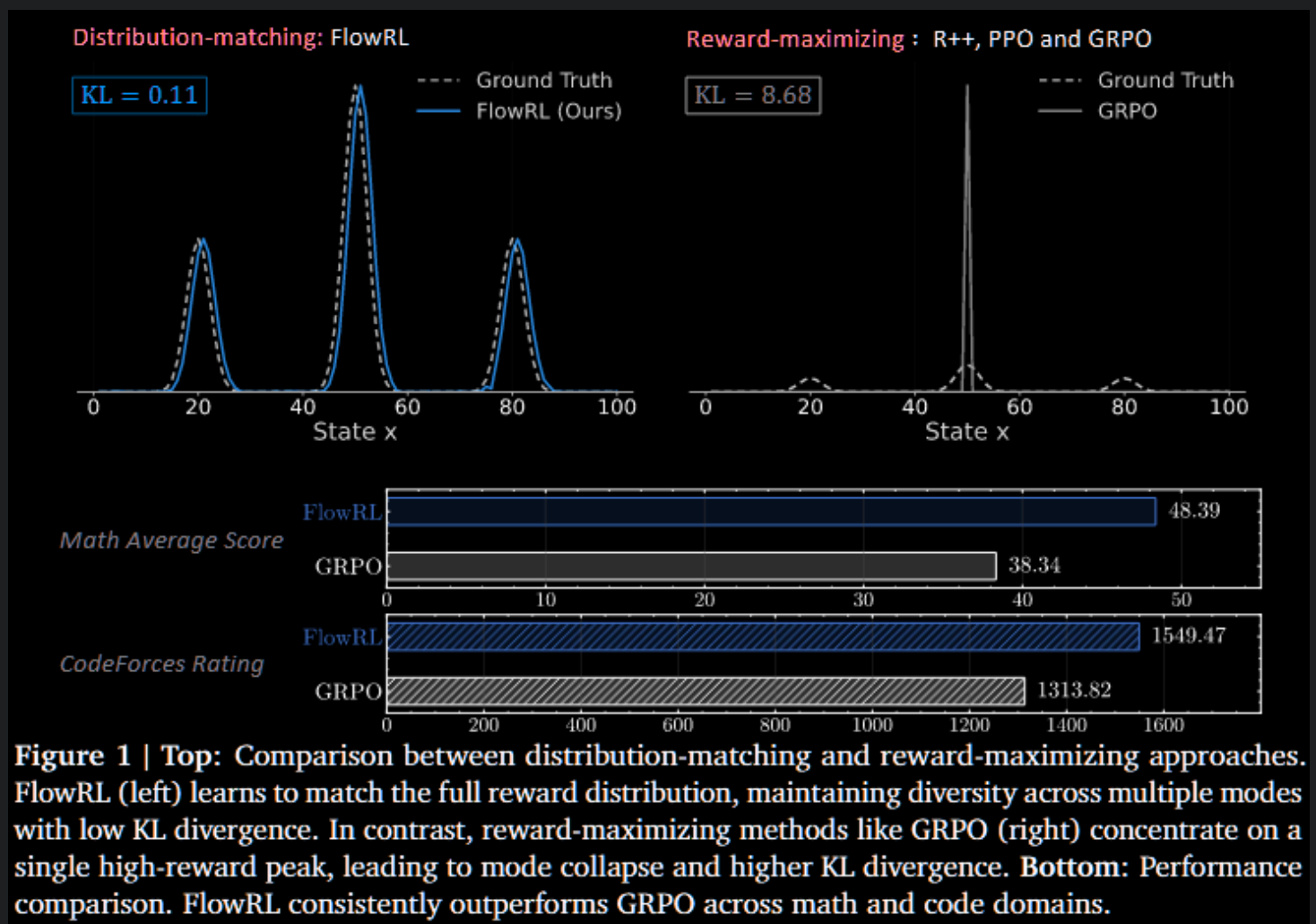



The key idea

Reinforcement learning (RL) has become the dominant paradigm for fine-tuning LLMs to give them 'reasoning capabilities'. Recent reasoning models adopt *reward-maximising* methods (eg, PPO, GRPO or one of the many variants). These can over-optimize dominant reward signals, potentially neglecting less frequent but equally valid reasoning paths, thus reducing diversity.

FlowRL uses the framework of GFlowNets (2023) to transform *scalar* rewards into a target *distribution*, using a learnable partition function, and minimises the (reverse) KL divergence between the policy and target distribution. More diverse generation is observed, with improved scores on certain benchmarks.



 Distribution matching vs reward maximisation

NB. It is worth pointing out that GFlowNets have been used in LLM fine-tuning in the past. Particularly, *Flow of Reasoning* (2025) formulates multi-step LLM reasoning as a Markovian flow on a directed acyclic graph (DAG).

Background

RL plays a crucial role in post-training of LLMs. Training algorithms have progressed through several key stages: from REINFORCE (1999), through TRPO (2015) and PPO (2017), to GRPO (2024) and its variants. These all aim to maximise reward signals, potentially leading to model collapse. One of the key aspects of TRPO was to introduce a KL penalty to prevent this collapse. More recent adaptations include adjusting the clip ratio (DAPO, 2025) or resetting the reference model (ProRL, 2025), all with the objective of increasing diversity.

The paper uses the machinery of *generative flow networks* (GFlowNets) introduced by Bengio et al (2023). The following brief description is taken from §2. GFlowNets are a probabilistic framework for training stochastic policies to sample discrete, compositional objects (eg, graphs or sequences) in proportion to a given reward. The core principle (see Figure 2) is to balance forward and backward probability flows at each state, inspired by flow matching (Bengio et al, 2021).



Figure 2 | GFlowNets [Bengio et al., 2023a], a flow-balance perspective on reinforcement learning. The initial flow $Z_\phi(s_0)$ injects probability mass into the environment, which is transported through intermediate states by the policy π_θ and accumulated at terminal states in proportion to the scalar rewards.

 GFlowNets

Their method

The paper introduces **FlowRL**, a policy optimisation algorithm designed to align the policy model with the *full reward distribution* encouraging mode coverage—rather than *reward maximisation* which tries to fine the 'best' local mode. The core idea is to introduce a learnable partition function that converts scalar rewards into a target distribution; the objective is to then minimise the (reverse) KL divergence between the policy and this.

This KL objective is based on the trajectory balance formulation from GFlowNets. Prior work on GFlowNets typically operated on short trajectories in small discrete spaces. To address challenges of long CoT (chain-of-thought) training, two key techniques are adjusted:

- *length normalisation* to tackle gradient explosion in variable-length CoT reasoning;
- *importance sampling* to correct for distribution mismatch between generated rollouts and the current policy.

Both these arise frequently in other LLM post-training contexts but need to be adjusted to the GFlowNets set-up.

Results


FlowRL is compared with 'vanilla' RL algorithms: REINFORCE++ ([2025/01](#)), PPO ([2017/07](#)) and GRPO ([2024/12](#)). The benchmarks are across maths and code domains, using both base and distilled LLMs of reasonable size: 32B and 7B, respectively. Consistent performance improvements are observed (see note below), as well as (perhaps, more importantly) increased diversity of generated reasoning paths—although, metrics on such are less precisely defined.

	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg
Qwen2.5-32B-Base, Max Response Len=8K							
Backbone	4.6	2.1	28.6	52.5	27.0	21.4	22.7
R++	14.8 _{+10.2}	9.2 _{+7.1}	52.7 _{+24.1}	44.4 _{-8.1}	17.4 _{-9.6}	24.5 _{+3.1}	27.1
PPO	26.9 _{+22.3}	20.4 _{+18.3}	76.4 _{+47.8}	69.2 _{+16.7}	28.8 _{+1.8}	37.9 _{+16.5}	43.3
GRPO	23.1 _{+18.5}	14.6 _{+12.5}	76.9 _{+48.3}	61.6 _{+9.1}	19.0 _{-8.0}	34.9 _{+13.5}	38.3
FlowRL	24.0 _{+19.4}	21.9 _{+19.8}	73.8 _{+45.2}	80.8 _{+28.3}	38.2 _{+11.2}	51.8 _{+30.4}	48.4
Qwen2.5-7B-Base, Max Response Len=8K							
Backbone	4.4	2.1	30.8	54.5	22.4	24.0	23.0
R++	11.0 _{+6.6}	5.4 _{+3.3}	66.7 _{+35.9}	54.3 _{-0.2}	24.4 _{+2.0}	27.3 _{+3.3}	31.5
PPO	9.4 _{+5.0}	7.3 _{+5.2}	63.4 _{+32.6}	58.0 _{+3.5}	26.5 _{+4.1}	27.3 _{+3.3}	32.0
GRPO	13.5 _{+9.1}	9.8 _{+7.7}	64.5 _{+33.7}	57.1 _{+2.6}	23.1 _{+0.7}	26.9 _{+2.9}	32.5
FlowRL	15.4 _{+11.0}	10.8 _{+8.7}	54.5 _{+23.7}	67.0 _{+12.5}	31.4 _{+9.0}	34.6 _{+10.6}	35.6

Table 1 | Results on math benchmarks. We report Avg@16 accuracy with relative improvements shown as subscripts. Positive gains are shown in **green** and negative changes in **red**. FlowRL outperforms all baselines across both 7B and 32B model scales.

Models	LiveCodeBench		CodeForces		HumanEval+
	Avg@16	Pass@16	Rating	Percentile	Avg@16
DeepSeek-R1-Distill-Qwen-7B, Max Response Len=8K					
Backbone	30.7	49.5	886.7	19.4	80.9
R++	30.5 _{-0.2}	52.7 _{+3.2}	1208.0 _{+321.3}	56.8 _{+37.4}	76.6 _{-4.3}
PPO	35.1 _{+4.4}	54.5 _{+5.0}	1403.1 _{+516.4}	73.7 _{+54.3}	82.3 _{+1.4}
GRPO	32.8 _{+2.1}	52.3 _{+2.8}	1313.8 _{+427.1}	67.1 _{+47.7}	80.1 _{-0.8}
FlowRL	37.4 _{+6.7}	56.3 _{+6.8}	1549.5 _{+662.8}	83.3 _{+63.9}	83.3 _{+2.4}

Table 2 | Results on code benchmarks. We report metrics with relative improvements shown as subscripts. Positive gains are shown in **green** and negative changes in **red**. FlowRL achieves the strongest performance across all three benchmarks, demonstrating its effectiveness in code reasoning tasks.

 Results on maths and code benchmarks, respectively

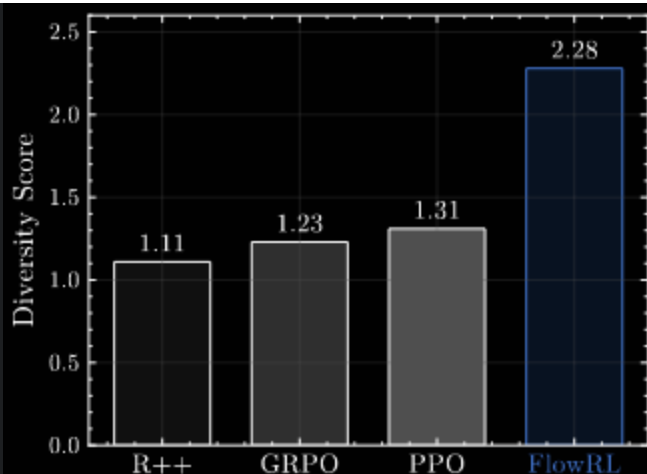



Figure 4 | GPT-judged diversity scores on roll-outs of AIME 24/25 problems. FlowRL generates more diverse solutions than R++, GRPO, and PPO.

 Diversity scores

NB. It must be pointed out that that none of the more modern variants, such as DAPO ([2025/03](#)), ProRL ([2025/05](#)) or GSPO/GMPO ([2025/07/2025/07](#)) are compared. Many of these *explicitly* address lack of diversity. Little can be inferred regarding, say, FlowRL vs DAPO or ProRL.

Takeaways

The paper introduces a new framework for LLM reasoning, and benchmarks it vs some modern, albeit vanilla, RL algorithms.

- *Positives.*
 - Mainstream LLM reasoning approaches are (highly?) unsatisfying, particularly lack of diversity.
 - The paper effectively proposes an alternative paradigm, using GFlowNets as a tool, and the diversity certainly improves, significantly in some cases.
- *Negatives.*
 - The lack of discussion on the other recent (earlier) paper using GFlowNets (FoR, [2024](#)) is disappointing from an academic standing.
 - The lack of comparison with state-of-the-art RL training algorithms prohibits any real conclusion on whether GFlowNets are a better way to go than reward-maximisation.
- *Conclusion.*
 - Despite its shortcomings, this reviewer finds the paper very interesting, and could definitely serve as a foundation for further research.
 - Replication studies, with better benchmarks, are highly encouraged.