Power-Flow: Unlocking LLMs with α-Power Distribution Fine-Tuning

Ruishuo Chen
2025211106
crs25@mails.tsinghua.edu.cn

Kefei Chen
2025211108
ckf25@mails.tsinghua.edu.cn

Abstract

Fine-tuning Large Language Models (LLMs) with Reinforcement Learning (RL) effectively enhances their capabilities but typically relies on costly external reward signals. While recent self-rewarding methods offer an alternative, they often use heuristic rewards with unclear learning objectives. We posit that many advanced skills, such as reasoning and creativity, are already latent within the base model and can be activated by sampling from its power distribution, $p_{\text{base}}(x)^\alpha$. However, existing sampling methods like MCMC are inefficient at inference time. We propose a novel unsupervised fine-tuning framework using Generative Flow Networks (GFlowNets) to directly train a policy that samples from this target α-power trajectory distribution. We define an intrinsic reward signal based on the trajectory density of the base model, calculated using the frozen base model itself. This principled approach provides a unified mechanism to controllably unlock latent abilities: setting $\alpha > 1$ enhances reasoning by "sharpening" the distribution, while $\alpha < 1$ unlocks creative diversity by "flattening" it. We plan to demonstrate the effectiveness of our method on reasoning and creative generation benchmarks.

1 Introduction

Reinforcement Learning (RL) has proven to be a highly effective technique in the post-training of Large Language Models (LLMs). This paradigm has been pivotal for aligning models with human preferences [1], ensuring safety [2], and incentivizing reasoning capabilities [3].

However, acquiring high-quality, unbiased external reward signals for RL is notoriously difficult and expensive [4, 5]. Furthermore, a growing body of research argues that RL fine-tuning primarily *activates* latent pathways already present in the base model, rather than instilling entirely new skills [6]. Therefore, significant research efforts have begun to explore how to activate these latent abilities without relying on external reward signals.

While promising, these current unsupervised methods mostly rely on heuristics or statistical measures derived from intuition [7–9], leaving their precise learning objectives unclear. Recently, work has shown that sampling from a "sharpened" distribution via MCMC can enhance LLM reasoning capability [10]. Concurrently, other work claims that model creativity is "locked" in low-probability regions due to human bias towards typical answers during alignment [11].

Motivated by these findings, we propose a principled framework to directly fine-tune the LLM to sample from its α-power distribution, $p_{\text{base}}(x)^\alpha$. Our approach provides a clear and unified mechanism to controllably enhance these latent abilities, without incurring the substantial computational overhead of MCMC methods at inference time. To achieve this distributional matching goal, we employ Generative Flow Networks (GFlowNets) [12, 13] during fine-tuning, using the hyperparameter α to modulate the preference for specific model capabilities. We will test the effectiveness of our method

Preprint. AML course project.

on reasoning and creativity tasks, comparing it against both externally-rewarded RL methods and unsupervised fine-tuning or activation approaches.

2 Related Work

2.1 Unsupervised and Self-Rewarding Fine-Tuning

A growing body of work aims to fine-tune LLMs without external reward signals, often by defining heuristic-based intrinsic rewards. For example, [14] proposed using an LLM-as-a-Judge prompt structure to generate its own rewards during training. Similarly, TTRL [15] constructs a consistency reward via majority voting over model-generated outputs for RL fine-tuning, and Self-Rewarding PPO [9] uses a "coherence reward" defined as the log-policy ratio between the SFT and base models.

In particular, many of these approaches focus on the statistical property of entropy, albeit with conflicting goals. On one hand, ETPO [7] augments the RL objective with an entropy bonus to promote exploration and diversity. Conversely, a contrary line of work argues for minimizing entropy to improve reasoning and consistency. For instance, RENT [16] demonstrates that optimizing for model confidence, which is framed as minimizing token-level entropy, can enhance reasoning. Along this line, EMPO [8] also aims to reduce semantic entropy to incentivize coherent reasoning.

While effective, these methods often optimize for proxy objectives, whereas our work aims to learn a well-defined target distribution.

2.2 Inference-Time Capability Activation

Another line of research directly motivates our target distribution: the activation of latent capabilities during inference. For instance, [10] provided key evidence that high-quality reasoning is latent within base models and can be surfaced by using MCMC to sample from a sharpened distribution. However, their specialized sampling approach incurred an excessive 8.84-fold increase in inference cost, severely limiting its practical applicability. Complementary work by [11] demonstrated that creativity—often suppressed in aligned models due to human preference for typical, low-reward answers—could be unlocked merely by adjusting the prompt. This hints at the possibility of efficiently releasing the model's creative potential from low-density regions of its base distribution. Building on these insights, we propose to fine-tune the model to directly sample from an α-power distribution, leveraging the hyperparameter α to effectively activate different, latent capabilities of the base model.

3 Proposed Method

Our goal is to fine-tune an LLM policy, π, to sample from the α-power distribution of a base model, $p_{\text{base}}(x)^{\alpha}$. We employ Generative Flow Networks to achieve this distributional matching.

3.1 Preliminaries: Generative Flow Networks

Generative Flow Networks (GFlowNets) [12, 13] are a family of generative models designed to learn a policy $P_{F,\theta}(x)$ that samples objects x with a probability proportional to a given non-negative reward function $R(x)$, such that $P_{F,\theta}(x) \propto R(x)$.

A common objective used to train GFlowNets is Trajectory Balance (TB) [17]. The TB objective enforces a flow consistency constraint over a complete generation trajectory τ (which terminates in x) and is typically formulated in log-space, optimizing a set of parameters θ:

$$\mathcal{L}_{TB}(\tau;\theta) = \left(\log Z_\theta + \log P_{F,\theta}(\tau) - \log R(x) - \log P_{B,\theta}(\tau)\right)^2$$

Here, Z_θ is a learnable parameter estimating the partition function, $P_{F,\theta}(\tau)$ is the probability of the forward trajectory (sampling x), and $P_{B,\theta}(\tau)$ is the probability of the backward trajectory (deconstructing x). Minimizing this loss drives $P_{F,\theta}(x)$ to converge to the target distribution $R(x)/Z_\theta$.

3.2 GFlowNet Objective for LLM Fine-tuning

We model autoregressive text generation as a sequential decision making process in a directed acyclic graph (DAG). Given an initial prompt q, which serves as the root state s_0, a state s_t corresponds to a

partial token sequence $y_{<t} = (y_1, \ldots, y_{t-1})$. An action is the selection of the next token y_t from the vocabulary $\mathcal{V}$. The process continues until a terminal token (e.g., `[EOS]`) is sampled, resulting in a complete trajectory τ and a terminal sequence $y = (y_1, \ldots, y_T)$.

Our goal is to learn a policy π_θ that samples from the α-power distribution $p_{\text{base}}(y|q)^\alpha$. Inspired by FlowRL [18], which adapts the TB loss for LLMs, we propose a modified objective for our *unsupervised* setting. We replace the external reward term in the original FlowRL objective with our target log-density, $\log p_{\text{base}}(y|q)^\alpha$, normalized by sequence length. This yields our final loss function:[1]

$$\mathcal{L} = w \cdot \left(\log Z_\theta(q) + \frac{1}{|y|} \log \pi_\theta(y|q) - \frac{\alpha}{|y|} \log p_{\text{base}}(y|q) \right)^2$$

where $\pi_\theta(y|q)$ is the policy we are fine-tuning ($P_{F,\theta}$), $p_{\text{base}}(y|q)$ is the frozen base model, $Z_\theta(x)$ is a learnable, context-dependent partition function, and w is a PPO-style clipped importance weight used in [18]. This objective directly optimizes π_θ to match the target α-power distribution, $p_{\text{base}}(y|q)^\alpha$.

4 Experimental Setup

4.1 Models and Baselines

We will fine-tune open-source models, including **Qwen2.5-Math-7B** [19] and **Llama-3.1-8B-Instruct** (to ensure a convenient comparison with our baselines), on unlabeled mathematical and general prompts. We will compare our method against:

- **Base Model:** The original, unfine-tuned model.
- **GRPO (Supervised):** A leading RL method using golden answer rewards [20].
- **EMPO (Unsupervised):** The state-of-the-art unsupervised reasoning method [8].
- **MCMC Sampling (Inference-Time):** The method from [10].
- **Verbalized Sampling (Inference-Time):** The prompting method for diversity [11].

4.2 Task 1: Reasoning Enhancement ($\alpha > 1$)

Benchmarks: We will evaluate on standard reasoning benchmarks used by our baselines, including **MATH** (MATH500 subset) [21], **GPQA** [22], and **MMLU-Pro** [23].

Metrics: We will measure **Pass@1 Accuracy** (greedy decoding) against all baselines. We will also plot **Pass@k** curves to test if our fine-tuning "sharpens" the base model's latent knowledge.

4.3 Task 2: Diversity and Creativity ($\alpha < 1$)

Benchmarks: Following [11], we will use **Creative Writing** [24] and **Open-Ended QA** [25].

Metrics: We will measure **Semantic/Lexical Diversity** and **Quality** (using LLM-as-a-judge). For QA, we will also compute **KL Divergence** against the pre-training corpus distribution to measure mode collapse mitigation.

References

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[2] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon *et al.*, "Constitutional ai: Harmlessness from ai feedback," *arXiv preprint arXiv:2212.08073*, 2022.

[3] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[1] Practical implementation may require techniques to address potential numerical and gradient instabilities.

[4] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," *arXiv preprint arXiv:1909.08593*, 2019.

[5] L. Gao, J. Schulman, and J. Hilton, "Scaling laws for reward model overoptimization," in *International Conference on Machine Learning*. PMLR, 2023, pp. 10 835–10 866.

[6] Y. Yue, Z. Chen, R. Lu, A. Zhao, Z. Wang, S. Song, and G. Huang, "Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?" *arXiv preprint arXiv:2504.13837*, 2025.

[7] M. Wen, J. Liao, C. Deng, J. Wang, W. Zhang, and Y. Wen, "Entropy-regularized token-level policy optimization for language agent reinforcement," *arXiv preprint arXiv:2402.06700*, 2024.

[8] Q. Zhang, H. Wu, C. Zhang, P. Zhao, and Y. Bian, "Right question is already half the answer: Fully unsupervised llm reasoning incentivization," *arXiv preprint arXiv:2504.05812*, 2025.

[9] Q. Zhang, L. Qiu, I. Hong, Z. Xu, T. Liu, S. Li, R. Zhang, Z. Li, L. Li, B. Yin *et al.*, "Self-rewarding ppo: Aligning large language models with demonstrations only," *arXiv preprint arXiv:2510.21090*, 2025.

[10] A. Karan and Y. Du, "Reasoning with sampling: Your base model is smarter than you think," *arXiv preprint arXiv:2510.14901*, 2025.

[11] J. Zhang, S. Yu, D. Chong, A. Sicilia, M. R. Tomz, C. D. Manning, and W. Shi, "Verbalized sampling: How to mitigate mode collapse and unlock llm diversity," *arXiv preprint arXiv:2510.01171*, 2025.

[12] E. Bengio, M. Jain, M. Korablyov, D. Precup, and Y. Bengio, "Flow network based generative models for non-iterative diverse candidate generation," *Advances in neural information processing systems*, vol. 34, pp. 27 381–27 394, 2021.

[13] Y. Bengio, S. Lahlou, T. Deleu, E. J. Hu, M. Tiwari, and E. Bengio, "Gflownet foundations," *Journal of Machine Learning Research*, vol. 24, no. 210, pp. 1–55, 2023.

[14] W. Yuan, R. Y. Pang, K. Cho, X. Li, S. Sukhbaatar, J. Xu, and J. E. Weston, "Self-rewarding language models," in *Forty-first International Conference on Machine Learning*, 2024.

[15] Y. Zuo, K. Zhang, L. Sheng, S. Qu, G. Cui, X. Zhu, H. Li, Y. Zhang, X. Long, E. Hua *et al.*, "Ttrl: Test-time reinforcement learning," *arXiv preprint arXiv:2504.16084*, 2025.

[16] M. Prabhudesai, L. Chen, A. Ippoliti, K. Fragkiadaki, H. Liu, and D. Pathak, "Maximizing confidence alone improves reasoning," *arXiv preprint arXiv:2505.22660*, 2025.

[17] N. Malkin, M. Jain, E. Bengio, C. Sun, and Y. Bengio, "Trajectory balance: Improved credit assignment in gflownets," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5955–5967, 2022.

[18] X. Zhu, D. Cheng, D. Zhang, H. Li, K. Zhang, C. Jiang, Y. Sun, E. Hua, Y. Zuo, X. Lv *et al.*, "Flowrl: Matching reward distributions for llm reasoning," *arXiv preprint arXiv:2509.15207*, 2025.

[19] A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, K. Lu, M. Xue, R. Lin, T. Liu, X. Ren, and Z. Zhang, "Qwen2.5-math technical report: Toward mathematical expert model via self-improvement," *ArXiv*, vol. abs/2409.12122, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272707652

[20] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu *et al.*, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[21] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe, "Let's verify step by step," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=v8L0pN6EOi

[22] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman, "GPQA: A graduate-level google-proof q&a benchmark," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=Ti67584b98

[23] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, T. Li, M. W. Ku, K. Wang, A. Zhuang, R. R. Fan, X. Yue, and W. Chen, "Mmlu-pro: A more robust and challenging multi-task language understanding benchmark," *ArXiv*, vol. abs/2406.01574, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:270210486

[24] S. J. Paech, "Eq-bench: An emotional intelligence benchmark for large language models," *ArXiv*, vol. abs/2312.06281, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266163145

[25] J. Wong, Y. Orlovskiy, M. Luo, S. A. Seshia, and J. Gonzalez, "Simplestrat: Diversifying language model generation with stratification," *ArXiv*, vol. abs/2410.09038, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:273323783