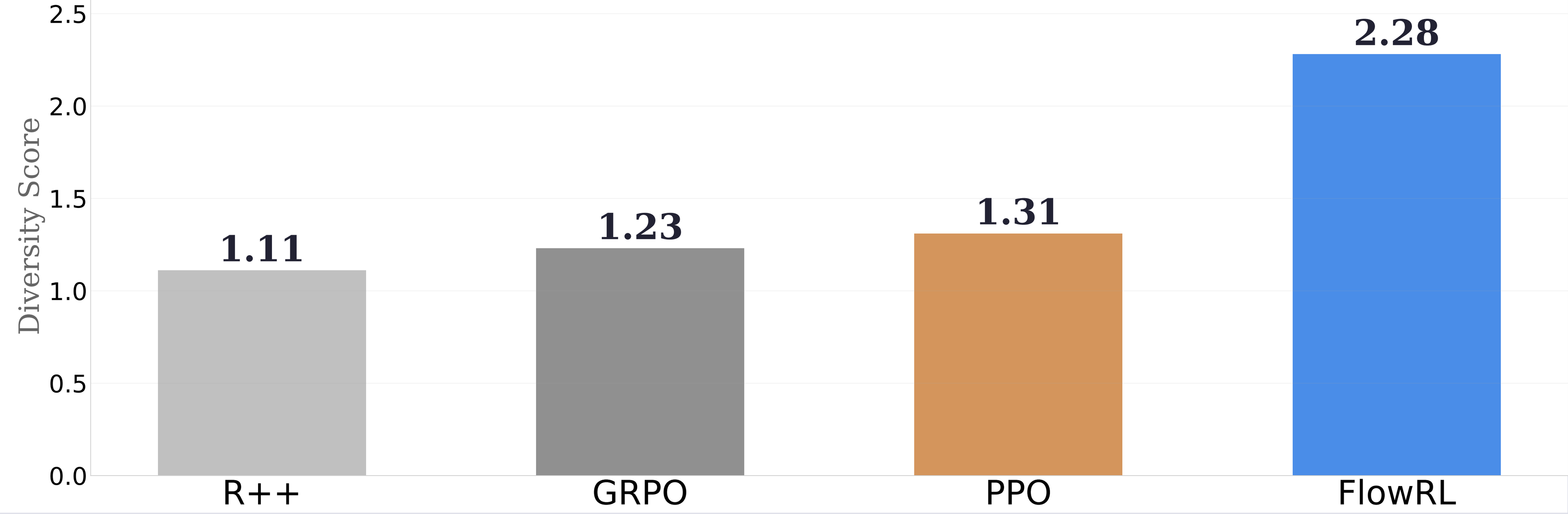
Training pipeline:

Scalar Reward r → **Partition Zφ(x)** → **Flow Balance** → **Diverse Policy πθ**

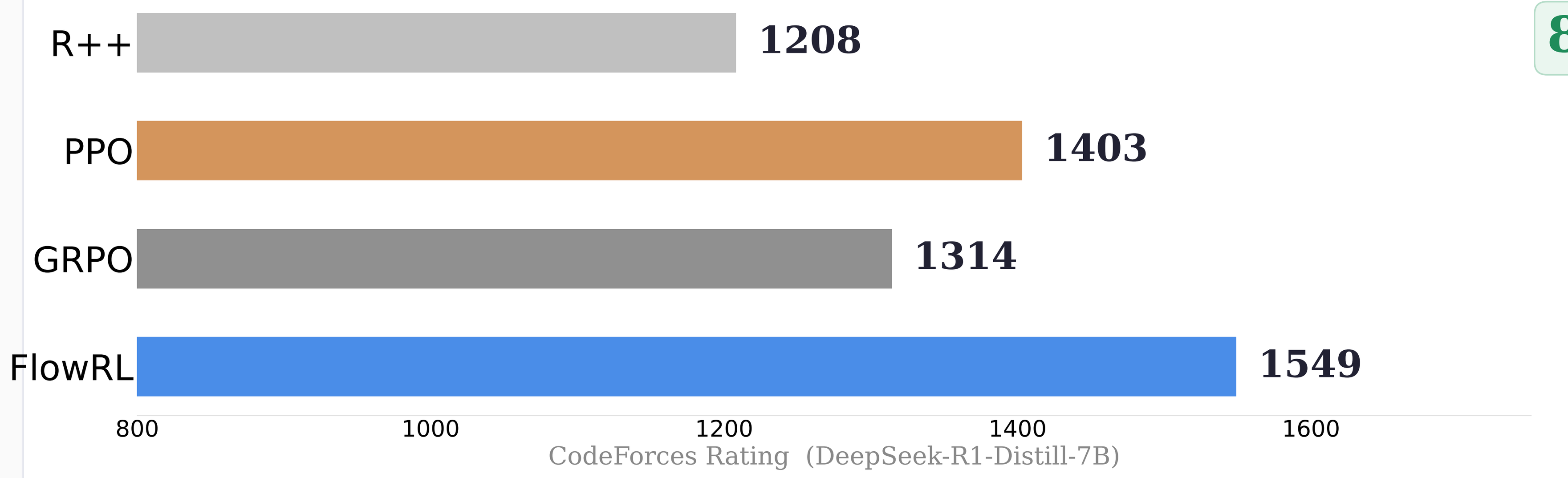
Zφ: 3-layer MLP · veRL framework · β=15 · rollout n=8 · max 8K tokens

Reasoning Diversity

We measure diversity by asking GPT-4o-mini to judge whether Avg@16 rollouts on AIME 24/25 yield genuinely different solution strategies (not just rephrasing). FlowRL roughly doubles the diversity of all baselines.



Code Benchmarks



83.3% percentile

Summary

1. We reframe LLM RL as distribution matching rather than reward maximization.
2. Proposition 1 connects KL minimization to GFlowNet trajectory balance.
3. FlowRL outperforms GRPO by +10.1 and PPO by +5.1 across 6 math benchmarks.
4. Solution diversity roughly doubles versus all baselines tested.