RLFR: EXTENDING REINFORCEMENT LEARNING FOR LLMS WITH FLOW ENVIRONMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a promising framework for strengthening reasoning abilities of Large Language Models (LLMs). While policy optimized with binary verification prone to overlook potential valuable exploration in reasoning trajectory. Given the heavy annotation cost of Process Reward Models (PRMs), recent works investigate auxiliary signals of reward shaping for trajectory tokens, including entropy, likelihood, and teacher divergence derived from the 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 yet context dependent latent space is much underexplored. Moreover, RLFR is able to compress any off-policy expert trajecories as reference for constituting reward signals, and we show that RLFR achieves efficiency on par with RLVR with reliable process rewarding. Experiments on both language and multimodal benchmarks demonstrate the effectiveness of flow rewards, suggesting a promising paradigm for credit allocation.

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., 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, binary verification is prone to overlooking potentially valuable policy explorations, particularly when trajectories are partially correct with derived incorrect answers (Hammoud et al., 2025), thereby producing an overly strict reward signal that suppresses exploratory behavior (Cui et al., 2025).

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., 2025b), 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., 2025a) and token entropy (Cheng et al., 2025; Wang et al., 2025a). 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., 2025). As the self-policy rewarding may cause LLM over exploits its own confidence estimates rather than learning genuinely improved reasoning strategies, and the potential hacking risks are non-negligible, and may undermine prolonged 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*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

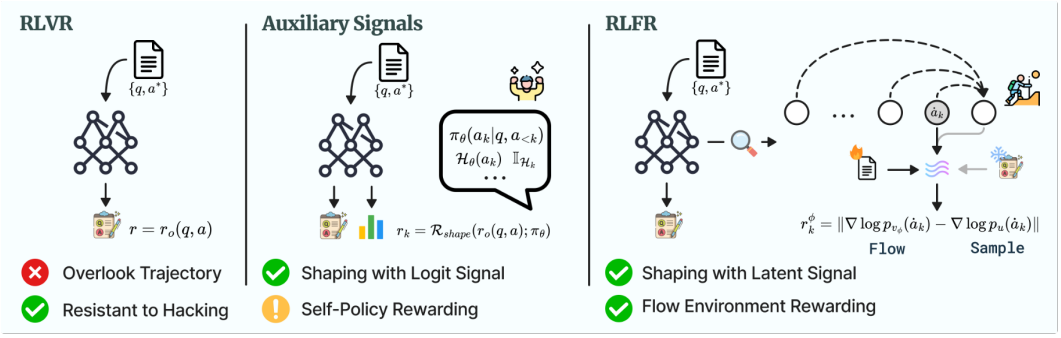


Figure 1: Policy optimized with RLVR prone to overlook potential valuable explorations in reasoning trajectories. Beyond binary verification, logit space signals including token entropy and likelihood are collected to shape rewards over process tokens, however, reliance on self-policy evaluation introduces non-negligible hacking risks. 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.

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 further show that the negative velocity deviation constitutes the evidence lower bound on the log-likelihood, establishing an inverse correlation between velocity deviation and likelihood.

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 yet context accumulated latent space is highly underexplored as a substrate for reward design, complementing prior auxiliary signals from logit space with eliminated context dependence for vocabulary projection. Moreover, RLFR provides a natural way to leverage off-policy expert trajectories into the constitution of on-policy reward signals through reference flow fields. And we further show that the additional process rewarding from RLFR introduces negligible overhead, compared to heavy costs of PRM baselines and achieves on par efficiency with RLVR. Experiments on both language and multimodal reasoning benchmarks 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 as reference for constituting flow reward. (4) Comprehensive experiments on a broad range of reasoning benchmarks across both language and multimodal models demonstrate the reliability of our method.

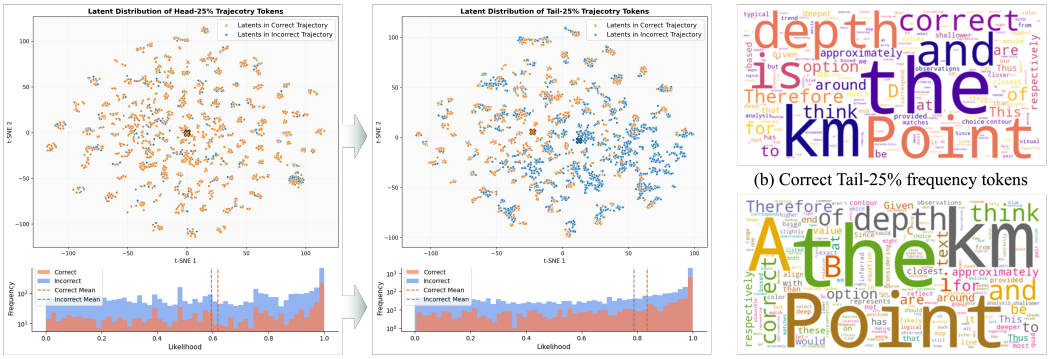
2 Preliminaries

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 $\boldsymbol{a} = (a_1, ..., a_K)$ as response. A binary verifier then assigns a scaler reward $r_o(q, \boldsymbol{a}) \in \{0, 1\}$ to each prompt-response pair, where r_o underlines its outcome nature. The goal of RL is to maximize the expected reward:

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

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.



(a) Distribution of trajectory tokens in latent space (up) and logit space (down)

(b) Correct Tail-25% frequency tokens

(c) Incorrect Tail-25% frequency tokens

Figure 2: 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.

RLVR Algorithms. Group Relative Policy Optimization (GRPO) (Shao et al., 2024a) as a widely used reinforcement learning algorithm that simplfies 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 objective:

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

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 $\widehat{A}_{i,o}$ is shared among all tokens within response $\boldsymbol{a}_i$ and is computed as

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

We denote $r_{i,o} = r_o(q, \boldsymbol{a}_i)$ 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 $(\dot{\boldsymbol{x}}_0, \dot{\boldsymbol{x}}_1)$ sampled from $p_{\text{init}} \times p_{\text{data}}$, the forward process is given by the linear interpolation: $\dot{\boldsymbol{x}}_t = (1-t)\dot{\boldsymbol{x}}_0 + t\dot{\boldsymbol{x}}_1$, and the neural network $\boldsymbol{v}_\phi$ is trained to predict the target velocity field $\boldsymbol{u} = \dot{\boldsymbol{x}}_1 - \dot{\boldsymbol{x}}_0$ by minimizing the flow matching loss:

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

We denote $\dot{\boldsymbol{x}}$ as latent signal and distinguish it from corresponding tokens $\boldsymbol{x}$. The $\boldsymbol{v}_\phi$ characterizes the p_{data} with flow field through velocity prediction, where the outsider can be notably identified.

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 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), 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 Qwen3-4B (Yang et al., 2025a) 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 $\mathcal{L} = (\tau_0 L, ...\tau_N L)$ 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 2, while additional results are provided in Appendix B and show the similar tendency.

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.

Latents of subset tokens within incorrect reasoning trajectories exhibit patterns close to those in correct trajectories, however, RLVR with outcome verification equally 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

$$\mathcal{R}^{\phi}_{\text{FM}}(\dot{\boldsymbol{a}}_k; t, \tau) = \|\boldsymbol{v}_\phi(\dot{\boldsymbol{a}}_{k,t}, t) - (\dot{\boldsymbol{a}}_{k,1} - \epsilon)\|^2, \tag{5}$$

where $t, \tau \sim \mathcal{U}[0, 1]$, $\epsilon \sim \mathcal{N}(0, I)$, $\dot{\boldsymbol{a}}_k$ denotes latents of token $\boldsymbol{a}_k$, and $\dot{\boldsymbol{a}}_{k,t}$ is the linear interpolation between $\dot{\boldsymbol{a}}_k$ and ϵ. The flow network $\boldsymbol{v}_\phi$ 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 $\dot{\boldsymbol{a}}_k$ under the reference flow field. In Figure 3, we examine the velocity deviations of policy latents under the formed flow field in training, where correct trajectories align more closely with the flow field, resulting in smaller deviations, while incorrect trajectories exhibit larger deviations.

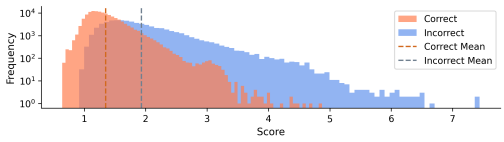


Figure 3: Distribution of velocity deviations for correct and incorrect trajectories.

Debiasing the Timestep Impacting. The flow network $\boldsymbol{v}_\phi$ 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 $\boldsymbol{v}_\phi$ as:

$$\nabla_{\dot{\boldsymbol{a}}_{k,t}} \log p_{\boldsymbol{v}_\phi}(\dot{\boldsymbol{a}}_{k,t}) = -\frac{\dot{\boldsymbol{a}}_{k,t}}{1-t} + \frac{t}{1-t}\boldsymbol{v}_\phi(\dot{\boldsymbol{a}}_{k,t}, t). \tag{6}$$

A detailed proof is provided in Appendix C.1. Instead of relying on velocity for measuring distribution deviation, which typically enforces a roughly timestep-consistent direction, we practically use score function as a more sensitive metric for directional distribution estimation, and we can derive

Algorithm 1 Reinforcement Learning with Flow Rewards

Inputs: Online data $\mathcal{Q}\{q, a\}$, offline data $\mathcal{D}_{off}\{q, z, a\}$, reference data buffer $\mathcal{B}$, initial flow model $\boldsymbol{v}_\phi$, validated batch size κ, layer collection $\mathcal{L} = (l_1, ..., l_N)$, response quality $\mathrm{metric}(\cdot)$
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
  $\mathcal{B} \leftarrow$ Rejection-Sampling($\mathcal{G}, \mathrm{metric}$)
  while $|\mathcal{B}| > \kappa$ **do**
    Optimize flow $\boldsymbol{v}_\phi$ on batch data from $\mathcal{B}$ with policy latents in $\mathcal{L}$ using Eq. 4
    $\mathcal{B}$.pop(batch data)
  end while
end while
return policy, flow $\boldsymbol{v}_\phi$ and end training

$\|\nabla_{\dot{\boldsymbol{a}}_{k,t}} \log p_{\boldsymbol{v}_\phi}(\dot{\boldsymbol{a}}_{k,t}) - \nabla_{\dot{\boldsymbol{a}}_{k,t}} \log p_{\boldsymbol{u}}(\dot{\boldsymbol{a}}_{k,t})\|^2$ for

$$\mathcal{R}^\phi_{CFM}(\dot{\boldsymbol{a}}_k; \mathcal{T}, \mathcal{L}) = \mathbb{E}_{t\sim\mathcal{T},\tau\sim\mathcal{L}}\left[\frac{t}{1-t}\mathcal{R}^\phi_{FM}(\dot{\boldsymbol{a}}_k; t, \tau)\right], \tag{7}$$

where $\mathcal{R}^\phi_{CFM}$ 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 using different timesteps for flow reward, where larger timesteps with lower noise level are more favorable, which is consistent with suggestions from Eq. 7.

Theoretical Analysis with Likelihood. Compared to likelihood-based estimation, velocity deviations characterize directional drift toward the reference distribution. These two paradigms provide fundamentally different perspectives on distribution estimation and we further clarify their underlying relationship as:

$$\log p_{\boldsymbol{v}_\phi}(\dot{\boldsymbol{a}}_k) \geq C(\dot{\boldsymbol{a}}_k) - \lambda\mathbb{E}_{t,\tau\sim\mathcal{U}[0,1]}\left[\mathcal{R}^\phi_{\mathrm{FM}}(\dot{\boldsymbol{a}}_k; t, \tau)\right], \tag{8}$$

where $\log p_{\boldsymbol{v}_\phi}(\dot{\boldsymbol{a}}_k)$ is the log-likelihood under distribution parameterized by $\boldsymbol{v}_\phi$, $\lambda > 0$ is constant, and $C(\dot{\boldsymbol{a}}_k)$ 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 $\mathcal{R}^\phi_{\mathrm{FM}}(\dot{\boldsymbol{a}}_k; t, \tau)$ with respect to a given $\dot{\boldsymbol{a}}_k$ corresponds to the maximal evidence lower bound (ELBO) on the log-likelihood $\log p_{\boldsymbol{v}_\phi}(\dot{\boldsymbol{a}}_k)$. 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 $\boldsymbol{a}$, we shape the token-level advantage by accumulating decayed flow rewards. The advantage shaping formulation provides flexible applicability across different RLVR algorithms compared to direct reward shaping. We have:

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

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

and $\mathbb{I}[\cdot]$ 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

Table 1: Overall performance on language reasoning benchmarks. The AIME24/25 and HMMT25 are reported with Avg@16, and other benchmarks are reported with Avg@4. † means the model trained within our setting for evaluation. **Bold** indicates the best results among the comparison.

Model	In-Distribution					Out-of-Distribution			
	AIME25	AIME24	HMMT25	Minerva	**Avg.**	WinoG*	GPQA*	MusR	**Avg.**
Qwen2.5-Math-7B	9.4	13.8	2.9	55.5	20.4	51.6	25.7	23.2	33.5
ReasonFlux-PRM	13.6	25.2	3.3	**56.3**	24.6	53.8	26.6	**44.5**	41.6
Entropy Adv.†	11.6	22.8	2.4	54.1	22.7	49.2	24.4	43.2	38.9
RLVR	12.9	24.2	2.7	54.8	23.6	51.7	26.4	42.7	40.3
RLFR	**13.8**+0.9	**25.6**+1.4	**3.6**+1.1	56.1+1.3	**24.8**+1.2	**54.3**+2.6	**27.1**+0.7	44.1+1.4	**41.8**+1.5
Qwen3-4B	62.9	71.2	40.0	79.8	63.5	77.7	54.1	61.4	64.4
Entropy Adv.†	65.8	73.8	41.9	80.1	65.4	77.8	54.5	61.7	64.6
OPSD†	66.4	74.1	42.5	80.8	65.9	77.4	55.2	**61.9**	64.8
RLVR	65.3	73.1	41.7	80.9	65.2	76.8	55.6	60.9	64.4
RLFR	**66.8**+1.5	**74.9**+1.6	**44.1**+2.4	**82.7**+1.8	**67.1**+1.9	**78.2**+1.4	**56.8**+1.2	61.7+0.8	**65.6**+1.2
Qwen3-8B	66.3	74.8	41.8	82.3	66.3	81.3	59.6	62.6	67.9
Entropy Adv.†	68.4	75.9	43.1	81.9	67.5	**82.8**	62.4	64.1	69.8
OPSD†	69.3	76.1	**44.5**	84.3	68.5	82.5	62.9	64.4	69.9
RLVR	68.6	75.6	42.7	83.6	67.6	82.0	62.7	63.9	69.5
RLFR	**69.8**+1.2	**76.6**+1.0	44.2+1.5	**84.8**+1.2	**68.9**+1.3	82.6+0.6	**63.7**+1.0	**64.8**+0.9	**70.4**+0.9

$\mathrm{minmax\text{-}norm}$ 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 $\hat{\boldsymbol{a}}_{k+1}$ to serve as conditions for yielding velocity prediction in flow reward, establishing context dependence with enlarged interactive space, and more ablations are provide in Sec 4.3. The flow reward provides a stable assessment of model latents by quantifying their velocity deviation from a flow network pre-trained on high-quality data.

Updated Rewards with Rejection-Sampling. As the policy evolves throughout optimization, the induced latent distribution continuously shifts (Huan et al., 2025). As a result, a frozen flow network $\boldsymbol{v}_\phi$ pre-trained on initial offline data become increasingly misaligned with the evolving policy, leading to a growing distribution gap in flow reward estimation. To maintain alignment with evolved policy, we continuously update flow $\boldsymbol{v}_\phi$ by Eq. 4 throughout policy optimization with online rejection-samping data, where the filtered criterion can be flexible to shape the reference distribution used for flow reward computation. The overall framework is detailed in Algorithm 1. Empirically, we find correctness to be the most effective criterion, with additional ablations provided in Sec. 4.3.

4 Experiments

4.1 Experimental Setup

Training Data. We conduct experiments for both language and multimodal models for evaluation. In language setting, we use openr1 (Hugging Face, 2025) as offline data for flow pretraining, which contains 93k carefully curated mathematical reasoning problems. The reinforcement learning of RLFR is performed on DeepMath-103K (He et al., 2025b). In multimodal settings, we filter 115k mathematical subset from MMPR (Wang et al., 2024b) as offline data. The reinforcement learning is conducted on MMK12 (Meng et al., 2025), which includes mathematics, and general science.

Evaluation. We evaluate language performance on in-distributional mathematical benchmarks: AIME24/25 [1], HMMT25 (Balunović et al., 2025), and Minerva (Lewkowycz et al., 2022), and out-of-distributional reasoning benchmarks, including Winogrande (Sakaguchi et al., 2021) for commonsense reasoning, GPQA-Diamond (Rein et al., 2023) for STEM reasoning, and MuSR (Sprague et al., 2023) for multistep soft reasoning. We use a rollout temperature of 0.6, and top-p sampling with p=0.95, under a maximum response length of 32768. We report the Avg@16 metric for AIME24/25 and HMMT25, and report the Avg@4 metric for other benchmarks. For multimodal benchmarks, we include MathVista (Lu et al., 2023), MathVision (Wang et al., 2024a), MathVerse (Zhang et al., 2024), WeMath (Qiao et al., 2024) for in-distributional evaluation, and LogicVista (Xiao et al., 2024) and VisuLogic (Xu et al., 2025) for out-of-distributional evaluaiton. We report Pass@1 metric with temperature of 0 for greedy decoding, under a maximum response length of 16384.

[1] https://huggingface.co/datasets/AI-MO/aimo-validation-aime

Table 2: Overall performance on multimodal benchmarks. Pass@1 metric is reported. † means the model trained within our setting for evaluation. **Bold** indicates the best results among the comparison.

Model	In-Distribution					Out-of-Distribution		
	MathVista	MathVision	MathVerse	WeMath	**Avg.**	LogicVista	VisuLogic	**Avg.**
Qwen2.5VL-7B-Inst	65.6	23.4	47.3	53.4	47.4	47.8	**27.1**	37.5
R1-OneVision-7B	63.7	22.4	45.2	52.9	46.1	38.9	18.3	28.6
OpenVLThinker-7B	64.5	24.3	46.1	50.3	46.3	38.7	10.6	24.6
MM-Eureka-7B	73.5	27.9	51.9	58.7	53.0	46.9	25.5	36.2
Entropy Adv. †	71.1	26.3	**52.1**	63.1	53.2	44.5	25.7	35.1
RLVR	73.1	27.6	48.0	64.6	53.3	48.3	24.8	36.5
RLFR	**73.4**+0.3	**31.6**+4.0	52.0+4.0	**66.1**+1.5	**55.8**+2.4	**48.3**+0.0	26.7+1.9	**37.5**+1.0
Qwen3VL-4B-Inst	73.7	36.5	46.8	70.1	56.8	53.2	25.6	39.4
Entropy Adv. †	75.7	42.3	52.6	78.3	62.2	54.4	25.8	40.1
RLVR	75.3	40.2	51.4	77.4	61.1	54.7	26.2	40.5
RLFR	**76.2**+0.9	**44.6**+4.4	**53.8**+2.4	**78.9**+1.5	**63.4**+2.3	**55.3**+0.6	**27.5**+1.3	**41.4**+0.9

Implementation Details. 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 $\mathcal{L} = \{0.25, 0.5, 0.75\}$ throughout the model, where the layer position embedding are added. The flow network comprises 4 layers for 4B models and 6 layers for 7B/8B models. In reinforcement learning, we exclude both KL divergence and entropy regularization, 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 kept consistent with the evaluation setting. We use the threshold η of 0.6 to discard noisy fluctuations, the validated batch size κ of 32 for flow update. The discount factor γ and coefficient β are set as 1 and 0.01. Timestep collection for yielding flow reward is set for $\mathcal{T} = \{0.8\}$, and we use the temperature of 1 for response rollout.

4.2 Main Results

The main experimental results in Table 1 and Table 2 show that RLFR consistently outperforms a broad range of baselines across both language and multimodal reasoning benchmarks. The comparison include vanilla GRPO (Shao et al., 2024b), process reward modeling (PRM) baseline (Zou et al., 2025), token-level optimization baselines (Cheng et al., 2025; Zhao et al., 2026), and advanced methods (Yang et al., 2025c; Deng et al., 2025; Meng et al., 2025). Table 1 shows that, on Qwen3-4B, RLFR outperforms RLVR by 1.9% on mathematical and 1.2% on general reasoning benchmarks, respectively. And the improvements on Qwen3-8B further demonstrate the effectiveness and stability of the flow reward. Additionally, compared with dense reward signals derived from the logit space, such as Entropy Adv. (Cheng et al., 2025), and OPSD (Zhao et al., 2026), the superior performance of RLFR highlights the promising latent space signals for policy optimization.

Table 3: Efficiency for training time comparison.

	RLVR	ReasonFlux-PRM	RLFR
Training Time (h)	12.38	21.53	12.62

Table 4: Non-verifiable task performance.

	CreativeWritingV3	AlpacaEval2 LC
Qwen3-4B	53.6	48.2
RLFR-Qwen3-4B	56.7	51.6

Table 2 present the results on multimodal benchmarks, where RLFR consistently improves over RLVR with 2.3% on mathematical and 0.9% on logic reasoning benchmarks with Qwen3VL-4B. In Table 3, we show that RLFR provides substantially more efficient process reward estimation compared to computationally intensive PRM baseline, while maintaining competitive performance as reported in Table 1. And the overall training efficiency is even comparable with RLVR methods.

Non verifiable tasks. To assess whether flow rewards can generalize to non-verifiable tasks, we follow the RLMT setting (Bhaskar et al., 2025), using 6K SFT samples to initialize the flow network offline and 7.8K RL prompts for reinforcement learinng. We evaluate the resulting policy on CreativeWritingV3 (Paech, 2025) and AlpacaEval2 (Dubois et al., 2024). The results show that RLFR remains effective where explicit verifiable rewards are unavailable with solely flow rewards.

4.3 Ablations

To validate the reliability of RLFR, we conduct ablations on Qwen3-4B, including the framework components, online rejection-sampling metric for shaping reference distribution, effectiveness of timestep debiasing for flow reward estimation, and the enlarged context used for velocity prediction.

Steps	Type	Frequently Reward Tokens
25	+	frac theta +-={} sqrt cdot matrix times abs pi math symbol sin line product set angle...
	-	To output of need equation problem value these many how determine import break maximum...
286	+	frac sqrt area times cdot average triangle sin equality outer second integers length sum...
	-	To problem output determine understand python frac denote Given integers use properties...

Table 5: Case study of reward tokens in training progress. "+" means positive flow reward and "-" means negative flow reward.

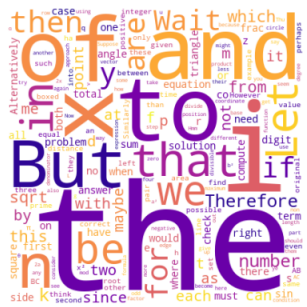


Figure 4: Textual cloud of offline warmup dataset.

Effect of online rejection sampling and corresponding metrics. We present the ablations of RLFR framework in Table 6, where both offline warmup of flow network and continuously online updated with rejection sampling data show steady contribution to the final performance. Additionally, noisy fluctuations in the flow reward can destabilize policy optimization, making the filtering of large reward deviations important for stable RL training. In Table 7, we investigate the effects of different metrics in rejection sampling that shape the desired reference distribution, including outcome correctness, average trajectory entropy with 50th percentile threshold, and their combination. The correctness is still validated to be the most effective criterion on filtering the reference data.

Table 6: Ablation results of key components in flow reward construction.

Method	HMMT25	GPQA-d
RLFR	**44.1**	**56.8**
w/o offline warmup	40.8-3.3	54.4-2.4
w/o online updated	42.3-1.8	55.2-1.6
w/o reward filtering	43.2-0.9	55.8-1.0

Table 7: Ablation results of metrics used in rejection sampling that contribute to the online updated reference distribution for flow reward.

Metric	HMMT25	GPQA-d
Outcome correctness	**44.1**	**56.8**
Trajectory entropy	43.5	55.9
Correctness+Entropy	43.2	56.3

Timesteps debiasing in flow reward. The flow reward derived from velocity deviations delivers several impact factors, including timesteps used for velocity prediction and conditions in flow network, and we provide their effect on reward quality in Figure 5 and Table 8. The results in Figure 5 provide strong evidence for timesteps debiasing method suggested in Sec. 3.2, where the larger timesteps with less noisy are preferred for flow reward, as their more reliable velocity prediction, and when using collection of timesteps such as $\mathcal{T} = \{0.2, 0.4, 0.6, 0.8\}$ for calculating velocity deviation, the debiasing weighting further shows superiority by 1.3% improvement compared to basic average.

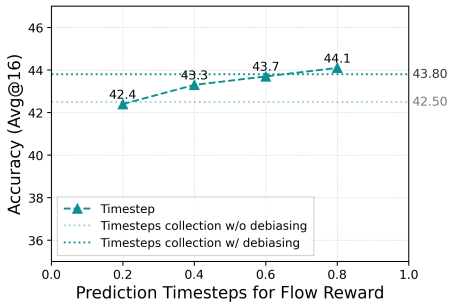


Figure 5: Results on different timesteps for flow reward and debiasing effect.

Effect of context conditions in velocity prediction. We provide results with different context conditions for velocity prediction in constructing flow reward for $\boldsymbol{a}_k$ in Table 8, including latents from identity token or neighbor tokens, where $\dot{\boldsymbol{a}}_{k+1}$ exhibits clear advantages over the other two paradigms, validating that richer context in velocity prediction yields high-quality reward, attributing to efficient context accumulation in latent space.

Table 8: Ablations results of context conditions in flow network for velocity prediction.

Condition	Position	Math Avg.	Logic Avg.
$\dot{\boldsymbol{a}}_k$	Identity	**44.5**	56.1
$\dot{\boldsymbol{a}}_{k-1}$	Previous	43.6	55.8
$\dot{\boldsymbol{a}}_{k+1}$	Post	44.1	**56.8**

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 5, we found that: **(1)** In contrast to previous entropy-based methods that reinforce tokens serving as logical connectors in steering the reasoning

process, the flow rewards prefer tokens that effectively contribute to solving 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 warmup 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., 2025; 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., 2025; 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., 2025a; 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 and Future Work

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 perspective on shaping RLVR with flow reward, where the flow field of model latents 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 utilization. RLFR first demonstrates that a well-established flow field can be a sound environment for reward signal collection, yielding steady performance improvements across both language and multimodal benchmarks, highlighting the potential of latent substrate for reward design. Additionally, RLFR naturally leverages expert reasoning trajectories from off-policy data into the constitution of reward signal, instead of relying on self-confidence. Future directions include scaling the flow environment to release the latent potential, and the prospect of latent signals for test-time scaling.

AI USE STATEMENT

In this work, we did not use generative AI tools for research ideation, methodology development, experimental design, result interpretation, or implementation of the core research code. We used generative AI tools only for language polishing and to assist with writing a small portion of the plotting code. All AI-assisted text was reviewed and revised by the authors, and the generated plotting code was manually inspected and verified against the underlying experimental results. The core research ideas, methods, experiments, and implementation were developed independently by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025.

Adithya Bhaskar, Xi Ye, and Danqi Chen. Language models that think, chat better, 2025.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint arXiv:2507.16806*, 2025.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *arXiv preprint arXiv:2503.17352*, 2025.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning. *arXiv preprint arXiv:2506.19767*, 2025.

Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin P Murphy, and Tim Salimans. Diffusion meets flow matching: Two sides of the same coin. 2024. *URL https://diffusionflow. github. io*, 2024.

Hasan Abed Al Kader Hammoud, Hani Itani, and Bernard Ghanem. Beyond the last answer: Your reasoning trace uncovers more than you think. *arXiv preprint arXiv:2504.20708*, 2025.

Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*, 2025a.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025b.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL `https://github.com/huggingface/open-r1`.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025a.

Yi-Chen Li, Tian Xu, Yang Yu, Xuqin Zhang, Xiong-Hui Chen, Zhongxiang Ling, Ningjing Chao, Lei Yuan, and Zhi-Hua Zhou. Generalist reward models: Found inside large language models. *arXiv preprint arXiv:2506.23235*, 2025b.

Yizhuo Li, Yuying Ge, Yixiao Ge, Ying Shan, and Ping Luo. Aligning latent spaces with flow priors. *arXiv preprint arXiv:2506.05240*, 2025c.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025a.

Mingjie Liu, Shizhe Diao, Jian Hu, Ximing Lu, Xin Dong, Hao Zhang, Alexander Bukharin, Shaokun Zhang, Jiaqi Zeng, Makesh Narsimhan Sreedhar, et al. Scaling up rl: Unlocking diverse reasoning in llms via prolonged training. *arXiv preprint arXiv:2507.12507*, 2025b.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025c.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*, 2025d.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

Samuel J Paech. Eq-bench creative writing benchmark v3. `https://github.com/EQ-bench/creative-writing-bench`, 2025.

Samuel Pfrommer, Yixiao Huang, and Somayeh Sojoudi. Reinforcement learning for flow-matching policies. *arXiv preprint arXiv:2507.15073*, 2025.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024a. URL `https://arxiv.org/abs/2402.03300`.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *arXiv preprint arXiv:2310.16049*, 2023.

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr. *arXiv preprint arXiv:2507.15778*, 2025a.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024a.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025b.

Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, et al. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024b.

Zihao Wang, Chirag Nagpal, Jonathan Berant, Jacob Eisenstein, Alex D'Amour, Sanmi Koyejo, and Victor Veitch. Transforming and combining rewards for aligning large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024c.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.

Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, et al. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. *arXiv preprint arXiv:2504.15279*, 2025.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Wenkai Yang, Jingwen Chen, Yankai Lin, and Ji-Rong Wen. Deepcritic: Deliberate critique with large language models. *arXiv preprint arXiv:2505.00662*, 2025b.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025c.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through rl training. *arXiv preprint arXiv:2509.03403*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, et al. Genprm: Scaling test-time compute of process reward models via generative reasoning. *arXiv preprint arXiv:2504.00891*, 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2506.18896*, 2025.

A Experimental Details

A.1 Settings

Experiments are conducted on 8 H200 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. 9, where we adopt bbox template for language training and tag template for multimodal training. In evaluation, we deploy a Qwen3-32B model server for answer extraction and judge.

RLFR training prompt **Bbox**

```
<|im_start|>system
Please reason step by step, and put your final answer within \\boxed{}.
<|im_end|>
<|im_start|>user
{{question}}<|im_end|>
<|im_start|>assistant
```

RLFR training prompt **Tag**

```
<|im_start|>system
You should first thinks 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 $...$. The reasoning process and answer are
enclosed within <think> </think> and <answer> </answer> tags,
respectively, i.e., <think> Since $1+1=2$, so the answer is $2$.
</think> <answer> $2$ </answer>, which means your output should start
with <think> and end with </answer>.
<|im_end|>
<|im_start|>user
{{question}}<|im_end|>
<|im_start|>assistant
```

Table 9: 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.

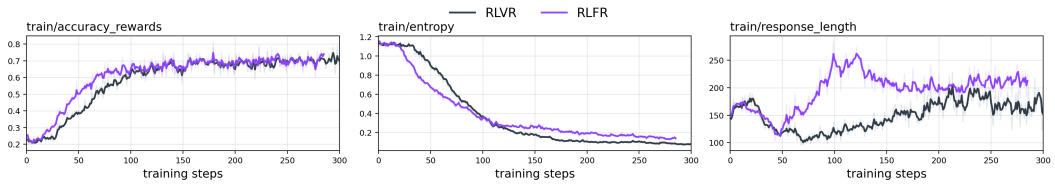
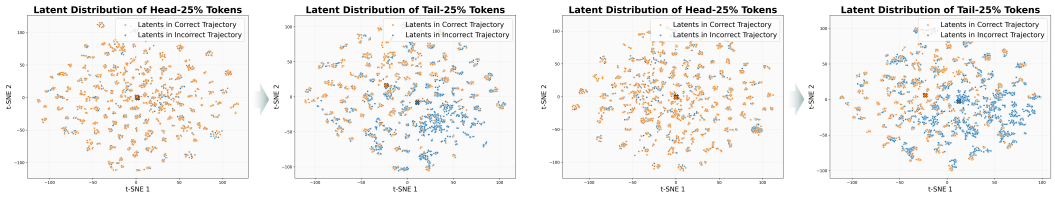


Figure 6: Training logs of **RLVR** and **RLFR** on Qwen3VL-4B.

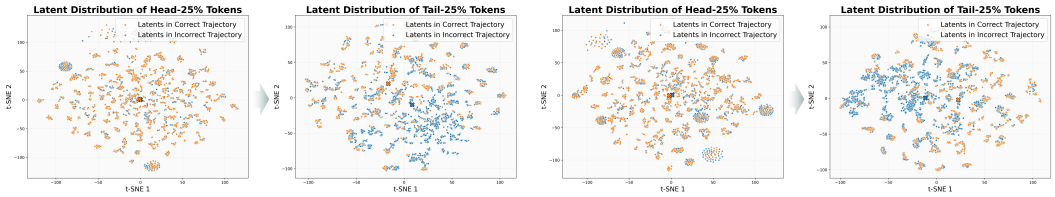
B Extended Analysis of Latent Space

Sec. 3.1 analysis the latent space of Qwen3-4B 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 the latent distributions of reasoning trajectory tokens at the $\{25, 50, 75, 100\}$ layer percentiles, and found that



(a) Distribution of trajectory tokens in latent space with layer percentail-25.

(b) Distribution of trajectory tokens in latent space with layer percentail-50.



(c) Distribution of trajectory tokens in latent space with layer percentail-75.

(d) Distribution of trajectory tokens in latent space with layer percentail-100.

Figure 7: Distribution of reasoning trajectory tokens in latent space across different layer percentiles of the Qwen3-4B, which consistently show expressive signals on tail trajectory tokens, highlighting the broader potential of latent space for reward signal collection.

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.

We consider the linear interpolation

$$\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \beta_t \boldsymbol{x}_1, \qquad x_0 \sim \mathcal{N}(0, I), \;\; x_1 \sim p_{\text{data}},$$

where $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ 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 $\boldsymbol{x}_1$, the noisy variable $\boldsymbol{x}_t$ is Gaussian:

$$p_t(\boldsymbol{x}_t \mid \boldsymbol{x}_1) \;\sim\; \mathcal{N}\big(\beta_t \boldsymbol{x}_1,\; \alpha_t^2 I\big),$$

with conditional score

$$\nabla_x \log p(\boldsymbol{x}_t \mid \boldsymbol{x}_1) = -\frac{\boldsymbol{x}_t - \beta_t \boldsymbol{x}_1}{\alpha_t^2}. \tag{11}$$

By Fisher's identity, the marginal score given by

$$\begin{aligned} \boldsymbol{s}_t(x) &= \nabla_x \log p_t(x) = \mathbb{E}[\,\nabla_x \log p(\boldsymbol{x}_t \mid \boldsymbol{x}_1) \mid \boldsymbol{x}_t = x] \\ &= -\frac{1}{\alpha_t^2}\Big(x - \beta_t\, \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t = x]\Big). \end{aligned} \tag{12}$$

Rearranging yields

$$\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t = x] = \frac{1}{\beta_t}\Big(x + \alpha_t^2 \boldsymbol{s}_t(x)\Big). \tag{13}$$

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

$$\begin{aligned} \boldsymbol{v}_t(x) &= \mathbb{E}\Big[\dot{\alpha}_t \boldsymbol{x}_0 + \dot{\beta}_t \boldsymbol{x}_1 \;\Big|\; \boldsymbol{x}_t = x\Big] \\ &= \dot{\alpha}_t \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t = x] + \dot{\beta} \mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t = x]\,. \end{aligned} \tag{14}$$

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

$$\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t = x] = \frac{x}{\alpha_t} - \frac{\beta_t}{\alpha_t}\,\mathbb{E}[\boldsymbol{x}_1 \mid \boldsymbol{x}_t = x]. \tag{15}$$

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

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

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

$$\boldsymbol{v}_t(x) = \frac{1}{t}\,x \;+\; \frac{1-t}{t}\,\boldsymbol{s}_t(x). \tag{17}$$

The score function is given by

$$\boldsymbol{s}_t(x) = -\frac{x}{1-t} \;+\; \frac{t}{1-t}\,\boldsymbol{v}_t(x). \tag{18}$$

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

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

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

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

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

$$\partial_t \log p_{\boldsymbol{v}_\phi}(\boldsymbol{y}_t) = -\nabla \cdot \boldsymbol{v}_\phi(\boldsymbol{y}_t, t) - \boldsymbol{v}_\phi(\boldsymbol{y}_t, t) \cdot \nabla_y \log p_{\boldsymbol{v}_\phi}(\boldsymbol{y}_t). \tag{20}$$

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

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

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

$$\begin{aligned}\log p_{\boldsymbol{v}_\phi}(\boldsymbol{y}) &= \log p_{\text{init}}(\boldsymbol{x}_0) - \int_0^1 \nabla \cdot \boldsymbol{v}_\phi(\boldsymbol{y}_t, t)\,dt \\ &= \mathbb{E}_{x_0\sim p_{\text{init}}}\big[\log p_{\text{init}}(\boldsymbol{x}_0)\big] - \int_0^1 \mathbb{E}_{y\sim p_{\text{data}}}\big[\nabla \cdot \boldsymbol{v}_\phi(\boldsymbol{y}_t, t)\big]\,dt \\ &= C_0(x) - \int_0^1 \mathbb{E}_{y\sim p_{\text{data}}}\big[\langle \boldsymbol{v}_\phi(\,\boldsymbol{y}_t, t),\ \nabla \log p_{\text{data}}(\,\boldsymbol{y}_t\,)\rangle\big]\,dt\end{aligned} \tag{22}$$

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

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

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

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

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

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

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

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

Considering the Fenchel–Young inequality

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

where we substitute into Eq. 23, and havex

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

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

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

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

D Case Study

Question and Ground Truth

Question: Let $ABCD$ be a rectangle with $BC = 24$. Point X lies inside the rectangle such that

$$\angle AXB = 90^\circ.$$

Given that triangles $\triangle AXD$ and $\triangle BXC$ are both acute and have circumradii 13 and 15, respectively, compute AB.
Ground Truth: $14 + 4\sqrt{37}$

Response from RLVR Baseline

To solve the problem, we consider the geometry of the rectangle $ABCD$ with $BC = 24$, and a point X inside the rectangle such that $\angle AXB = 90^\circ$. Additionally, triangles $\triangle AXD$ and $\triangle BXC$ are both **acute** and have **circumradii** 13 and 15 respectively. We are to find the length of side AB, which we'll denote as a.

Step 1: Coordinate Setup

We place the rectangle in the coordinate plane:

- Let $A = (0, 0)$, $B = (a, 0)$, $C = (a, 24)$, $D = (0, 24)$.
- Let $X = (x, y)$, where $0 < x < a$, $0 < y < 24$.

Since $\angle AXB = 90^\circ$, the vectors $\vec{XA} = (-x, -y)$ and $\vec{XB} = (a - x, -y)$ are perpendicular, so their dot product is zero:

$$(-x)(a - x) + (-y)(-y) = 0 \Rightarrow x^2 + y^2 = ax \tag{1}$$

Step 2: Circumradius of Triangle $\triangle AXD$

Triangle AXD has vertices $A(0, 0)$, $X(x, y)$, $D(0, 24)$. The area of the triangle is:

$$\text{Area} = \frac{1}{2} \cdot \text{base} \cdot \text{height} = \frac{1}{2} \cdot 24 \cdot x = 12x$$

The side lengths are:

- $AX = \sqrt{x^2 + y^2} = \sqrt{ax}$ (from equation (1))
- $XD = \sqrt{x^2 + (y - 24)^2}$
- $AD = 24$

Using the formula for the circumradius $R = \frac{abc}{4\Delta}$, we get:

$$R_{AXD} = \frac{AX \cdot XD \cdot AD}{4 \cdot \text{Area}} = \frac{\sqrt{ax} \cdot \sqrt{x^2 + (y - 24)^2} \cdot 24}{4 \cdot 12x} = \frac{\sqrt{ax} \cdot \sqrt{x^2 + (y - 24)^2}}{2x}$$

Set this equal to 13:

$$\frac{\sqrt{ax} \cdot \sqrt{x^2 + (y - 24)^2}}{2x} = 13 \Rightarrow \sqrt{ax} \cdot \sqrt{x^2 + (y - 24)^2} = 26x \tag{2}$$

Step 3: Circumradius of Triangle $\triangle BXC$

Similarly, triangle BXC has vertices $B(a, 0)$, $X(x, y)$, $C(a, 24)$. The area is:

$$\text{Area} = \frac{1}{2} \cdot 24 \cdot (a - x) = 12(a - x)$$

Side lengths:

- $BX = \sqrt{(x - a)^2 + y^2} = \sqrt{a(a - x)}$
- $XC = \sqrt{(x - a)^2 + (y - 24)^2}$
- $BC = 24$

Using the same formula:

$$R_{BXC} = \frac{BX \cdot XC \cdot BC}{4 \cdot \text{Area}} = \frac{24 \cdot \sqrt{a(a - x)} \cdot \sqrt{(x - a)^2 + (y - 24)^2}}{48(a - x)}$$

Set this equal to 15:

$$\frac{\sqrt{a(a - x)} \cdot \sqrt{(x - a)^2 + (y - 24)^2}}{2(a - x)} = 15 \Rightarrow \sqrt{a(a - x)} \cdot \sqrt{(x - a)^2 + (y - 24)^2} = 30(a - x) \tag{3}$$

Response from RLVR Baseline

Step 4: Solving the System
From equations (2) and (3), we can express x in terms of a using geometric and algebraic manipulations. After simplification and substitution, we derive a cubic equation in a:

$$a^3 - 38a^2 - 116a + 3960 = 0$$

Annotation: The equations (2) and (3) only determine the sine of the relevant angles and do not distinguish between the acute and obtuse branches. The condition that both triangles are acute is necessary to obtain $\cos\angle AXD = \frac{5}{13}$ and $\cos\angle BXC = \frac{3}{5}$. Without using this condition, the subsequent cubic equation is not justified and may include configurations that do not satisfy the problem's geometric constraints.

Solving this, we find two real roots:

$$a = 10 \quad \text{and} \quad a = 14 + 4\sqrt{37}$$

However, $a = 10$ leads to $x = 10$, which places point X at point B, contradicting the condition that X is **inside** the rectangle. Therefore, the only valid solution is:

$$\boxed{14 + 4\sqrt{37}}$$

This is the length of side AB, satisfying all conditions of the problem.
Final Answer

$$\boxed{14 + 4\sqrt{37}}$$

Response from RLFR

To solve the problem, we are given a rectangle $ABCD$ with $BC = 24$, and a point X inside the rectangle such that $\angle AXB = 90^\circ$. Triangles $\triangle AXD$ and $\triangle BXC$ are both **acute**, and their **circumradii** are 13 and 15, respectively. We are to find the length $AB = a$.
Step 1: Coordinate Setup
Let's place the rectangle in the coordinate plane as follows:

- $A = (0, 0)$
- $B = (a, 0)$
- $C = (a, 24)$
- $D = (0, 24)$

Let $X = (x, y)$, with $0 < x < a$, $0 < y < 24$.
Step 2: Use of Right Angle at X
Since $\angle AXB = 90^\circ$, triangle AXB is right-angled at X, so by the **circle with diameter** AB, point X lies on the circle centered at the midpoint of AB with radius $a/2$. The equation of this circle is:

$$\left(x - \frac{a}{2}\right)^2 + y^2 = \left(\frac{a}{2}\right)^2 \Rightarrow x^2 - ax + y^2 = 0$$

So, we have the identity:

$$x^2 + y^2 = ax \tag{1}$$

Step 3: Use of Circumradius Conditions
The **circumradius** R of a triangle with sides a, b, c and area Δ is given by:

$$R = \frac{abc}{4\Delta}$$

We use this for both triangles:

Response from RLFR

Triangle $\triangle AXD$:

- Sides: $AX = \sqrt{x^2 + y^2} = \sqrt{ax}$, $XD = \sqrt{x^2 + (y-24)^2}$, $AD = 24$
- Area $\Delta_1 = \frac{1}{2} \cdot 24 \cdot x = 12x$

So the circumradius is:

$$R_1 = \frac{\sqrt{ax} \cdot \sqrt{x^2 + (y-24)^2} \cdot 24}{4 \cdot 12x} = \frac{\sqrt{ax} \cdot \sqrt{x^2 + (y-24)^2}}{2x}$$

Set this equal to 13:

$$\frac{\sqrt{ax} \cdot \sqrt{x^2 + (y-24)^2}}{2x} = 13 \Rightarrow \sqrt{ax} \cdot \sqrt{x^2 + (y-24)^2} = 26x \tag{2}$$

Triangle $\triangle BXC$:

- Sides: $BX = \sqrt{(a-x)^2 + y^2}$, $XC = \sqrt{(a-x)^2 + (y-24)^2}$, $BC = 24$
- Area $\Delta_2 = \frac{1}{2} \cdot 24 \cdot (a-x) = 12(a-x)$

So the circumradius is:

$$R_2 = \frac{\sqrt{(a-x)^2 + y^2} \cdot \sqrt{(a-x)^2 + (y-24)^2}}{2(a-x)}$$

Set this equal to 15:

$$\begin{aligned} &\frac{\sqrt{(a-x)^2 + y^2} \cdot \sqrt{(a-x)^2 + (y-24)^2}}{2(a-x)} = 15, \\ \Rightarrow \quad &\sqrt{(a-x)^2 + y^2} \cdot \sqrt{(a-x)^2 + (y-24)^2} = 30(a-x). \end{aligned} \tag{3}$$

Step 4: Use of Law of Cosines and Law of Sines
Using the **Law of Cosines** and **Law of Sines**, we find that the angle at X in triangle AXD has $\sin(\theta) = \frac{12}{13}$, and the angle at X in triangle BXC has $\sin(\phi) = \frac{4}{5}$. These lead to the following identities:

- From triangle AXD: $x(a-10) = 24y \Rightarrow y = \frac{x(a-10)}{24}$
- From triangle BXC: $(2a-36)(a-x) = 48y$

Substituting y from the first into the second, we derive:

$$(2a-36)(a-x) = 2x(a-10)$$

Solving this and using equation (1), we eventually arrive at the **cubic equation**:

$$a^3 - 38a^2 - 116a + 3960 = 0$$

Step 5: Solve the Cubic Equation
This cubic has one rational root: $a = 10$, but it leads to $x = a = 10$, which is on the boundary (not inside the rectangle). Therefore, we reject it.
The other root is:

$$a = 14 + 4\sqrt{37}$$

This is the only valid solution that satisfies all conditions, including the **acute triangle** requirement.
Final Answer

$$\boxed{14 + 4\sqrt{37}}$$