

RLFR: EXTENDING REINFORCEMENT LEARNING FOR LLMs WITH FLOW ENVIRONMENT

Anonymous authors

Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has recently emerged as a promising framework for improving reasoning abilities in Large Language Models (LLMs). However, policy optimized with binary verification prone to overlook potential valuable exploration in reasoning trajectory. In view of heavy annotation cost of golden Process Reward Models (PRMs), recent works attempt using auxiliary signals for reward shaping of process tokens, involving entropy and likelihood collected from logit space. In this work, we offer a novel perspective on shaping RLVR with flow rewards derived from latent space, and propose **RLFR**, where the flow fields of model latents are constructed from either off-policy high-quality data and on-policy rejection sampling data, and the velocity deviations of policy latents within it are quantified to serve as a reward signal. **RLFR** first demonstrates that a well-established flow field can be a sound environment for reward signal collection, highlighting the expressive latent space is much underexplored. Moreover, **RLFR** is able to compress any off-policy expert data as reference for constituting reward signals, and we show that the efficient context dependence compressed within the hidden states are utilized, rather than individual token-level denotation for context comprehending. Experiments on both language and multimodal reasoning benchmarks demonstrate the reliability of flow rewards, and suggesting a promising paradigm for reward shaping with auxiliary signals.

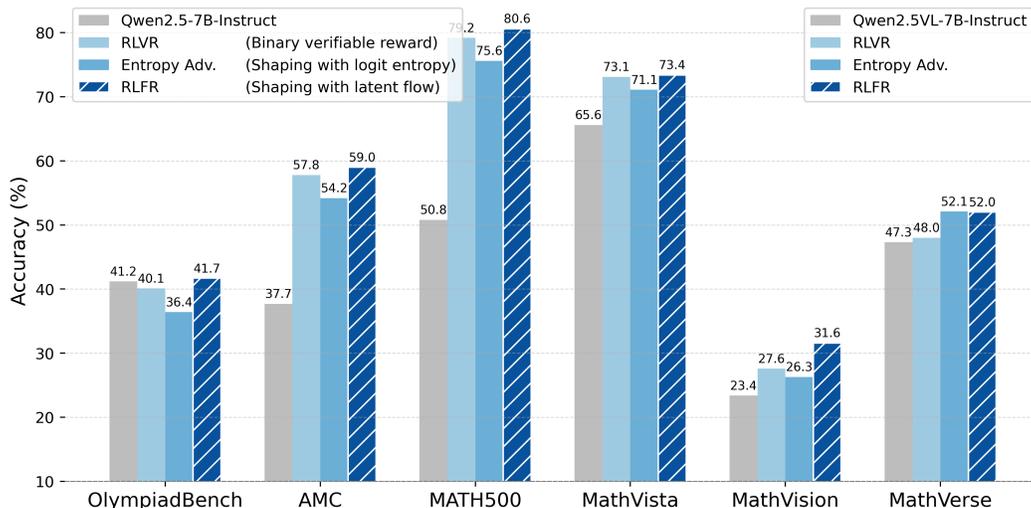


Figure 1: Overall performance on language (left) and multimodal (right) reasoning benchmarks. By introducing flow reward from latent space, RLFR shows consistent progress over RLVR with binary verification and entropy based shaping method (Cheng et al., 2025) from logit space, highlighting the expressive latent space is much underexplored for reward signal collection.

1 INTRODUCTION

Recent advances in improving reasoning abilities of Large Language Models (LLMs) underscore the substantial promise of Reinforcement Learning with Verifiable Rewards (RLVR) (Lambert et al.,

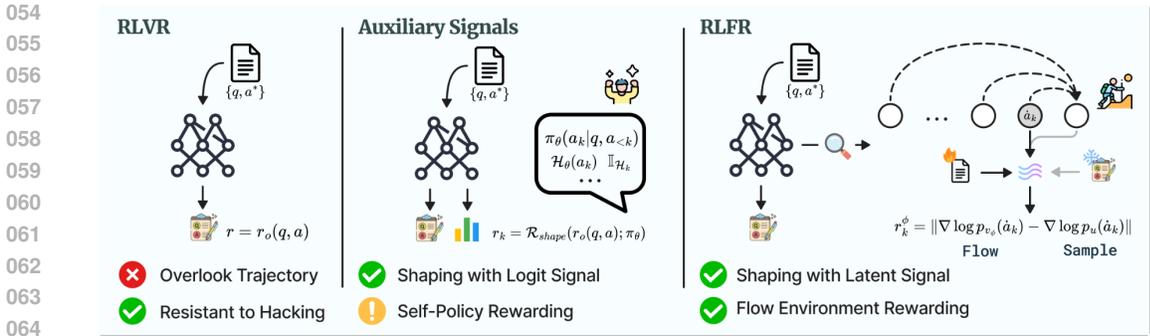


Figure 2: Policy optimized with RLVR prone to overlook potential valuable explorations in reasoning trajectories. To beyond binary verification, auxiliary signals are used for reward shaping of process tokens, involving token entropy and likelihood collected from logit space, where self-policy rewarding risks are non-negligible. Alternatively, we show that the latent space is much underexplored yet highly expressive and a well established flow field can be a sound environment for yielding flow reward from velocity deviations and extending RLVR with latent reward utilization.

2024; Jaech et al., 2024; DeepSeek-AI et al., 2025). By incentivizing the optimization of LLMs with outcome verification, there are far less susceptible to reward hacking. However, the binary verification prone to overlook potential valuable policy explorations in reasoning trajectories in cases that the answers are difficult to derive with part of correct trajectory (Hammoud et al., 2025), thus provides an intolerant reward signal with decreased exploratory behavior (Cui et al., 2025b).

A more natural way to address these issues is to provide step-by-step process rewards along reasoning trajectories with Process Reward Models (PRMs) (Zhao et al., 2025; Liu et al., 2025d; Yang et al., 2025a), however, the heavy annotation costs of intermediate steps pose a significant bottleneck for scalability at time, and the misalignment between PRMs training corpora and online reasoning trajectories further introduce reward gaps (Ye et al., 2025). Alternatively, the value model in PPO framework (Schulman et al., 2017; Yue et al., 2025; Yuan et al., 2025) offers a promising strategy, but the effective credit assignment over the binary outcome reward are still underexplored.

To encouraging policy exploration, recent works leverage auxiliary signals for reward shaping (Ng et al., 1999) beyond binary outcome verification, involving model likelihood (Damani et al., 2025; Li et al., 2025b; He et al., 2025) and token entropy (Cheng et al., 2025; Wang et al., 2025a) collected from logit space. While the confidence may serve as an appropriate indicator for examining policy states, it may not be well-suited for constituting reward signals for optimization (Wang et al., 2025b; Cui et al., 2025b). As the self-policy rewarding may cause LLM over exploits its own confidence estimates rather than learning genuinely improved reasoning strategies, where the potential hacking risks are non-negligible, and may undermine prolong RL training (Liu et al., 2025b;c).

In this work, we propose **RLFR**, that offering a novel framework on shaping RLVR with flow rewards for regarding reasoning trajectory. We aspire to explore *whether the broader latent space of LLMs encompass productive signals for reward utilization with reliable stability*. RLFR first constructs flow fields (Lipman et al., 2022; Liu et al., 2022) of model latents from either off-policy high-quality data and on-policy rejection sampling data, and the velocity deviations of policy latents within it are quantified to serve as a reward signal. While the larger deviations are penalized as drifting away from the reference distribution formed by flow, and smaller deviations are encouraged. The flow fields are online updated alongside the policy optimization, with rejection sampling data filtered by desired metrics. We also formally show that the evidence lower bound of log-likelihood is constituted by negative velocity deviation, thus establishing the connection between velocity deviation and probability likelihood with inverse correlation under reference distribution.

Particularly, RLFR first demonstrates that a well-established flow field can be a sound environment for reward signal collection, yielding stable performance gains throughout RL training with no sign of degeneration. And we also highlight that the expressive latent space are highly underexplored as a substrate for reward design, complementing prior auxiliary signals from logit space. Moreover, RLFR provides a natural way to leverage expert reasoning trajectories from off-policy data into the constitution of reward signals through reference flow fields. Experiments on both language and

multimodal reasoning benchmarks across Qwen and Llama models demonstrate the reliability of flow rewards with consistent progress over baseline RLVR methods.

The contributions of this work can be summarized as follows: (1) We propose RLFR, a promising reward shaping framework with flow rewards derived from LLMs latents, extending RLVR with latent rewards utilization. (2) We demonstrate that a well-established flow field can be a sound environment for reward signal collection, highlighting the expressive latent space is much underexplored. (3) Both off-policy expert data and on-policy rejection sampling data are introduced for constituting reward signals as flow reference. (4) Comprehensive experiments on diverse reasoning benchmarks across both language and multimodal models demonstrate the reliability of our framework. All the codes, data, and model weights are released to foster future research in this area.

2 PRELIMINARIES

2.1 REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS

Reinforcement Learning with Verifiable Rewards (RLVR) (Lambert et al., 2024) constitutes a general post-training paradigm where model response can be deterministically verified. Let π_θ be an LLM parameterized by θ , that receives a prompt q and generates a token sequence $\mathbf{a} = (a_1, \dots, a_K)$ as response. A binary verifier then assigns a scalar reward $r_o(q, \mathbf{a}) \in \{0, 1\}$ to each prompt-response pair, where r_o underlines its outcome nature. The goal of RL is to maximize the expected reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{q \sim \mathcal{D}, \mathbf{a} \sim \pi_\theta(\cdot|q)} [r_o(q, \mathbf{a})]. \quad (1)$$

Here, \mathcal{D} is a dataset of prompts with corresponding ground-truth answer. Despite robustness against reward hacking, the coarse granularity outcome rewards make RLVR prone to overlooking the potential valuable policy exploration in reasoning trajectories in cases of derived incorrect answers.

RLVR Algorithms. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as a widely used reinforcement learning algorithm that simplifies the Proximal Policy Optimization (PPO) (Schulman et al., 2017) by discarding the value model for baseline advantage estimation. Sampling a group of response $\{\mathbf{a}_i\}_{i=1}^G$ per prompt and using their average reward as baseline, and the clipped surrogate objective is preserved as PPO, leading to the following maximization objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{D}, \{\mathbf{a}_i\}_{i=1}^G \sim \pi_\theta(\cdot|q)} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|a_i|} \sum_{t=1}^{|a_i|} \min \left(\rho_{i,k} \hat{A}_i, \text{clip}(\rho_{i,k}, 1 - \varepsilon, 1 + \varepsilon) \hat{A}_i \right) \right], \quad (2)$$

where $\rho_{i,k} = \frac{\pi_\theta(a_{i,k}|q, a_{i,<k})}{\pi_{\theta_{\text{old}}}(a_{i,k}|q, a_{i,<k})}$ is the importance sampling ratio between the current and old policy models, and the advantage $\hat{A}_{i,o}$ is shared among all tokens within response \mathbf{a}_i and is computed as

$$\hat{A}_{i,o} = \frac{r_{i,o} - \text{mean}(\{r_{i,o}\}_{i=1}^G)}{\text{std}(\{r_{i,o}\}_{i=1}^G)}. \quad (3)$$

We denote $r_{i,o} = r_o(q, \mathbf{a}_i)$ for simplicity and clear comparison with later introduced dense rewards.

Reward Shaping. As a common technique in reinforcement learning for accelerating and stabilizing policy optimization, reward shaping transforms explicit environment-based rewards into a proxy reward function, which typically involves operations like clipping or shifting, *etc.*, (Wang et al., 2024c), and may also incorporate auxiliary signals, such as response length or token entropy to steer the model toward desired behavior more effectively (Arora & Zanette, 2025; Cheng et al., 2025).

2.2 FLOW MATCHING

Flow Matching (FM) (Lipman et al., 2022; Liu et al., 2022) defines a generative process that learns a continuous-time velocity field, transporting samples from a simple prior distribution p_{init} (e.g., Gaussian) into the target data distribution p_{data} . With data pairs $(\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1)$ sampled from $p_{\text{init}} \times p_{\text{data}}$, the forward process is given by the linear interpolation: $\hat{\mathbf{x}}_t = (1-t)\hat{\mathbf{x}}_0 + t\hat{\mathbf{x}}_1$, and the neural network \mathbf{v}_ϕ is trained to predict the target velocity field $\mathbf{u} = \hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_0$ by minimizing the flow matching loss:

$$\mathcal{L}_{\text{FM}}(\hat{\mathbf{x}}_t; \phi) = \mathbb{E}_{t \sim \mathcal{U}[0,1], \hat{\mathbf{x}}_0 \sim p_{\text{init}}, \hat{\mathbf{x}}_1 \sim p_{\text{data}}} [\|\mathbf{v}_\phi(\hat{\mathbf{x}}_t, t) - (\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_0)\|^2]. \quad (4)$$

We denote $\hat{\mathbf{x}}$ as latent signal and distinguish it from tokens. The \mathbf{v}_ϕ characterizes the p_{data} with the flow field through accurate velocity prediction, where the outsider can be notably identified.

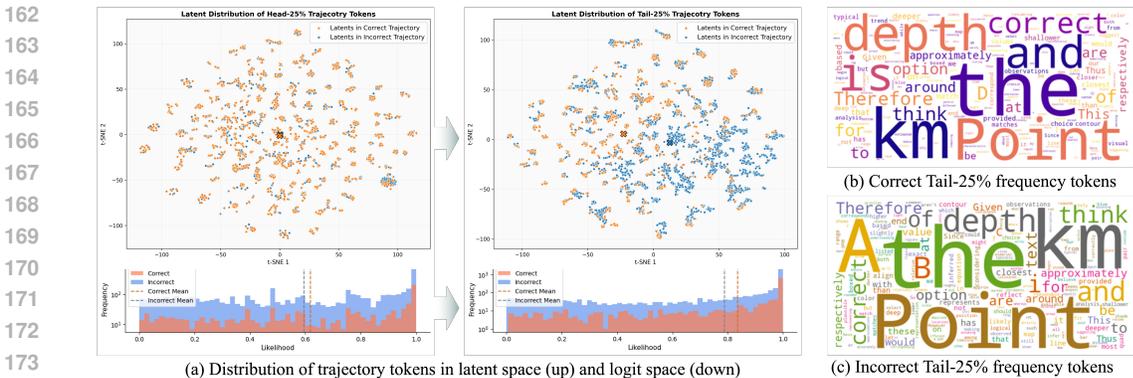


Figure 3: Distribution of trajectory tokens in LLM reasoning. **(a) Distribution of trajectory tokens in latent space (up) and logit space (down).** We perform 256 rollouts for prompt randomly sampled from MATH (Hendrycks et al., 2021). The latent distribution show progressively expressive signals on tail trajectory tokens, as continuously interacting with preceding tokens for context compression. In contrast, neither the logit distribution nor the **(b) & (c) textual clouds of reasoning trajectories** reveal any distinguishable signals, highlighting the potential of latent space for reward utilization.

3 RLFR

In this section, we first show that the latent space is highly expressive for reward utilization, and then introduce the flow reward formed by velocity deviation, where the connection with probability likelihood are established. We then introduce RLFR, that extends RLVR with flow rewards for process tokens in advantage estimation, while the flow conditions are involved to introduce context dependency. The flow fields are online updated with rejection sampling data during policy optimization.

3.1 ANALYZING LATENT SPACE IN REASONING TRAJECTORY

Previous studies have shown the effectiveness of signals from logit space in shaping binary verification reward (Damani et al., 2025; Cheng et al., 2025; Wang et al., 2025b; Yu et al., 2025b), while the broader latent space is much underexplored. In light of this, we further analyze the latents of reasoning trajectory tokens to evaluate the reliability of latent space for reward utilization. We use Qwen2.5-7B (Yang et al., 2024) to generate 256 rollouts for prompt randomly sampled from MATH (Hendrycks et al., 2021) with 0.7 temperature for decoding, and ultimately get 10^5 response tokens within the same subject content along with their corresponding latents. The latents are extracted from layers $\mathcal{L} = (\tau_0 L, \dots, \tau_N L)$ for probing, where L denotes the total number of model layers, and $\tau_i \in [0, 1]$ specifies the percentile positions for extraction. We show the case of $\tau_i = 0.5$ in Figure 3, while additional results are provided in Appendix B and show the similar tendency.

The latents of tail tokens in reasoning trajectory show progressively pronounced signals when distinguished by answer correction. We observe that compared to head tokens in reasoning trajectory where basically no identifiable signals are emerged, the latents of tail tokens are much more expressive as continuously interacting with all preceding tokens for context compression. However, neither the logit space likelihood nor the textual clouds exhibit any noticeable signals.

Subset of latents in reasoning trajectory with incorrect answer exhibit close to those in correct trajectories, while RLVR with outcome verification penalizes them. We discern that penalization should be applied to latents with substantial deviations from high-quality trajectory latents, while similar latents with minor deviations should be treated tolerantly.

3.2 FLOW REWARD FROM VELOCITY DEVIATION

Encouraged by the expressive latent space, a sound yet efficient metric is required to collect emerging signals for reward utilization, and we consider flow matching as underlying framework for which typically used in continuous modeling with established velocity field. Inspired by (Li et al., 2025c), we notice that instead of using predicted velocity to reverse the forward process for data distribution

generation, the accuracy of velocity prediction can serve as a sensible metric to evaluate whether current samples are within the data distribution formed by flow. And we rewrite the Eq. 4 as

$$\mathcal{R}_{\text{FM}}^\phi(\hat{\mathbf{a}}_k; t, \tau) = \|\mathbf{v}_\phi(\hat{\mathbf{a}}_{k,t}, t) - (\hat{\mathbf{a}}_{k,1} - \epsilon)\|^2, \quad \text{where } t, \tau \sim \mathcal{U}[0, 1], \epsilon \sim \mathcal{N}(0, I), \quad (5)$$

where $\hat{\mathbf{a}}_k$ denotes latents of token \mathbf{a}_k , and $\hat{\mathbf{a}}_{k,t}$ is the linear interpolation between $\hat{\mathbf{a}}_k$ and ϵ . The flow network \mathbf{v}_ϕ is first pre-trained on latents of high-quality data to establish the reference distribution for offline start, and then frozen for flow reward calculation that evaluates the velocity deviation of the current sample $\hat{\mathbf{a}}_k$ under the reference flow field.

Debiasing the Timestep Impacting. The flow network \mathbf{v}_ϕ provides velocity drifts toward reference distribution through the whole reverse process, which underscores the challenge on timestep priority in deviation evaluation for flow reward. Considering the connection between velocity prediction and score function (Gao et al., 2024; Liu et al., 2025a), we present the score given by \mathbf{v}_ϕ as:

$$\nabla_{\hat{\mathbf{a}}_{k,t}} \log p_{\mathbf{v}_\phi}(\hat{\mathbf{a}}_{k,t}) = -\frac{\hat{\mathbf{a}}_{k,t}}{1-t} + \frac{t}{1-t} \mathbf{v}_\phi(\hat{\mathbf{a}}_{k,t}, t). \quad (6)$$

A detailed proof is provided in Appendix C.1. Instead of relying on global-consistent direction provided by velocity prediction, the score function provides more accurate drift direction from local distributional gradients, which is more practical for deviation evaluation, and we have

$$\begin{aligned} \mathcal{R}_{CFM}^\phi(\hat{\mathbf{a}}_k; \mathcal{T}, \mathcal{L}) &= \mathbb{E}_{t \sim \mathcal{T}, \tau \sim \mathcal{L}} [\|\nabla_{\hat{\mathbf{a}}_{k,t}} \log p_{\mathbf{v}_\phi}(\hat{\mathbf{a}}_{k,t}) - \nabla_{\hat{\mathbf{a}}_{k,t}} \log p_{\mathbf{u}}(\hat{\mathbf{a}}_{k,t})\|^2] \\ &= \mathbb{E}_{t \sim \mathcal{T}, \tau \sim \mathcal{L}} \left[\frac{t}{1-t} \mathcal{R}_{FM}^\phi(\hat{\mathbf{a}}_k; t, \tau) \right], \end{aligned} \quad (7)$$

where \mathcal{R}_{CFM}^ϕ is used for calculating flow reward with timestep collection \mathcal{T} by debiasing weighting. We suggest that the velocity deviation serves as a surrogate for score deviation, while the coefficient emphasizes the timestep priority through the reverse process. We provide ablations in Sec. 4.3 that using flow reward at different timesteps for RL, where the larger timesteps with less noises are favorable in constituting reward signal, which is consistent with suggestion given by Eq. 7.

Theoretical Analysis with Likelihood. Comparing to probability likelihood under reference distribution, velocity deviations show directional drifts error toward reference distribution. These two paradigms appear as different profile for distribution evaluation, and we further clarify their underlying relationship as:

$$\log p_{\mathbf{v}_\phi}(\hat{\mathbf{a}}_k) \geq C(\hat{\mathbf{a}}_k) - \lambda \mathbb{E}_{t, \tau \sim \mathcal{U}[0,1]} [\mathcal{R}_{\text{FM}}^\phi(\hat{\mathbf{a}}_k; t, \tau)], \quad (8)$$

where $\log p_{\mathbf{v}_\phi}(\hat{\mathbf{a}}_k)$ is the log-likelihood under distribution parameterized by \mathbf{v}_ϕ , $\lambda > 0$ is constant, and $C(\hat{\mathbf{a}}_k)$ is the sundry term. We show that the evidence lower bound of log-likelihood is constituted by negative velocity deviation under reference distribution, indicated that the two paradigms are inversely correlated, and the minimal $\mathcal{R}_{\text{FM}}^\phi(\hat{\mathbf{a}}_k; t, \tau)$ with respect to a given $\hat{\mathbf{a}}_k$ corresponds to the maximal evidence lower bound (ELBO) on the log-likelihood $\log p_{\mathbf{v}_\phi}(\hat{\mathbf{a}}_k)$. We calibrate the sign of flow rewards in RLFR, and provide the detailed proof of Eq. 8 in Appendix C.2.

3.3 EXTENDING RLVR WITH FLOW REWARD

Velocity-Based Advantage Shaping. The main idea behind RLFR is to leverage the expressive latent space of LLMs with flow reward and thus extend RLVR with latent rewards utilization. Instead of sharing common advantage within response \mathbf{a} , we shape advantage term for each token \mathbf{a}_k with flow returns, yielding by the accumulation of decayed flow rewards. While the advantage shaping makes it more flexible for different RLVR algorithms, without considering the specific advantage estimation methods. We have:

$$\hat{A}_k = \sum_{s=k}^{|\mathbf{a}|} \gamma^{s-k} r_s^{\mathbf{v}_\phi} + \hat{A}_o, \quad (9)$$

$$r_k^{\mathbf{v}_\phi} = -\beta \cdot \hat{r}_k^{\mathbf{v}_\phi} \mathbb{I}[\|\hat{r}_k^{\mathbf{v}_\phi}\| > \eta], \quad \text{where } \hat{r}_k^{\mathbf{v}_\phi} = \minmax(\{\mathcal{R}_{CFM}^\phi(\hat{\mathbf{a}}_k); \mathcal{T}, \mathcal{L}\}_{k=1}^{|\mathbf{a}|}), \quad (10)$$

where $\mathbb{I}[\cdot]$ is the indicator function and return 1 if condition is true and 0 otherwise, where we discard noisy fluctuations in flow rewards and preserve only substantial deviations above η . We perform the

Algorithm 1: Reinforcement Learning with Flow Rewards (RLFR)

Inputs: Online data $\mathcal{Q}\{q, a\}$, offline data $\mathcal{D}_{off}\{q, z, a\}$, reference data buffer \mathcal{B} , initial flow model v_ϕ , validated batch size κ , layer collection $\mathcal{L} = (l_1, \dots, l_N)$, response quality metric(\cdot)

Offline Start:
 Extract policy latents from layer collection \mathcal{L} on \mathcal{D}_{off} and construct the \mathcal{D}_{off}^*
 Perform flow training on \mathcal{D}_{off}^* with Eq. 4 where the loss is only calculated on response tokens

Online Optimization:
 Initialize reference data buffer $\mathcal{B} \leftarrow \emptyset$
while Training **do**
 Generate rollouts \mathcal{G} for batch data from \mathcal{Q}
 Optimize the policy with RL algorithms such as Eq. 2 with advantage estimated for each token by Eq. 9
 $\mathcal{B} \leftarrow \text{Rejection-Sampling}(\mathcal{G}, \text{metric})$; // Recommend metrics: correctness, entropy
 while $|\mathcal{B}| > \kappa$ **do**
 Optimize flow v_ϕ on batch data from \mathcal{B} with policy latents in \mathcal{L} using Eq. 4
 $\mathcal{B}.\text{pop}(\text{batch data})$
 end while
end while
return policy, flow v_ϕ and end training

minmax-norm within the single sequence to regularize the numerical values between $[-1, 1]$. \mathcal{T} and \mathcal{L} are the collections of timesteps and layer percentiles used to calculate the velocity deviations, and the latents are detached from the computational graph for stopping backpropagation. Practically, we incorporate the latents of subsequent token \hat{a}_{k+1} to serve as conditions for assisting velocity prediction in flow reward, that further establishes context dependence with enlarged interactive space, and more ablations are provide in Sec 4.3. The flow reward provides a stable examination on model latents that quantify their velocity deviation from flow pre-trained on off-policy high-quality data.

Updated Rewards with Rejection-Sampling. As the policy are progressing during optimization alongside with their latents (Huan et al., 2025), yielding flow rewards from frozen v_ϕ pre-trained on offline start data introduces inherent distribution gap. Therefore, we update flow by Eq. 4 throughout the policy optimization with online rejection-sampling data, where the filtered metrics are flexible to direct the constitution of reference distribution for flow reward calculation. We provide detailed framework in Algorithm 1, and we empirically found that the correctness is still the most effective metric, where more ablations are provided in Sec. 4.3 for comparison.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Training Data.

We conduct experiments for both language and multimodal models for evaluation. In language settings, we use openr1 (Hugging Face, 2025) as offline start data for flow pretraining, which contains 93k carefully curated mathematical reasoning problems. The reinforcement learning of RLFR is performed on MATH (Hendrycks et al., 2021), which includes diverse reasoning-intensive problems spanning algebra, geometry, number theory, and combinatorics. In multimodal settings, we filter the math subset from MMPR (Wang et al., 2024b) as offline data for flow pretraining, which consists 115k multimodal mathematical reasoning problems. Subsequently, the reinforcement learning is conducted on the MMK12 (Meng et al., 2025a), which includes mathematics, physics, and general science with multimodal contexts.

Evaluation. We assess the language reasoning performance using a suite of standard mathematical reasoning benchmarks, including: AIME24/25¹, AMC23², MATH500 (Hendrycks et al., 2021), and OlympiadBench (He et al., 2024). We report Pass@1 metric with rollout temperature of 0 and Pass@32 with temperature of 0.7 for decoding, under a maximum response length of 8192 tokens. For multimodal reasoning benchmarks, we include MathVista (Lu et al., 2023), MathVerse (Zhang et al., 2024), MathVision (Wang et al., 2024a), WeMath (Qiao et al., 2024), and logic benchmarks

¹<https://huggingface.co/datasets/AI-MO/aimo-validation-aime>

²<https://huggingface.co/datasets/AI-MO/aimo-validation-amc>

Table 1: Overall performance on language reasoning benchmarks. Pass@32 and Pass@1 metrics are reported with zero-shot evaluation. † means the model trained in our setting for evaluation.

Model	AIME25		AIME24		AMC23		MATH500		OlympiadBench	
	Pass@32	Pass@1	Pass@32	Pass@1	Pass@32	Pass@1	Pass@32	Pass@1	Pass@32	Pass@1
<i>Qwen2.5-Math-1.5B</i>	26.7	0.0	43.3	13.3	79.5	32.5	90.2	31.8	61.9	22.8
RLVR	23.3	3.3	46.7	16.7	81.9	41.4	91.8	71.0	62.7	33.5
RLFR	30.0+6.7	6.7+3.4	50.0+3.3	13.3+3.4	83.1+1.2	44.6+3.2	92.4+0.6	72.0+1.0	63.1+0.6	35.7+2.2
<i>Qwen2.5-Math-7B</i>	30.0	3.3	56.6	16.6	80.7	37.3	94.2	50.8	61.2	17.2
Qwen2.5-Math-7B-Inst	36.6	10.0	46.6	13.3	79.5	50.6	-	79.8	-	41.2
Oat-Zero	30.0	6.7	56.6	30.0	81.9	55.4	-	79.6	-	42.6
PRIME	33.3	13.3	53.3	16.7	79.5	55.4	91.8	76.2	65.3	37.9
RLVR	30.0	10.0	56.6	26.7	80.7	57.8	92.0	79.2	64.8	40.1
Entropy Adv.†	26.6	6.6	53.3	20.0	75.9	54.2	93.0	75.6	61.2	36.4
RLFR	33.3+3.3	10.0+0.0	56.6+0.0	30.0+3.3	83.1+2.4	59.0+1.2	92.6+0.6	80.6+1.4	66.1+1.3	41.7+1.6
<i>Llama3.1-8B</i>	0.0	0.0	0.0	0.0	14.4	0.0	28.2	0.6	5.9	0.0
Llama3.1-8B-Inst	16.7	3.3	26.7	10.0	69.8	21.7	-	48.0	-	14.7
RLVR	6.7	0.0	6.7	0.0	25.3	7.2	46.0	18.0	18.6	4.0
RLFR	13.3+6.6	6.7+6.7	16.7+10.0	6.7+6.7	27.7+2.4	12.0+4.8	46.4+0.4	18.8+0.8	19.3+0.7	11.3+7.3

Table 2: Overall performance on multimodal reasoning benchmarks. **Logic Avg.** denotes average of LogicVista and VisuLogic. **Math Avg.** denotes average of other four math benchmarks.

	MathVista	MathVision	MathVerse	WeMath	LogicVista	VisuLogic	Math Avg.	Logic Avg.	Avg.
<i>Qwen2.5VL-3B-Inst</i>	62.0	21.1	33.7	40.7	38.9	26.8	39.4	32.9	37.2
RLVR	65.9	24.1	42.2	55.2	41.1	26.4	46.8	33.7	42.5
RLFR	67.7+1.8	29.6+5.5	42.6+0.4	56.9+1.7	42.3+1.2	25.4+1.0	49.2+2.4	33.8+0.1	44.1+1.6
<i>Qwen2.5VL-7B-Inst</i>	65.6	23.4	47.3	53.4	47.8	27.1	47.4	37.5	44.1
R1-OneVision-7B	63.7	22.4	45.2	52.9	38.9	18.3	46.1	28.6	40.2
OpenVLThinker-7B	64.5	24.3	46.1	50.3	38.7	10.6	46.3	24.6	39.1
MM-Eureka-7B	73.5	27.9	51.9	58.7	46.9	25.5	53.0	36.2	47.4
RLVR	73.1	27.6	48.0	64.6	48.3	24.8	53.3	36.5	47.7
Entropy Adv. †	71.1	26.3	52.1	63.1	44.5	25.7	53.2	35.1	47.1
RLFR	73.4+0.3	31.6+4.0	52.0+4.0	66.1+1.5	48.3+0.0	26.7+1.9	55.8+2.4	37.5+1.0	49.7+1.9

including LogicVista (Xiao et al., 2024) and VisuLogic (Xu et al., 2025). We adopt greedy decoding with temperature of 0 and report Pass@1 metric for evaluation.

Implementation Details. We adopt RL algorithm as GRPO in our experiments. In flow pretraining, we use a training batch size of 128 with 10^{-4} learning rate and warmup ratio of 0.1, where the LLM backbone is frozen. We empirically set the percentiles of layer collection as $\{0.25, 0.5, 0.75\}$ throughout the model, where the layer position embedding are added. The flow network comprises 4 layers for 3B models and 6 layers for 7B/8B models. In reinforcement finetuning, we exclude both KL divergence loss and entropy loss, and use a training batch size of 128 with policy learning rate of 10^{-6} and flow learning rate of 10^{-4} , where the maximal response length is set for 4096. We use the threshold η of 0.6 to discard noisy fluctuations, the validated batch size κ of 32 for flow update, while the discount factor γ and coefficient β are set as 1 and 0.01. The timestep collection \mathcal{T} for yielding flow reward is set for $\{0.8\}$, and we use the temperature of 1 for response rollout.

4.2 MAIN RESULTS

The main experimental results in Table 1 and Table 2 demonstrate that RLFR consistently outperforms baselines across both language and multimodal reasoning benchmarks. The baselines include: basic RLVR with binary verification, existing approaches (Liu et al., 2025e; Yang et al., 2025b; Deng et al., 2025; Meng et al., 2025b), process implicit rewards (Cui et al., 2025a), and entropy-based advantage shaping method (Cheng et al., 2025), which serves as a strong baseline for logit-space comparison. Table 1 reports results on language reasoning benchmarks using Qwen and Llama base models across 1.5B and 7B/8B size, where RLFR shows consistent improvement, surpassing basic RLVR by 1.5% average score on Qwen2.5-Math-7B and by 5.3% average score on Llama3.1-8B, while achieving superior performance compared to entropy-based shaping method in logit space. Table 2 presents results on multimodal reasoning benchmarks, where RLFR achieves compelling improvements on challenging benchmarks like MathVision and MathVerse, and show steady generalization on out-of-domain logic benchmarks. The performance gains across both language and multimodal reasoning benchmarks on different model families show that the flow rewards derived from latent space reliably advance the performance with binary verification, while competitive to logit space shaping.

4.4 ANALYSIS

We further analysis the reward behaviors derived from velocity deviations during training progress to better understand what is being encouraged. In Table 3, we found that: **(1)** Contrast to previous entropy-based method that encouraging tokens with logical connection function to dominate reasoning directions (Wang et al., 2025b; Cheng et al., 2025), the flow rewards prefer tokens that practically execute the question, and depress tokens with empty content such as connection tokens. We attribute this to that the high entropy tokens typically correspond to ambiguity hidden states that prepared for a large set of candidate tokens, which makes it hard to predict in flow field. **(2)** The positive reward tokens are initially related to offline start data as shown in Figure 4, such as `sqrt`, `angle`, and progressively updated with policy. **(3)** Despite the large portion of general words in pretrained dataset, the flow yields limited reward on these tokens rather than completely matching. Additionally, some tokens receive either positive reward or negative reward in cases, e.g., `frac`, indicating that the flow reward is capable of relying on efficient context dependence compressed within the hidden states, rather than individual token-level denotation for context comprehending.

Takeaways

- **Flow rewards prefer tokens that practically execute the question**, and depress tokens with empty content such as connection tokens.
- **High entropy in logit space makes larger velocity deviations in latent space**, attributing to the ambiguity hidden states correspond to a large set of candidate tokens.
- **Flow rewards rely on efficient context dependence compressed within the hidden states**, rather than individual token-level denotation for context comprehending.

5 RELATED WORK

Reinforcement learning beyond binary verification. Reinforcement learning with binary verifiable reward has recently demonstrates promising effectiveness in advancing reasoning abilities of Large Language Models (Yu et al., 2025a; Hu et al., 2025; DeepSeek-AI et al., 2025). Despite robustness to reward hacking, the binary verification largely restricts the potential valuable exploration in reasoning trajectory. Indicated by the policy entropy (Cui et al., 2025b; Wang et al., 2025b), recent practices leverage a variety of metrics derived from model likelihood either to shape the reward signal (Cheng et al., 2025; Damani et al., 2025; He et al., 2025; Li et al., 2025b), or to serve as indicators for identifying tokens with different optimization (Wang et al., 2025a;b; Fu et al., 2025). However, this work highlights the latent space could be an expressive substrate for reliable reward collection, complementing prior methods that primarily focus on logit space. Additionally, recent work also adopts the pass@k training (Chen et al., 2025) that tolerates incorrect answer with potential valuable trajectory, which is orthogonal to this work and also a promising direction.

Flow Matching in Reinforcement Learning. As the most effective continuous modeling framework, flow matching (Lipman et al., 2022; Liu et al., 2022) is especially expert at generating high-dimensional signals, and has achieved remarkable success across a wide range of domains. Building on recent progress of RLVR, series of works have been proposed to further advance the generation quality in respective areas, including visual generation (Liu et al., 2025a; Li et al., 2025a; Xue et al., 2025) and robotics (Pfrommer et al., 2025; McAllister et al., 2025). These works leverage the flow as the policy model for optimization, which is distinct from RLFR that uses flow as environment for reward collection, and concentrates on velocity deviation metrics rather than reverse the process.

6 CONCLUSION

In this work, we analysis the auxiliary signals for reward shaping of RLVR from the perspective of latent space, and show that the latent space is highly expressive yet underexplored, complementing prior methods that focus closely on logit space. In light of this, RLFR offers a novel framework on shaping RLVR with flow reward, where the flow field of model latent are constructed from either off-policy high-quality data and on-policy rejection sampling data, and the deviations of policy latents within it are quantified to serve as a reward signal, extending RLVR for latent reward uti-

486 lization. RLFR first demonstrates that a well-established flow field can be a sound environment for
 487 reward signal collection, yielding steady performance improvements across both language and mul-
 488 timodal benchmarks, highlighting the potential of latent substrate for reward design. Additionally,
 489 RLFR naturally leverages expert reasoning trajectories from off-policy data into the constitution
 490 of reward signal, instead of relying on self-confidence. Future directions involve scaling the flow
 491 environment to release the latent potential, and the prospect of latent signals for test-time scaling.

492 REFERENCES

- 493 Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint*
 494 *arXiv:2502.04463*, 2025.
- 495 Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi.
 496 Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models.
 497 *arXiv preprint arXiv:2508.10751*, 2025.
- 498 Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and
 499 Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*,
 500 2025.
- 501 Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li,
 502 Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards.
 503 *arXiv preprint arXiv:2502.01456*, 2025a.
- 504 Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen
 505 Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for
 506 reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025b.
- 507 Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob
 508 Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint*
 509 *arXiv:2507.16806*, 2025.
- 510 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu,
 511 Qihao Zhu, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement
 512 learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- 513 Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker:
 514 An early exploration to complex vision-language reasoning via iterative self-improvement. *arXiv*
 515 *preprint arXiv:2503.17352*, 2025.
- 516 Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao
 517 Zhang, Yuanheng Zhu, and Dongbin Zhao. Srft: A single-stage method with supervised and
 518 reinforcement fine-tuning for reasoning. *arXiv preprint arXiv:2506.19767*, 2025.
- 519 Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin P Murphy, and
 520 Tim Salimans. Diffusion meets flow matching: Two sides of the same coin. 2024. URL
 521 <https://diffusionflow.github.io>, 2024.
- 522 Hasan Abed Al Kader Hammoud, Hani Itani, and Bernard Ghanem. Beyond the last answer: Your
 523 reasoning trace uncovers more than you think. *arXiv preprint arXiv:2504.20708*, 2025.
- 524 Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution
 525 sharpening. *arXiv preprint arXiv:2506.02355*, 2025.
- 526 Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu,
 527 Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for
 528 promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint*
 529 *arXiv:2402.14008*, 2024.
- 530 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 531 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv*
 532 *preprint arXiv:2103.03874*, 2021.

- 540 Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum.
541 Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base
542 model. *arXiv preprint arXiv:2503.24290*, 2025.
- 543
- 544 Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Pooven-
545 dran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities?
546 understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.
- 547 Hugging Face. Open rl: A fully open reproduction of deepseek-rl, January 2025. URL <https://github.com/huggingface/open-rl>.
- 548
- 549
- 550 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec
551 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard
552 Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett,
553 Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey
554 Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben
555 Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen
556 Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin,
557 Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris
558 Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kap-
559 pler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson,
560 Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Eliz-
561 abeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang,
562 Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred
563 von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace
564 Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart An-
565 drin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichen,
566 Ian O’Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever,
567 Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng,
568 Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñero Candela, Joe Palermo, Joel Parish,
569 Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan
570 Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl
571 Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu,
572 Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam
573 Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kon-
574 draciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen,
575 Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet
576 Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael
577 Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles
578 Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil
579 Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg
580 Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov,
581 Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar
582 Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan
583 Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agar-
584 wal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu,
585 Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph
586 Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Tay-
587 lor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson,
588 Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna
589 Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiye
590 Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen,
591 Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li.
592 Openai o1 system card, 2024. URL <https://arxiv.org/abs/2412.16720>.
- 593
- 590 Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data
591 augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.
- 592
- 593 Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Ad-
vances in neural information processing systems*, 34:21696–21707, 2021.

- 594 Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brah-
595 man, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers
596 in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.
- 597 Junzhe Li, Yutao Cui, Tao Huang, Yinpeng Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo:
598 Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*,
599 2025a.
- 600 Yi-Chen Li, Tian Xu, Yang Yu, Xuqin Zhang, Xiong-Hui Chen, Zhongxiang Ling, Ningjing Chao,
601 Lei Yuan, and Zhi-Hua Zhou. Generalist reward models: Found inside large language models.
602 *arXiv preprint arXiv:2506.23235*, 2025b.
- 603 Yizhuo Li, Yuying Ge, Yixiao Ge, Ying Shan, and Ping Luo. Aligning latent spaces with flow priors.
604 *arXiv preprint arXiv:2506.05240*, 2025c.
- 605 Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching
606 for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- 607 Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan,
608 Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv*
609 *preprint arXiv:2505.05470*, 2025a.
- 610 Mingjie Liu, Shizhe Diao, Jian Hu, Ximing Lu, Xin Dong, Hao Zhang, Alexander Bukharin,
611 Shaokun Zhang, Jiaqi Zeng, Makesh Narsimhan Sreedhar, et al. Scaling up rl: Unlocking di-
612 verse reasoning in llms via prolonged training. *arXiv preprint arXiv:2507.12507*, 2025b.
- 613 Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong.
614 Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models.
615 *arXiv preprint arXiv:2505.24864*, 2025c.
- 616 Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen
617 Zhou. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint*
618 *arXiv:2502.06703*, 2025d.
- 619 Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and
620 transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- 621 Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee,
622 and Min Lin. Understanding rl-zero-like training: A critical perspective. *arXiv preprint*
623 *arXiv:2503.20783*, 2025e.
- 624 Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-
625 Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of
626 foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.
- 627 David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen
628 Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*,
629 2025.
- 630 Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng
631 Han, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng
632 Zhang, and Wenqi Shao. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-
633 based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025a.
- 634 Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng
635 Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of mul-
636 timodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*,
637 2025b.
- 638 Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations:
639 Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.
- 640 Samuel Pfrommer, Yixiao Huang, and Somayeh Sojoudi. Reinforcement learning for flow-matching
641 policies. *arXiv preprint arXiv:2507.15073*, 2025.

- 648 Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma
649 GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. We-math: Does your large multi-
650 modal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*,
651 2024.
- 652 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy
653 optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- 654 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
655 Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathe-
656 matical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- 657 Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of
658 score-based diffusion models. *Advances in neural information processing systems*, 34:1415–
659 1428, 2021.
- 660 Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. Stabilizing knowledge, pro-
661 moting reasoning: Dual-token constraints for rlvr. *arXiv preprint arXiv:2507.15778*, 2025a.
- 662 Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hong-
663 sheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in*
664 *Neural Information Processing Systems*, 37:95095–95169, 2024a.
- 665 Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen,
666 Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive
667 effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025b.
- 668 Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu,
669 Xizhou Zhu, Lewei Lu, Yu Qiao, et al. Enhancing the reasoning ability of multimodal large
670 language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024b.
- 671 Zihao Wang, Chirag Nagpal, Jonathan Berant, Jacob Eisenstein, Alex D’Amour, Sanmi Koyejo,
672 and Victor Veitch. Transforming and combining rewards for aligning large language models. In
673 *Proceedings of the 41st International Conference on Machine Learning, ICML’24*, 2024c.
- 674 Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning
675 benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.
- 676 Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu,
677 Houqiang Li, Xiaohua Wang, Xizhou Zhu, et al. Visulogic: A benchmark for evaluating visual
678 reasoning in multi-modal large language models. *arXiv preprint arXiv:2504.15279*, 2025.
- 679 Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei
680 Liu, Qiushan Guo, Weilin Huang, et al. Dancegrp: Unleashing grp on visual generation. *arXiv*
681 *preprint arXiv:2505.07818*, 2025.
- 682 An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li,
683 Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint*
684 *arXiv:2412.15115*, 2024.
- 685 Wenkai Yang, Jingwen Chen, Yankai Lin, and Ji-Rong Wen. Deepcritic: Deliberate critique with
686 large language models. *arXiv preprint arXiv:2505.00662*, 2025a.
- 687 Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng
688 Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal rea-
689 soning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025b.
- 700 Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and
701 Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through rl
training. *arXiv preprint arXiv:2509.03403*, 2025.

702 Qiyong Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong
703 Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at
704 scale. *arXiv preprint arXiv:2503.14476*, 2025a.
705
706 Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan
707 Yao, Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv
708 preprint arXiv:2506.18254*, 2025b.
709
710 Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What’s behind ppo’s collapse in
711 long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025.
712
713 Yu Yue, Yufeng Yuan, Qiyong Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiase Chen, Chengyi
714 Wang, Tiantian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for
715 advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.
716
717 Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou,
718 Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the
719 diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186.
720 Springer, 2024.
721
722 Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian,
723 Biqing Qi, Xiu Li, et al. Genprm: Scaling test-time compute of process reward models via
724 generative reasoning. *arXiv preprint arXiv:2504.00891*, 2025.
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

A EXPERIMENTAL DETAILS

A.1 SETTINGS

Experiments are conducted on 8 H20 GPUs. We use AdamW optimizer and sample 8 rollouts per prompt with (0.2, 0.28) clip range in policy loss for training. The prompt template is shown in Tab. 7, where we adopt bbox template for language training and tag template for multimodal training. In evaluation, we deploy a Qwen2.5-7B-Instruct model server for answer extraction and judge, and adopt DeepSeek v3.1 for more complex benchmarks, such as MathVision and MathVerse.

RLFR training prompt	Bbox
<pre> < im_start >system Please reason step by step, and put your final answer within <code>\\boxed{}</code>. < im_end > < im_start >user {{question}}< im_end > < im_start >assistant </pre>	
RLFR training prompt	Tag
<pre> < im_start >system You should first think about the reasoning process in the mind and then provides the user with the answer. Your answer must be in latex format and wrapped in <code>\$...\$</code>. The reasoning process and answer are enclosed within <code><think></code> <code></think></code> and <code><answer></code> <code></answer></code> tags, respectively, i.e., <code><think></code> Since $1+1=2$, so the answer is 2. <code></think></code> <code><answer></code> 2 <code></answer></code>, which means your output should start with <code><think></code> and end with <code></answer></code>. < im_end > < im_start >user {{question}}< im_end > < im_start >assistant </pre>	

Table 7: Training prompt for RLFR.

A.2 TRAINING LOGS

We monitor the training dynamics of RLVR and RLFR in Fig. 6 for comparison. During training, the flow reward derived from latent space steadily improves the reasoning performance and accelerates the policy optimization, validating the reliability of the latent signals and their underexplored expressiveness. The policy entropy of RLFR also stabilized at a slightly higher level during the training plateau compared to RLVR, underscoring the effectiveness of velocity deviation as the dense reward for encouraging exploration. While the response length shows healthy behavior with steady increases and adjustment, while no sign of degeneration are observed.

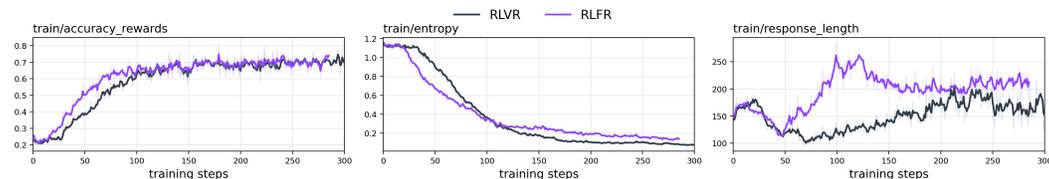


Figure 6: Training logs of **RLVR** and **RLFR** on Qwen2.5VL-3B.

B EXTENDED ANALYSIS OF LATENT SPACE

Sec. 3.1 analysis the latent space of Qwen2.5-Base-7B at layer percentile of 0.5. We further show that the broad latent space exhibits the similar tendency throughout the LLM. In Fig. 7, we provide

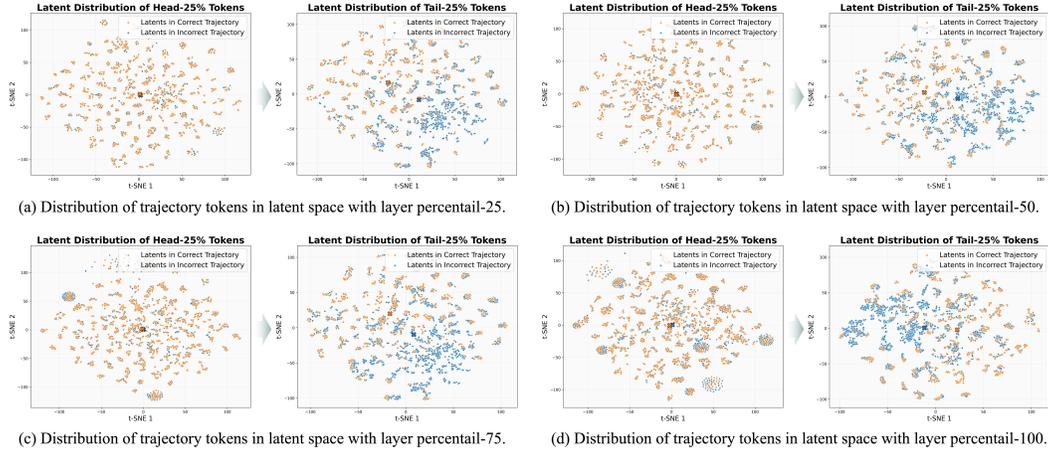


Figure 7: Distribution of reasoning trajectory tokens in latent space across different layer percentiles of the Qwen2.5-Base-7B, which consistently show expressive signals on tail trajectory tokens, highlighting the broader potential of latent space for reward signal collection.

the latent distributions of reasoning trajectory tokens at the $\{25, 50, 75, 100\}$ layer percentiles, and found that there is no evidence of contradiction and specialization across layer positions, instead, the latent space exhibits coherent and consistent signals for trajectory quality identification. In practice, we exclude the 100th percentile in training, as the last hidden states are heavily modulated by the *lm.head* for logit prediction, and we therefore rely on intermediate percentiles for reward collection.

C THEORETICAL ANALYSIS

C.1 DERIVATION OF SCORE FUNCTION FOR VELOCITY

Here, we provide the derivations for Eq. 6 that establish the connection between velocity prediction and score function.

Proof. We consider the linear interpolation

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \beta_t \mathbf{x}_1, \quad \mathbf{x}_0 \sim \mathcal{N}(0, I), \quad \mathbf{x}_1 \sim p_{\text{data}},$$

where \mathbf{x}_0 and \mathbf{x}_1 are independent. We denote derivatives $\dot{\alpha}_t = \frac{d}{dt} \alpha_t$, $\dot{\beta}_t = \frac{d}{dt} \beta_t$. Conditioned on the data \mathbf{x}_1 , the noisy variable \mathbf{x}_t is Gaussian:

$$p_t(\mathbf{x}_t | \mathbf{x}_1) \sim \mathcal{N}(\beta_t \mathbf{x}_1, \alpha_t^2 I),$$

with conditional score

$$\nabla_x \log p(\mathbf{x}_t | \mathbf{x}_1) = -\frac{\mathbf{x}_t - \beta_t \mathbf{x}_1}{\alpha_t^2}. \quad (11)$$

By Fisher’s identity, the marginal score given by

$$\begin{aligned} \mathbf{s}_t(x) &= \nabla_x \log p_t(x) = \mathbb{E}[\nabla_x \log p(\mathbf{x}_t | \mathbf{x}_1) | \mathbf{x}_t = x] \\ &= -\frac{1}{\alpha_t^2} \left(x - \beta_t \mathbb{E}[\mathbf{x}_1 | \mathbf{x}_t = x] \right). \end{aligned} \quad (12)$$

Rearranging yields

$$\mathbb{E}[\mathbf{x}_1 | \mathbf{x}_t = x] = \frac{1}{\beta_t} \left(x + \alpha_t^2 \mathbf{s}_t(x) \right). \quad (13)$$

Considering $\frac{d}{dt} \mathbf{x}_t = \dot{\alpha}_t \mathbf{x}_0 + \dot{\beta}_t \mathbf{x}_1$, thus the velocity field is

$$\begin{aligned} \mathbf{v}_t(x) &= \mathbb{E} \left[\dot{\alpha}_t \mathbf{x}_0 + \dot{\beta}_t \mathbf{x}_1 \mid \mathbf{x}_t = x \right] \\ &= \dot{\alpha}_t \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t = x] + \dot{\beta}_t \mathbb{E}[\mathbf{x}_1 | \mathbf{x}_t = x]. \end{aligned} \quad (14)$$

Since $\mathbf{x}_0 = (\mathbf{x}_t - \beta_t \mathbf{x}_1)/\alpha_t$, we have

$$\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t = x] = \frac{x}{\alpha_t} - \frac{\beta_t}{\alpha_t} \mathbb{E}[\mathbf{x}_1 | \mathbf{x}_t = x]. \quad (15)$$

Substituting the Eq. 15 and Eq. 13, the velocity field is given by

$$\begin{aligned} \mathbf{v}_t(x) &= \frac{\dot{\alpha}_t}{\alpha_t} x + \left(\dot{\beta}_t - \frac{\dot{\alpha}_t \beta_t}{\alpha_t} \right) \mathbb{E}[\mathbf{x}_1 | \mathbf{x}_t = x] \\ &= \frac{\dot{\beta}_t}{\beta_t} x + \alpha_t^2 \left(\frac{\dot{\beta}_t}{\beta_t} - \frac{\dot{\alpha}_t}{\alpha_t} \right) \mathbf{s}_t(x). \end{aligned} \quad (16)$$

For linear schedule, where $\alpha_t = 1 - t$ and $\beta_t = t$, we have

$$\mathbf{v}_t(x) = \frac{1}{t} x + \frac{1-t}{t} \mathbf{s}_t(x). \quad (17)$$

The score function is given by

$$\mathbf{s}_t(x) = -\frac{x}{1-t} + \frac{t}{1-t} \mathbf{v}_t(x). \quad (18)$$

Eq. 17 and Eq. 18 show the exact equivalence between the score function and velocity field under the linear interpolation schedule.

□

C.2 PROOF OF VARIATIONAL LOWER BOUND

Here, we provide the proof of Eq. 8 that establish the connection between the velocity deviations which measured by flow matching objective, and the likelihood given by evidence lower bound (ELBO) under target distribution. We indicate that the connection between ELBO and diffusion objective has been shown by previous works Kingma et al. (2021); Song et al. (2021); Kingma & Gao (2023), and we further extend it to velocity field.

Proof. For sample \mathbf{y} that need to be evaluated for velocity deviation in reference flow distribution and consider the linear interpolation $\mathbf{y}_t = \alpha_t \mathbf{x}_0 + \beta_t \mathbf{y}$, with $\mathbf{x}_0 \sim p_{\text{init}}$. Let $\mathbf{s}_t(\mathbf{y}_t) = \nabla_{\mathbf{y}} \log p_{\mathbf{v}_\phi}(\mathbf{y}_t)$ be the score of \mathbf{y}_t . Recall the flow matching objective in Eq. 5, we take

$$\mathcal{L}_{\text{FM}}(\mathbf{y}; \phi) = \int_0^1 \mathbb{E}_{\mathbf{y} \sim q_t} \left[\frac{1}{2} \|v_\phi(\mathbf{y}, t) - u_t^{\mathbf{y}}\|^2 \right] dt.$$

By continuity equation, the density and velocity of flow defined by $\mathbf{v}_\phi(\mathbf{y}_t, t)$ satisfies

$$\partial_t p_{\mathbf{v}_\phi}(\mathbf{y}_t) + \nabla \cdot (p_{\mathbf{v}_\phi}(\mathbf{y}_t) \mathbf{v}_\phi(\mathbf{y}_t, t)) = 0. \quad (19)$$

Given $\nabla \log p_{\mathbf{v}_\phi}(\mathbf{y}_t) = \nabla p_{\mathbf{v}_\phi}(\mathbf{y}_t)/p_{\mathbf{v}_\phi}(\mathbf{y}_t)$, we have

$$\partial_t \log p_{\mathbf{v}_\phi}(\mathbf{y}_t) = -\nabla \cdot \mathbf{v}_\phi(\mathbf{y}_t, t) - \mathbf{v}_\phi(\mathbf{y}_t, t) \cdot \nabla_{\mathbf{y}} \log p_{\mathbf{v}_\phi}(\mathbf{y}_t). \quad (20)$$

Considering $\frac{d}{dt} \mathbf{y}_t = v_\phi(\mathbf{y}_t, t)$, and substituting the Eq. 20, the total derivative of $\log p_{\mathbf{v}_\phi}(\mathbf{y}_t)$ is

$$\begin{aligned} \frac{d}{dt} \log p_{\mathbf{v}_\phi}(\mathbf{y}_t) &= \partial_t \log p_{\mathbf{v}_\phi}(\mathbf{y}_t) + \nabla_{\mathbf{y}} \log p_{\mathbf{v}_\phi}(\mathbf{y}_t) \cdot \frac{d}{dt} \mathbf{y}_t \\ &= -\nabla \cdot \mathbf{v}_\phi(\mathbf{y}_t, t) - \mathbf{v}_\phi(\mathbf{y}_t, t) \cdot \nabla_{\mathbf{y}} \log p_{\mathbf{v}_\phi}(\mathbf{y}_t) + \nabla_{\mathbf{y}} \log p_{\mathbf{v}_\phi}(\mathbf{y}_t) \cdot \frac{d}{dt} \mathbf{y}_t \\ &= -\nabla \cdot \mathbf{v}_\phi(\mathbf{y}_t, t). \end{aligned} \quad (21)$$

Therefore, integrating over $t \in [0, 1]$ yields the standard change-of-variables formula

$$\begin{aligned} \log p_{\mathbf{v}_\phi}(\mathbf{y}) &= \log p_{\text{init}}(\mathbf{x}_0) - \int_0^1 \nabla \cdot \mathbf{v}_\phi(\mathbf{y}_t, t) dt \\ &= \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{init}}}[\log p_{\text{init}}(\mathbf{x}_0)] - \int_0^1 \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}}[\nabla \cdot \mathbf{v}_\phi(\mathbf{y}_t, t)] dt \\ &= C_0(x) - \int_0^1 \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}}[\langle \mathbf{v}_\phi(\mathbf{y}_t, t), \nabla \log p_{\text{data}}(\mathbf{y}_t) \rangle] dt \end{aligned} \quad (22)$$

where $\langle \cdot, \cdot \rangle$ denotes inner product, and we derive the last step from the Stein's identity. Considering

$$\langle a, g \rangle = \langle a, b \rangle - \frac{1}{2} \|b\|^2 + \frac{1}{2} \|g\|^2 - \frac{1}{2} \|g - b\|^2,$$

we thus substitute $a = \mathbf{v}_\phi(\mathbf{y}, t)$, $b = u_t^y$, and $g = \nabla \log p_{\text{data}}(y)$, yielding

$$\langle \mathbf{v}_\phi, \nabla \log p_{\text{data}} \rangle = \langle \mathbf{v}_\phi, u_t^y \rangle - \frac{1}{2} \|u_t^y\|^2 + \frac{1}{2} \|\nabla \log p_{\text{data}}\|^2 - \frac{1}{2} \|\nabla \log p_{\text{data}} - u_t^y\|^2.$$

We simplify $\mathbf{v}_\phi(\mathbf{y}_t, t)$ as \mathbf{v}_ϕ and $p_{\text{data}}(\mathbf{y}_t)$ as p_{data} for clarity, and have

$$\int_0^1 \mathbb{E}_{p_{\text{data}}}[\langle \mathbf{v}_\phi, \nabla \log p_{\text{data}} \rangle] dt = \int_0^1 \mathbb{E}_{p_{\text{data}}}[\langle \mathbf{v}_\phi, u_t^y \rangle] dt + B(y), \quad (23)$$

where $B(y)$ depends only on the fixed bridge (p_{data}, u_t^y) , and have

$$B(y) = \int_0^1 \mathbb{E}_{p_{\text{data}}} \left[-\frac{1}{2} \|u_t^y\|^2 + \frac{1}{2} \|\nabla \log p_{\text{data}}\|^2 - \frac{1}{2} \|\nabla \log p_{\text{data}} - u_t^y\|^2 \right] dt. \quad (24)$$

Considering the Fenchel–Young inequality

$$\langle \mathbf{v}_\phi, u_t^y \rangle \geq -\frac{1}{2\lambda} \|\mathbf{v}_\phi\|^2 - \frac{\lambda}{2} \|u_t^y\|^2, \quad \lambda > 0,$$

where we substitute into Eq. 23, and have

$$\begin{aligned} \int_0^1 \mathbb{E}_{p_{\text{data}}}[\langle \mathbf{v}_\phi, \nabla \log p_{\text{data}} \rangle] dt &\geq \int_0^1 \mathbb{E}_{p_{\text{data}}} \left[-\frac{1}{2\lambda} \|\mathbf{v}_\phi\|^2 - \frac{\lambda}{2} \|u_t^y\|^2 \right] dt + B(y) \\ &= \int_0^1 \mathbb{E}_{p_{\text{data}}} \left[-\frac{1}{2\lambda} \|\mathbf{v}_\phi - \lambda u_t^y\|^2 \right] dt + B(y) \end{aligned} \quad (25)$$

For the special case $\lambda = 1$, this simplifies to the equivalence given by Eq 8. Therefore, combining with Eq. 22, we obtain

$$\log p_{\mathbf{v}_\phi}(\mathbf{y}) \geq C(y) - \lambda \int_0^1 \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}} \left[\frac{1}{2} \|\mathbf{v}_\phi(\mathbf{y}_y, t) - u_t^y\|^2 \right] dt,$$

which establish the connection between the velocity and likelihood under target distribution. \square

D CASE STUDY

Question and Ground Truth

Question: Jen enters a lottery by picking 4 distinct numbers from $S = \{1, 2, 3, \dots, 9, 10\}$. 4 numbers are randomly chosen from S . She wins a prize if at least two of her numbers were 2 of the randomly chosen numbers, and wins the grand prize if all four of her numbers were the randomly chosen numbers. The probability of her winning the grand prize given that she won a prize is $\frac{m}{n}$ where m and n are relatively prime positive integers. Find $m + n$.

Ground Truth: 116

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

Response from RLVR Baseline

To solve this problem, we need to calculate the probability of Jen winning the grand prize given that she won a prize. We can break this down into several steps:

1. Calculate the total number of ways to choose 4 numbers from the set ($S = \{1, 2, 3, \dots, 10\}$).
2. Calculate the number of ways Jen can win a prize (i.e., at least 2 of her numbers are among the 4 chosen numbers).
3. Calculate the number of ways Jen can win the grand prize (i.e., all 4 of her numbers are among the 4 chosen numbers).
4. Use the conditional probability formula to find the probability of winning the grand prize given that she won a prize.

Let's go through these steps one by one.

1. The total number of ways to choose 4 numbers from 10 is given by the combination formula ($\binom{10}{4}$).
2. The number of ways Jen can win a prize is the sum of the number of ways she can have exactly 2, 3, or 4 of her numbers among the 4 chosen numbers. This can be calculated as:
 - Number of ways to have exactly 2 of her numbers among the 4 chosen numbers: ($\binom{4}{2} \cdot \binom{6}{2}$)
 - Number of ways to have exactly 3 of her numbers among the 4 chosen numbers: ($\binom{4}{3} \cdot \binom{6}{1}$)
 - Number of ways to have exactly 4 of her numbers among the 4 chosen numbers: ($\binom{4}{4} \cdot \binom{6}{0}$)
3. The number of ways Jen can win the grand prize is simply the number of ways to choose 4 numbers from her 4 chosen numbers, which is ($\binom{4}{4} \cdot \binom{6}{0} = 1$).
4. The probability of winning the grand prize given that she won a prize is the ratio of the number of ways to win the grand prize to the number of ways to win a prize.

Let's calculate this using Python.

```
import math

# Total number of ways to choose 4 numbers from 10
total_ways = math.comb(10, 4)

# Number of ways to win a prize
ways_to_win_prize = (math.comb(4, 2) * math.comb(6, 2)
                    + math.comb(4, 3) * math.comb(6, 1)
                    + math.comb(4, 4) * math.comb(6, 0))

# Number of ways to win the grand prize
ways_to_win_grand_prize = math.comb(4, 4) * math.comb(6, 0)

# Probability of winning the grand prize given that she
# won a prize
probability = ways_to_win_grand_prize / ways_to_win_prize

from fractions import Fraction
fraction = Fraction(ways_to_win_grand_prize, ways_to_win_prize)
m, n = fraction.numerator, fraction.denominator

result = m + n
print(result)

output 47
```

The probability of Jen winning the grand prize given that she won a prize is ($\frac{1}{47}$). Therefore, the values of (m) and (n) are 1 and 47, respectively. The sum (m + n) is (1 + 47 = 48).

So, the final answer is ($\boxed{48}$).

1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

Response from RLFR

To solve this problem, we need to calculate the probability of Jen winning the grand prize given that she won a prize. Let's break down the problem step-by-step:

1. **Total number of ways to choose 4 numbers from 10:** $\binom{10}{4} = \frac{10!}{4!(10-4)!} = 210$

2. **Number of ways Jen can win a prize:**

- Jen wins a prize if at least 2 of her numbers are among the 4 chosen numbers.

- We need to calculate the number of ways she can have 2, 3, or 4 of her numbers among the 4 chosen numbers.

- Number of ways to have exactly 2 of her numbers among the 4 chosen numbers: $\binom{4}{2} \cdot \binom{6}{2} = 6 \cdot 15 = 90$

- Number of ways to have exactly 3 of her numbers among the 4 chosen numbers: $\binom{4}{3} \cdot \binom{6}{1} = 4 \cdot 6 = 24$

- Number of ways to have exactly 4 of her numbers among the 4 chosen numbers: $\binom{4}{4} \cdot \binom{6}{0} = 1 \cdot 1 = 1$

- Total number of ways Jen can win a prize: $90 + 24 + 1 = 115$

3. **Number of ways Jen can win the grand prize:** - Jen wins the grand prize if all 4 of her numbers are among the 4 chosen numbers.

- There is only 1 way for this to happen: $\binom{4}{4} = 1$

4. **Probability of winning the grand prize given that she won a prize:**

$$\frac{\text{Number of ways to win the grand prize}}{\text{Number of ways to win a prize}} = \frac{1}{115}$$

5. **Final answer:**

The probability is $\frac{1}{115}$, so $m = 1$ and $n = 115$. Since 1 and 115 are relatively prime, the final answer is: $m + n = 1 + 115 = 116$

The final answer is: $\boxed{116}$

E THE USE OF LARGE LANGUAGE MODELS

We used GPT5 solely for limited auxiliary purposes: (i) polishing the language of paragraphs, and (ii) generating refined format of latex code for showing. All scientific concepts, methodologies, and results were independently conceived, verified, and validated by the authors. Any outputs suggested by the LLM were manually reviewed to ensure correctness.