VANE: Guiding High-Value Exploration in RLVR via Outcome-Process Novelty Shaping

Xu He[1], Jialiang Guo[1], Fucheng Xiong[1], Haodong Zhao[1*],
Xingyang Li[1*], Ke Zeng[1], Xunliang Cai[1],
[1]Meituan,
Correspondence: hexu16@126.com, zhaohaodong04@meituan.com, lixingyang04@meituan.com

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) frequently suffers from mode collapse due to the inherent sparsity of feedback signals. While strategies such as entropy regularization introduce randomness, they lack directionality. Simply incorporating diversity rewards is overly one-sided and fails to identify potential logical errors or hallucinations. To address these limitations, we propose VANE (Value-Aligned Novelty Exploration), a method that simultaneously quantifies novelty across the outcome space (via reward or solution divergence) and the semantic process space (via semantic process divergence). Moreover, VANE employs a value-alignment mechanism that symmetrically amplifies scarce, high-quality solutions while explicitly penalizing diverse yet erroneous reasoning paths. Extensive experiments on models such as Qwen2.5-Math-7B across eight benchmarks—encompassing both large-scale mathematical reasoning and out-of-distribution (OOD) tasks—demonstrate the effectiveness and generalization of the proposed method.

1 Introduction

The rapid evolution of Large Language Models (LLMs), has been significantly driven by Reinforcement Learning with Verifiable Rewards (RLVR) (Lambert et al., 2024). By optimizing against deterministic signals such as correct answers, algorithms like GRPO (Shao et al., 2024) efficiently guide models toward high-performance solutions (Guo et al., 2025a). However, this outcome-reward-dependent approach has a limitation: it cannot provide feedback on the reasoning process itself, causing models to collapse into safe but limited reasoning patterns and undermines the overall performance (Yu et al., 2025; Cui et al., 2025a).

In reinforcement learning, generating diverse outputs is pivotal for exploring the action space and

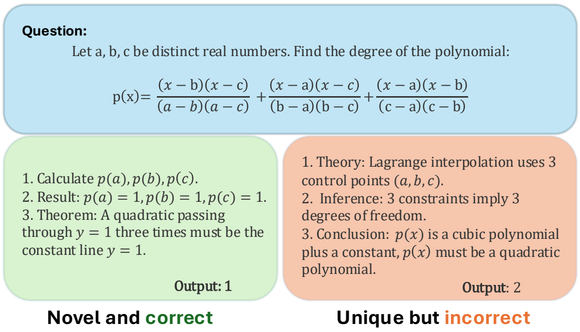


Figure 1: Two responses to a math problem: one deriving the correct solution via a novel approach, while the other reaching an incorrect result with flawed logic.

discovering novel strategies (Cheng et al., 2025; An et al., 2025). Some studies have attempted methods such as entropy regularization (Tiapkin et al., 2024) or decoupled KL divergence (Slocum et al., 2025) to enhance the model's exploration capability. More recently, other studies have tried adding diversity incentives via reward shaping (Li et al., 2025b; He et al., 2025). However, these works often focus narrowly on outcome or trajectory diversity. Moreover, they treat all incorrect samples indiscriminately. This approach leads to passive ignorance: by assigning an identical zero reward to both minor errors and severe hallucinations, such methods fail to distinguish between distinct failure modes. In mathematical reasoning, unique paths deviating from group consensus that lead to errors often signify severe logical fallacies (Lightman et al., 2023; Kuhn et al., 2023). For instance, Figure 1 illustrates a case where the model falsely infers a cubic polynomial from three control points, only to contradictorily conclude the function is quadratic. Without explicit penalties for such novel errors, the model will explore invalid regions.

To address these limitations, we propose the Value-Aligned Novelty Exploration (VANE). Unlike prior methods that focus on a single dimension or passively ignore errors, VANE first estab-

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 28721–28739

lishes a comprehensive novelty metric by synthesizing outcome and semantic statistics. Specifically, we quantify Outcome-Level Novelty by measuring the deviation of a sample's reward from the group mean (RSD) or the sparsity of its discrete answer frequency (SSS), and complementarily assess Semantic-Level Novelty via Semantic Process Divergence (SPD), which calculates the geometric distance of a reasoning trajectory from the group's embedding centroid. These metrics are aggregated into a unified Hybrid Novelty Score through a linear combination. Subsequently, we apply a Value-Aligned Symmetric Shaping mechanism to this score. This mechanism functions as a directional filter that symmetrically modulates the novelty incentive—amplifying it for correct solutions while inverting it into a penalty for errors. Finally, this shaped value is superimposed onto the original raw reward, ensuring that the model pursues diversity that is strictly anchored in correctness.

To evaluate VANE, we structure our experiments around two primary dimensions. First, we examine the performance frontiers, assessing whether VANE achieves superior accuracy and generalization across in-distribution (ID) and out-of-distribution (OD) tasks while maintaining robustness across varying model scales. Second, we validate the methodological design through ablation studies and sensitivity analyses, specifically isolating the contributions of outcome versus semantic modules and verifying the role of symmetric shaping in preventing ungrounded exploration. our contributions are as follows:

- We propose VANE, which quantifies novelty by integrating outcome statistics with process semantics from dual perspectives and employs a symmetric shaping mechanism to targetedly assign rewards or penalties to diverse samples based on their validity.

- We empirically demonstrate that imposing explicit penalties on incorrect yet diverse rollout samples outperforms the indiscriminate treatment of errors, resulting in models that exhibit superior performance.

- We validate the effectiveness of our approach through extensive experiments across eight benchmarks involving multiple model families. The results confirm that the proposed method achieves consistent performance gains over baselines, demonstrating superior generalization in both ID and OOD tasks.

2 Related Work

2.1 Reward Shaping

Reinforcement Learning with Verifiable Rewards (RLVR) (Lambert et al., 2024; Team et al., 2025) has emerged as a key paradigm for enhancing LLM reasoning by utilizing deterministic signals from math (Yang et al., 2024; Guo et al., 2025a) and coding (Liu and Zhang, 2025; Luo et al., 2025), etc. Yet, its reliance on sparse binary outcomes hinders efficient optimization (Yu et al., 2025). To address this, reward shaping is employed to densify feedback and guide models toward desired solutions (Goyal et al., 2019; Gupta et al., 2022; Xie et al., 2023). Liao et al. (2025); Liu et al. (2025c); Guo et al. (2025b) combined rule-based verifiers with learned reward models to generate more informative signals. Other studies have integrated multi-level rule-based rewards, including format (Xin et al., 2025; Li et al., 2025b), length (Liu et al., 2025a), and outcome (Anschel et al., 2025) components, to facilitate long Chain-of-Thought (CoT) reasoning (Guo et al., 2025a) and mitigate output anomalies. Despite these advancements, there remains a lack of suitable, multidimensional metrics for quantifying and enhancing reasoning creativity.

2.2 Diversity in Large Language Models

Language models are prone to generating repetitive or low-diversity outputs (Li et al., 2016; Holtzman et al., 2019; Zhang et al., 2021). To address this, existing strategies encompass optimizations across both inference and training stages. During inference, early approaches relied on stochastic sampling techniques, such as temperature scaling (ACKLEY et al., 1985; Peeperkorn et al., 2024), Top-k sampling (Fan et al., 2018), and Nucleus sampling (Holtzman et al., 2019), to introduce randomness. To internalize diversity during optimization, researchers have integrated distribution-level constraints into training objectives, employing techniques like Maximizing Mutual Information (Bowman et al., 2016; Li et al., 2016) and unlikelihood training (Papineni et al., 2002) to enforce broader distributional coverage. In the realm of online Reinforcement Learning, methods such as decoupled KL regularization (Slocum et al., 2025), decoupled clipping (Yu et al., 2025), and entropy regularization (Tiapkin et al., 2024; Hou et al., 2025; Lanchantin et al., 2025; Cheng et al., 2025) have garnered significant attention. Additionally, some studies have attempted to incorporate diversity-related incentives via reward shaping (Li et al., 2025b; Anschel et al., 2025; Li et al., 2025a); however, these approaches are mostly limited to a single dimension and focus on the diversity of correct samples only.

3 Method

We formulate the RLVR task as aligning a policy $\pi_\theta(y|x)$ to generate responses y for prompts x based on deterministic rewards. While algorithms like GRPO efficiently optimize for correctness, their focus on exploitation frequently leads to mode collapse. To address this, we introduce Value-Aligned Novelty Exploration (VANE). This method restores exploration by first quantifying novelty via outcome and semantic metrics (Section 3.1), then applying symmetric shaping to strictly couple diversity with validity (Section 3.2), and finally incorporating these signals into the policy update (Section 3.3).

3.1 Multi-view Novelty Quantification

To capture exploration behavior, we decompose novelty into two dimensions: the Outcome Space, which focuses on the uniqueness of the final result, and the Semantic Space, which evaluates the diversity of the reasoning process. This dual-view formulation allows us to distinguish between discovering new solutions and finding alternative derivation paths for existing solutions.

3.1.1 Outcome-Level Novelty ($\mathcal{N}_{out}$)

We posit that a highly novel outcome manifests as statistical rarity, which can be identified either by its deviation from the group's average raw reward performance or by its low frequency within the solution set. Accordingly, we propose two strategies to capture this sparsity.

Strategy I: Reward-Space Divergence (RSD). Drawing inspiration from the advantage formulation in GRPO (Shao et al., 2024), we design RSD based on the fact that unique samples in the output results yield original rewards that are significantly higher or lower than the average level of the same group. Let $\mathcal{R} = \{R_1, \ldots, R_G\}$ be the set of raw rewards. The divergence is calculated as the standardized absolute deviation:

$$\mathcal{N}_{out}^{RSD}(y_i) = \frac{|R_i - \mu_{\mathcal{R}}|}{\sigma_{\mathcal{R}} + \epsilon}, \tag{1}$$

where $\mu_{\mathcal{R}}$ and $\sigma_{\mathcal{R}}$ denote the group mean and standard deviation, respectively. Consequently, RSD is driven by the distribution of validity within the group: if a correct (or incorrect) outcome is sparse, its reward will deviate significantly from the group mean $\mu_{\mathcal{R}}$, thereby yielding a larger RSD value.

Strategy II: Solution-Space Sparsity (SSS). Alternatively, for tasks with discrete answers, we define novelty based on frequency. Let o_i be the extracted answer for response y_i. We compute sparsity as:

$$\mathcal{N}_{out}^{SSS}(y_i) = 1 - \frac{1}{G}\sum_{j=1}^{G} \mathbb{I}(o_j = o_i). \tag{2}$$

This metric assigns scores approaching 1 for unique solutions and near 0 for frequent ones, directly incentivizing the generation of rare answers.

3.1.2 Semantic-Level Novelty ($\mathcal{N}_{sem}$)

We design the Semantic Process Divergence (SPD) based on the principle that a semantically novel trajectory should exhibit a significant geometric distance from the group's semantic center compared with homogenized samples (Ju et al., 2025). Let $\mathbf{e}_i \in \mathbb{R}^d$ be the embedding of response y_i. We first compute the semantic centroid $\mathbf{c}$ and the raw cosine deviation d_i as:

$$\mathbf{c} = \frac{1}{G}\sum_{j=1}^{G} \mathbf{e}_j, \tag{3}$$

$$d_i = 1 - \frac{\mathbf{e}_i \cdot \mathbf{c}}{\|\mathbf{e}_i\|\|\mathbf{c}\|}. \tag{4}$$

To align this metric with the magnitude of RSD (Eq. 1) and to amplify the signal for semantically long-tail samples, we apply Z-score standardization to d_i (Eq. 4):

$$\mathcal{N}_{sem}(y_i) = \frac{d_i - \mu_d}{\sigma_d + \epsilon}, \tag{5}$$

where μ_d and σ_d are the mean and standard deviation of the distances within the group. Notably, $\mathcal{N}_{sem}$ allows for negative scores, effectively characterizing highly homogenized reasoning patterns.

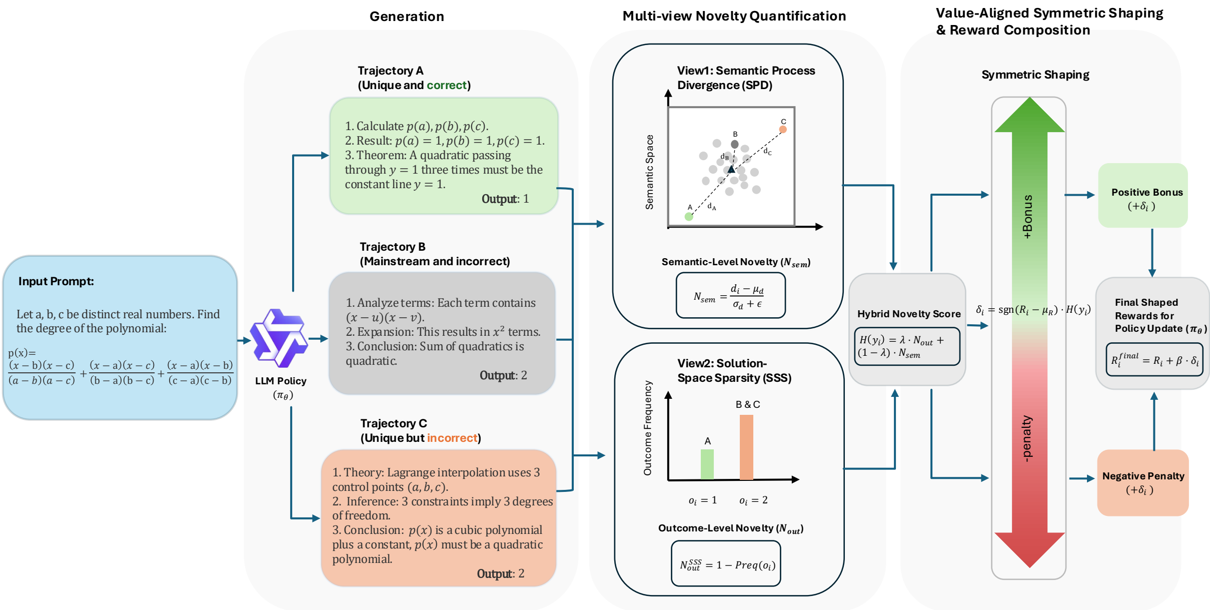


Figure 2: Overview of the proposed VANE. Taking SSS+SPD as an example, Trajectory A receives a substantial novelty bonus for its unique reasoning and sparse, correct outcome. Conversely, Trajectory C incurs a heavy penalty; despite high semantic novelty stemming from logical errors, it is penalized for incorrectness. Trajectory B yields an intermediate reward.

3.1.3 Hybrid Aggregation

We synthesize these metrics into a Hybrid Novelty Score $\mathcal{H}(y_i)$, using a weighting coefficient $\lambda \in [0, 1]$ to control the trade-off between outcome and process novelty:

$$\mathcal{H}(y_i) = \lambda \cdot \mathcal{N}_{out}(y_i) + (1 - \lambda) \cdot \mathcal{N}_{sem}(y_i). \quad (6)$$

3.2 Value-Aligned Symmetric Shaping

Indiscriminate diversity incentives often lead to reward hacking, where the model optimizes for variance at the expense of correctness. Furthermore, distinct failure modes require differentiated handling. Therefore, we introduce a Value-Alignment Function $\psi(y_i)$ to couple exploration with validity, defined based on the sign of the relative advantage:

$$\psi(y_i) = \mathrm{sgn}(R_i - \mu_{\mathcal{R}}). \quad (7)$$

In the case of outcome-based RLVR, $\psi(y_i)$ yields discrete values: We set the value to 0 for groups exhibiting zero variance (all correct/incorrect) so as to exclude special rollout groups (Yu et al., 2025), whereas for groups with mixed results, it distinguishes valid trajectories with +1 and invalid ones with -1. The Directional Novelty Bias δ_i is then computed as:

$$\delta_i = \psi(y_i) \cdot \mathcal{H}(y_i). \quad (8)$$

This implements symmetric shaping, where high-performing outliers ($R_i > \mu_{\mathcal{R}}$) receive a positive bias to amplify rare, superior solutions, whereas low-performing outliers ($R_i < \mu_{\mathcal{R}}$) incur a negative penalty to prune diverse yet erroneous hallucinations.

3.3 Final Optimization

The final shaped reward R_i^{final} incorporates the bias (Eq. 8) scaled by an intensity factor $\beta \geq 0$. This coefficient controls the contribution of the novelty reward to the overall objective, with a higher β encouraging the model to prioritize optimization towards directions with greater novelty without sacrificing correctness:

$$R_i^{final} = R_i + \beta \cdot \delta_i. \quad (9)$$

This shaped reward replaces the raw reward in the GRPO advantage calculation. The algorithm is detailed in Appendix A.

4 Experiment Setup

Detailed setup is provided in Appendix B.

4.1 RLVR Configuration

Models. We adopt Qwen2.5-Math-7B (Yang et al., 2024) as the primary backbone for our experiments. To verify scalability and cross-family generalization, we extend our evaluation to Qwen2.5-

1.5B-Math, Qwen2.5-7B-Instruct, Qwen2.5-14B, and Llama3.1-8B-Instruct (Grattafiori et al., 2024). For the SPD method, we adopt Qwen3-Embedding-0.6B (Zhang et al., 2025) to convert responses into embeddings.

Training Details. All reinforcement learning experiments are conducted using the VeRL framework (Sheng et al., 2025). We utilize the deduplicated version of the DAPO-17k dataset (Yu et al., 2025) for training. The rollout and update batch sizes are both set to 256. During the rollout phase, we sample 8 responses for each prompt. For VANE, we set the shaping intensity $\beta = 2$. The balancing coefficient λ is set to 0.3 specifically for the RSD+SPD variant on Qwen2.5-Math-7B, while $\lambda = 0.5$ is maintained for all other configurations.

4.2 Evaluation

We assess performance on nine challenging benchmarks, comprising six In-Distribution (ID) mathematical tasks—AIME 2024/2025, AMC (Li et al., 2024), MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024)—and two Out-Of-Distribution (OOD) datasets, specifically ARC-c (Clark et al., 2018) and MMLU-Pro (Wang et al., 2024), to evaluate broader reasoning capabilities. We follow the evaluation settings in (Zhan et al., 2025), report $Avg@32$ for datasets with limited samples (AIME and AMC) to mitigate variance, while the standard $Pass@1$ accuracy is used for all other benchmarks. All evaluations are conducted with a sampling temperature of 0.6 and a top-p value of 0.95, utilizing the `Oat-evaluator` library (Liu et al., 2025b) for answer extraction and verification.

4.3 Baselines

We benchmark VANE against GRPO (the backbone algorithm) and DAPO (Yu et al., 2025), which optimizes GRPO via decoupled clip hyperparameters and token-level loss, etc. Additionally, we compare against state-of-the-art exploration-enhanced methods, including Ent-Adv (Cheng et al., 2025), which promotes exploration by encouraging longer chains of thought, as well as Clip-Cov and KL-Cov (Cui et al., 2025b), which control policy updates based on token entropy covariance. For hyperparameters, we follow the recommendations in their respective original papers, while keeping all common hyperparameters consistent with the VANE.

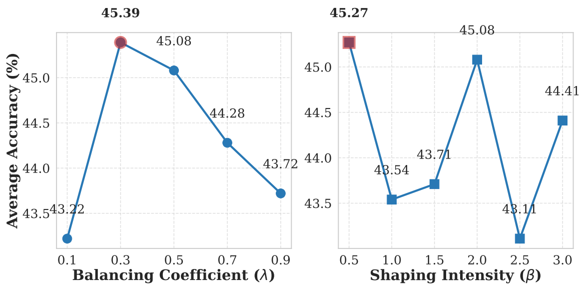


Figure 3: The plots illustrate the impact of the balancing coefficient λ (top) and shaping intensity β (bottom) on the average accuracy across six math benchmarks.

5 Experiments

To evaluate VANE, we address four core research questions: 1) How do Shaping Intensity (β) and Novelty Balancing (λ) affect performance? 2) How does VANE influence the learning process compared to baselines? 3) How does VANE perform against baselines on ID/OOD benchmarks across varying models? 4) Is the symmetric shaping mechanism valid—specifically, should high-novelty errors be penalized?

5.1 Sensitivity Analysis

Shaping Intensity (β). We investigate the impact of the shaping intensity factor β, which controls the magnitude of the novelty bias. Fixing $\lambda = 0.5$ for the RSD+SPD setting, we vary $\beta \in \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$. As shown in Figure 3, the relationship between average accuracy and shaping intensity is not strictly linear. We observe that performance remains at a high level when the value is around 0.5, 2.0, and 3.0. This is attributed to the fact that our reward shaping direction is aligned with the correctness of the original results. Therefore, the pursuit of diversity does not significantly compromise performance, demonstrating the high robustness of the method.

Novelty Balancing (λ). We analyze the balancing coefficient λ, which governs the trade-off between outcome (RSD) and process (SPD) novelty. We fix $\beta = 2.0$ and vary λ. Results in Figure 3 indicate that performance improves as λ increases up to 0.3, where it peaks, before gradually declining. This suggests that a higher weight on process novelty ($\lambda < 0.5$) enhances model capabilities, underscoring the importance of diverse reasoning paths.

Based on these analyses, we set $\beta = 2.0$ and $\lambda = 0.5$ as the default configuration for the major-

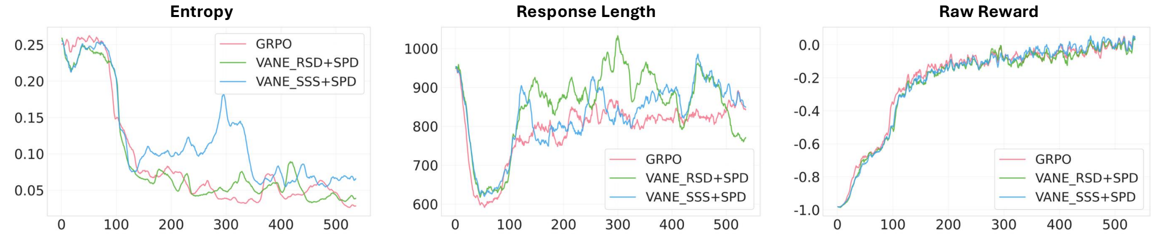


Figure 4: Training dynamics (536 steps) of GRPO, VANE(RSD+SPD), and VANE(SSS+SPD). while achieving similar raw rewards to GRPO, VANE shows stronger exploration by sustaining higher entropy and response length.

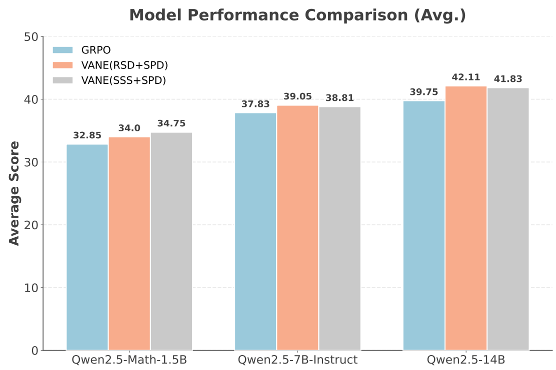


Figure 5: The average performance of the Qwen model family across six math benchmarks shows that VANE-trained models consistently outperform the GRPO.

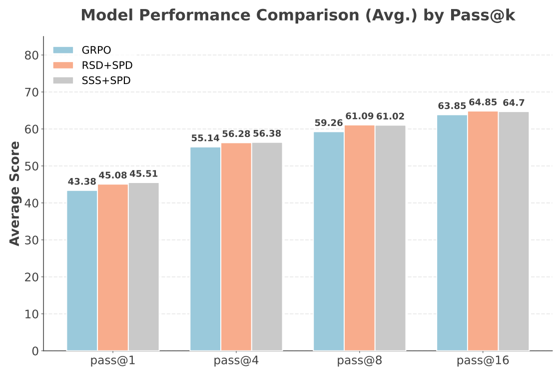


Figure 6: Aggregated Pass@k performance across six math benchmarks with scaling sample counts.

ity of our experiments. The sole exception is the Qwen-Math-7B model under the RSD+SPD setting, where we adopt $\lambda = 0.3$ due to its observed superior performance.

5.2 Training Dynamics

We further analyze the training process before detailing the main results. As shown in Figure 4, although the average reward curves converge similarly, VANE exhibits distinct behavior in average response length and entropy. Notably, SSS+SPD maintains higher response length and entropy during the mid-training phase before convergence, indicating a sustained exploratory mode.

5.3 Main Results

We present the main experimental results in Table 1 and 5. Our analysis focuses on performance across varying models and computational scaling.

Performance Analysis. Overall, VANE consistently outperforms the GRPO and advanced methods across diverse model architectures and scales on both ID and OOD benchmarks.

VANE exhibits good performance and scalability on mathematical tasks. On Qwen-2.5-Math-7B, VANE (SSS+SPD) achieves an average score of 45.51%, surpassing GRPO by 2.13% and excelling in challenging benchmarks like AIME24. This robustness is consistent across architectures and scales: it is mirrored in Llama-3.1-8B-Instruct (23.12% vs. 22.09%) and extends to larger models, boosting the Qwen-2.5-14B in-distribution average to 42.11%.

VANE demonstrates good OOD generalization capabilities across most scenarios. The RSD+SPD dominates general reasoning on Qwen-2.5-Math-7B with a score of 59.34%, exceeding GRPO by 12.73%. This trend persists with Llama-3.1-8B-Instruct (67.51%) and systematic gains are observed across the Qwen-2.5 family (1.5B–14B). Notably, even on the resource-constrained 1.5B model, OOD performance improves from 24.62% to 31.28%, highlighting the method's efficiency in preventing overfitting.

Distinct novelty strategies favor different domains. The SSS+SPD variant excels in mathematical tasks. Unlike RSD, which is constrained by reward correctness and thus acts conservatively, SSS directly incentivizes output diversity, meeting the high innovation demands of complex reasoning. Conversely, RSD+SPD proves superior for OOD generalization. Since OOD tasks often involve restricted candidate sets (e.g., multiple-choice op-

Model	In-Distribution							Out-of-Distribution		
	AIME24	AIME25	AMC	MATH500	Minerva	Olympiad	**Avg.**	ARC-c	MMLU-Pro	**Avg.**
Llama-3.1-8B-Instruct	6.77	**1.15**	20.97	45.80	24.63	16.74	19.34	30.97	42.77	36.87
GRPO	8.44	0	25.72	50.00	**31.62**	16.74	22.09	83.79	49.76	66.78
+ **VANE (RSD+SPD)**	7.50	0.31	**31.63**	49.80	29.78	18.52	22.92	84.90	**50.12**	**67.51**
+ **VANE (SSS+SPD)**	**8.75**	0	27.67	**53.20**	28.68	**20.44**	**23.12**	**84.98**	48.94	66.96
Qwen-2.5-Math-7B	10.94	4.38	33.77	48.20	10.29	14.52	20.35	16.13	17.47	16.80
GRPO	25.42	10.83	**63.06**	81.80	36.40	42.81	43.38	52.47	40.74	46.61
+ DAPO	29.27	11.46	61.90	82.00	37.50	42.67	44.13	55.12	43.99	49.56
+ Ent-Adv	26.67	12.08	59.00	77.60	35.66	41.33	42.06	56.66	45.32	50.99
+ KL-Cov	30.21	10.83	62.46	81.80	36.40	41.78	43.91	54.95	44.32	49.64
+ Clip-Cov	29.90	**13.44**	61.90	79.20	39.70	44.00	44.69	66.72	47.48	57.10
+ **VANE (RSD+SPD)**	28.23	12.93	60.39	**85.60**	38.60	44.74	45.08	**69.20**	**49.48**	**59.34**
+ **VANE (SSS+SPD)**	**32.29**	12.08	60.81	81.40	**40.44**	**46.07**	**45.51**	47.70	46.55	47.13

Table 1: Performance comparison of Llama-3.1-8B-Instruct and Qwen-2.5-Math-7B. GRPO is listed at the same level as a primary baseline, while subsequent methods (denoted with "+") indicate improvements built upon GRPO.

tions), the applicability of solution-based sparsity is limited, whereas RSD remains robust.

5.4 Computational Scale-up Analysis.

We investigate performance scaling with increased inference budgets ($k = 1$ to 16). As shown in Figure 6 and Appendix Table 8, VANE consistently outperforms GRPO. While the margin narrows slightly compared to the single-sample setting, VANE (RSD+SPD) retains its advantage, achieving 64.85% Pass@16 versus GRPO's 63.85%, confirming robust competitiveness across sampling scales.

5.5 Validation of Symmetric Shaping

A key distinction of VANE is its symmetric shaping. To validate this design, we compare three settings on Qwen2.5-Math-7B and Llama3.1-8B-Instruct:

- **Penalize (VANE SSS+SPD):** Our standard setting where $\delta_i = -\beta \cdot \mathcal{H}(y_i)$ for errors. This applies a heavier penalty to "unique failures" than common ones.
- **Ignore:** Errors receive a fixed lower-bound reward, regardless of novelty.
- **Reward:** We remove the sign term $\psi(y_i)$ and set $\delta_i = +\beta \cdot \mathcal{H}(y_i)$ for all samples. This incentivizes being "different" regardless of correctness.

We analyze the training trajectories over 536 steps, as illustrated in Figure 7 (refer to Appendix C, Figure 9 for Llama-3.1-8B-Instruct curves). The Reward setting suffers from immediate reward hacking, where the model optimizes for diversity at the expense of correctness, causing the raw reward to collapse. A closer comparison between Penalize and Ignore reveals that while the Ignore group initially learns, it eventually underperforms. Notably, the Penalize group exhibits distinct dynamic characteristics, maintaining lower entropy and longer response lengths throughout the training process.

To quantify the resulting performance gap, we compare these settings in Appendix Table 9 (visualized in Figure 8). The results show a consistent degradation when penalties are removed. For instance, under the SSS+SPD configuration, the average math accuracy for Qwen2.5-Math-7B drops from 45.51% to 40.54%, and for Llama-3.1-8B-Instruct from 23.12% to 20.47%. Similar regressions are observed in OOD tasks (e.g., Llama RSD+SPD declines from 67.51% to 62.40%).

Synthesizing the observed training dynamics with these performance outcomes, we hypothesize that penalizing novel failures acts as a regularizer. Inspired by (Cheng et al., 2025), we posit that this mechanism curbs the random, high-entropy guessing prevalent in the Ignore setting and compels the model to generate longer, deliberative reasoning chains to ensure correctness, a hypothesis further validated through case studies in Appendix D.

5.6 Ablation Study

To understand the impact of each component in VANE, we conducted comprehensive ablation studies on the Qwen2.5-Math-7B model. The summary results are presented in Table 2, with full experimental details provided in Appendix Table 9.

Single components fail to achieve the optimal trade-off between specialization and generalization. RSD-only provides a balanced uplift, improving both math (44.20%) and general reasoning

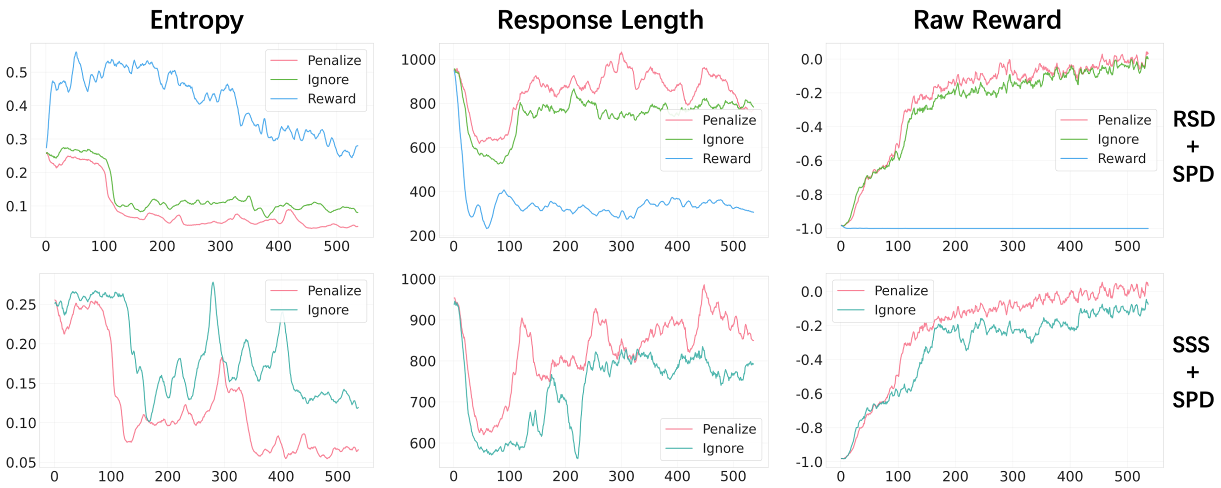


Figure 7: Training dynamics of VANE on Qwen2.5-Math-7B across Penalize, Ignore, and Reward settings. Indiscriminate diversity rewards in the Reward setting led to training collapse. Conversely, the Penalize setting outperformed the Ignore setting, sustaining higher raw rewards, increased response lengths, and lower entropy.

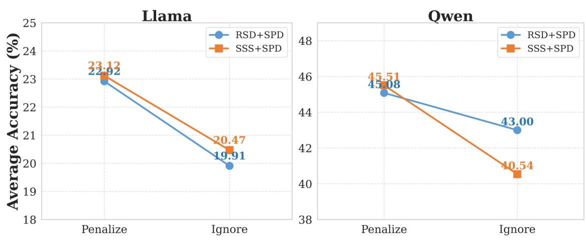


Figure 8: Average results on math benchmarks for Qwen2.5-Math-7B and Llama3.1-8B-Instruct trained with Penalize versus Ignore settings.

Model	Avg.(ID)	Avg.(OOD)
Qwen2.5-Math-7B-GRPO	43.38	46.61
Single Component		
+ RSD-only	44.20	57.25
+ SSS-only	44.01	40.83
+ SPD-only	42.82	58.04
Combined Components		
+ RSD + SPD	45.08	**59.34**
+ SSS + SPD (VANE)	**45.51**	47.13
Ablation (Ignore)		
+ RSD + SPD (Ignore)	43.00	54.58
+ SSS + SPD (Ignore)	40.54	55.24

Table 2: Ablation study on In-Distribution and Out-of-Distribution performance.

(57.25%). In contrast, SSS-only sharpens mathematical accuracy to 44.01% but suffers a regression in OOD tasks (40.83%) compared to the baseline (46.61%). SPD-only, while maintaining robust OOD performance (58.04%), yields slightly lower math performance (42.82%) than the baseline.

The combined models leverage the strengths of these components. RSD+SPD achieves the highest OOD generalization (59.34%) while securing strong math scores (45.08%). The SSS+SPD configuration yields the peak In-Distribution accuracy (45.51%), effectively using SPD to recover a portion of the OOD capability lost by SSS-only, improving it from 40.83% to 47.13%.

The "Ignore" ablation (detailed in Section 5.5) leads to performance degradation. For the RSD+SPD setup, removing negative penalties causes the Math Avg to drop from 45.08% to 43.00%. Similarly, the VANE (SSS+SPD) model sees its math accuracy plummet from 45.51% to 40.54% when errors are ignored. This confirms that symmetric shaping is vital for grounding exploration in correctness and preventing the accumulation of invalid patterns.

6 Conclusion

To alleviate mode collapse in RLVR and address the limitations of existing diversity incentives ,which are typically constrained to a single perspective and fail to differentiate among erroneous samples, we propose VANE. Our method quantifies novelty at both outcome and semantic levels and employs a symmetric shaping mechanism to explicitly couple novelty with validity within the reward signal. Extensive experiments demonstrate that VANE achieves excellent performance on mathematical task benchmarks and exhibits robust out-of-distribution (OOD) generalization. Moreover, our empirical results verify the necessity of penalizing diverse errors, confirming that this synergy is critical for sustaining performance and preventing training degeneration.

Limitations

Despite its effectiveness, VANE presents certain limitations. First, the introduction of additional hyperparameters, such as the shaping intensity and balancing coefficient, increases the complexity of tuning compared to standard RLVR. Second, the framework exhibits strategy dependency, where distinct novelty metrics favor different domains (e.g., SSS for math versus RSD for OOD), currently lacking an adaptive selection mechanism. Finally, the calculation of semantic divergence requires an external embedding model, imposing additional computational overhead during the training rollout phase.

References

DAVID H ACKLEY, GEOFFREY E HINTON, and TERRENCE J SEJNOWSKI. 1985. A learning algorithm for boltzmann machines. *COGNITIVE SCIENCE*, 9:147–169.

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. 2025. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models.

Oron Anschel, Alon Shoshan, Adam Botach, Shunit Haviv Hakimi, Asaf Gendler, Emanuel Ben Baruch, Nadav Bhonker, Igor Kviatkovsky, Manoj Aggarwal, and Gérard G. Medioni. 2025. Group-aware reinforcement learning for output diversity in large language models. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pages 10–21.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Hao-Si Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. 2025a. The entropy mechanism of reinforcement learning for reasoning language models. *ArXiv*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025b. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Prasoon Goyal, Scott Niekum, and Raymond J Mooney. 2019. Using natural language for reward shaping in reinforcement learning. *arXiv preprint arXiv:1903.02020*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2025b. Reward reasoning model. *arXiv preprint arXiv:2505.14674*.

Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. 2022. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. *Advances in Neural Information Processing Systems*, 35:15281–15295.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

Jujie He, Jiacai Liu, Chris Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. 2025. Skywork open reasoner 1 technical report. *ArXiv*.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. 2025. Advancing language model reasoning through reinforcement learning and inference scaling. *arXiv preprint arXiv:2501.11651*.

Feng Ju, Zeyu Qin, Rui Min, Zhitao He, Lingpeng Kong, and Yi R Fung. 2025. Reasoning path divergence: A new metric and curation strategy to unlock llm diverse thinking. *arXiv preprint arXiv:2510.26122*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *ArXiv*.

Nathan Lambert, Jacob Daniel Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *ArXiv*.

Jack Lanchantin, Angelica Chen, Janice Lan, Xian Li, Swarnadeep Saha, Tianlu Wang, Jing Xu, Ping Yu, Weizhe Yuan, Jason E Weston, and 1 others. 2025. Bridging offline and online reinforcement learning for llms. *arXiv preprint arXiv:2506.21495*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Ang Li, Zhihang Yuan, Yang Zhang, Shouda Liu, and Yisen Wang. 2025a. Know when to explore: Difficulty-aware certainty as a guide for llm reinforcement learning. *arXiv preprint arXiv:2509.00125*.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 110–119.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason E. Weston, Jack Lanchantin, and Tianlu Wang. 2025b. Jointly reinforcing diversity and quality in language model generations. *ArXiv*.

Jianxing Liao, Tian Zhang, Xiao Feng, Yusong Zhang, Rui Yang, Haorui Wang, Bosi Wen, Ziying Wang, and Runzhi Shi. 2025. Rlmr: Reinforcement learning with mixed rewards for creative writing. *arXiv preprint arXiv:2508.18642*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *ArXiv*.

Jiawei Liu and Lingming Zhang. 2025. Code-r1: Reproducing r1 for code with reliable rewards. *arXiv preprint arXiv:2503.18470*, 3.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025a. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025c. Inference-time scaling for generalist reward modeling. *arXiv preprint arXiv:2504.02495*.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, and 1 others. 2025. Deepcoder: A fully open-source 14b coder at o3-mini level. *Notion Blog*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. 2024. Is temperature the creativity parameter of large language models? *arXiv preprint arXiv:2405.00492*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Stewart Slocum, Asher Parker-Sartori, and Dylan Hadfield-Menell. 2025. Diverse preference learning for capabilities and alignment. In *The Thirteenth International Conference on Learning Representations*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Daniil Tiapkin, Nikita Morozov, Alexey Naumov, and Dmitry P Vetrov. 2024. Generative flow networks as entropy-regularized rl. In *International Conference on Artificial Intelligence and Statistics*, pages 4213–4221. PMLR.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. 2023. Text2reward: Reward shaping with language models for reinforcement learning. *arXiv preprint arXiv:2309.11489*.

Rihui Xin, Han Liu, Zecheng Wang, Yupeng Zhang, Dianbo Sui, Xiaolin Hu, and Bingning Wang. 2025. Surrogate signals from format and length: Reinforcement learning for solving mathematical problems without ground truth answers. *arXiv preprint arXiv:2505.19439*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F Wong, and Yu Cheng. 2025. Exgrpo: Learning to reason from experience. *arXiv preprint arXiv:2510.02245*.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A VANE Algorithm

Algorithm 1 VANE: Value-Aligned Novelty Exploration

Input: Prompt batch $\mathcal{B}$, Policy π_θ, Reference π_{ref}, Group size G, Hyperparams λ, β.
for each training step **do**
  Sample prompts $x \sim \mathcal{B}$.
  for each prompt x **do**
    Sample group $\{y_1, \dots, y_G\} \sim \pi_\theta(\cdot|x)$.
    Compute rewards $\mathcal{R} = \{r(x, y_1), \dots, r(x, y_G)\}$.
    Compute Embeddings $\{\mathbf{e}_1, \dots, \mathbf{e}_G\}$ and Centroid $\mathbf{c}$.
    for $i = 1$ to G **do**
      Calculate $\mathcal{N}_{out}(y_i)$ (RSD or SSS).
      Calculate $\mathcal{N}_{sem}(y_i)$ via Z-scored semantic distance.
      $\mathcal{H}(y_i) \leftarrow \lambda \mathcal{N}_{out} + (1-\lambda)\mathcal{N}_{sem}$.
      $\psi(y_i) \leftarrow \text{sgn}(R_i - \mu_{\mathcal{R}})$.
      $R_i^{final} \leftarrow R_i + \beta \cdot \psi(y_i) \cdot \mathcal{H}(y_i)$.
    end for
    Compute Advantages $\hat{A}$ using R^{final}.
  end for
  Update π_θ via GRPO objective using $\hat{A}$.
end for

B detailed Experimental Setup

Dataset Details. The detailed sizes of the training set and each benchmark used in our experiments are presented in 3.

Dataset	Size
DAPO-Math	17398
In-Distribution	
AMC	2,656
OlympiadBench	675
MATH-500	500
Minerva	272
AIME 2024	30
AIME 2025	30
Out-of-Distribution	
MMLU-Pro	12,032
ARC-Challenge	1,172

Table 3: Detailed statistics of training dataset and benchmarks.

VANE Implementation Details. All models are trained using the VERL framework with the GRPO algorithm on 16×NVIDIA A100 80GB GPUs. We set both the global batch size and mini-batch size to 256, initializing the learning rate at 1×10^{-6}. During training, we perform 8 rollouts with a sampling temperature of 1.0, training for 8 epochs with a clipping ratio of 0.28, while disabling both KL divergence and entropy regularization losses. Considering context window constraints and the presence of lengthy prompts, we set both the maximum prompt and response lengths to 2048 tokens for Qwen2.5-Math-7B. For evaluation, we employ a temperature of 0.6, top-p sampling with $p = 0.95$, and a maximum response length of 8192 tokens.

Baseline Implementation Details. For DAPO, we set the lower and upper clipping ratios to 0.2 and 0.28, respectively, and utilize the token-mean loss. For Clip-Cov and KL-Cov, the selected token ratio is set to 2×10^{-3}, with coverage lower and upper bounds set to 1 and 5, respectively, and a penalty coefficient of 1. For Ent-Adv, we configure $\alpha = 0.4$ and $\kappa = 2.0$. All other hyperparameters remain consistent with the VANE configuration.

C Additional Results & Figures

This section presents the comprehensive experimental results. 5 and 6 analyze the sensitivity of VANE (configured with RSD+SPD) to hyperparameter variations: 5 fixes $\lambda = 0.5$ while varying β, whereas 6 fixes $\beta = 2$ while varying λ. 7 details the performance of Qwen series models across various scales on eight benchmarks, trained via GRPO and VANE. 8 reports the Pass@K results for Qwen2.5-Math-7B using both methods. 9 presents the complete ablation study results for Qwen2.5-Math-7B and Llama-3.1-8B-Instruct. Finally, Figure 9 training dynamics (536 steps) of VANE on Llama3.1-8B-Instruct comparing Penalize and Ignore settings.

D Case Study

To verify our hypothesis, we conducted a case study on the Qwen2.5-Math-7B model trained with three distinct paradigms (GRPO, Penalize setting, and Ignore setting), with a primary focus on tasks such as number theory and probability theory that demand rigorous constraint-handling capabilities. A distinct behavioral dichotomy emerges among the groups. The **GRPO Baseline** exhibits a simplification bias, tending to rely on common heuristics seen in training data. For instance, in a number theory problem requiring the factorization of $(a - b)(a + b - 10) = 0$, GRPO prematurely divided by $(a - b)$, implicitly assuming $a \neq b$ and thus failing to recover the valid edge solution "11". Similarly, in a probability task involving a modified die, it defaulted to a standard uniform prior ($P = 0.5$) rather than tracking the specific face counts. Conversely, the **Ignore** setting suffers from ungrounded exploration. Lacking negative feedback for errors, it generates diverse but logically fragile chains. It notably exhibited severe instability, such as correctly identifying a candidate solution but rejecting it due to conflated logical conditions, and ending responses with textual hallucinations (e.g., appending unrelated proper nouns). In contrast, **Penalize** demonstrates robust structured exploration. By penalizing novel failures, the model is constrained to verify its hypotheses rigorously. It successfully recovered algebraic edge cases by analyzing multiple branches and correctly computed non-standard probability distributions, confirming that symmetric shaping effectively channels exploration towards mathematically valid complexity rather than mere variance.

E Training Efficiency Analysis

We evaluate the computational cost of VANE compared to the GRPO on Qwen2.5-Math-7B. As shown in Table 4, introducing the novelty exploration mechanism results in a moderate increase in training time, ranging from approximately 15.2% to 22.2%.

Method	Training Time (h)
GRPO	22.25
VANE (RSD+SPD)	27.19
VANE (SSS+SPD)	25.63

Table 4: We report the total training time (in hours) required for convergence on the same hardware setup.

The primary overhead stems from embedding extraction for the SPD metric. Specifically, RSD+SPD incurs a slightly higher cost (27.19h) than SSS+SPD (25.63h), as RSD encourages longer reasoning chains (Figure 4) that increase the token processing load. However, this marginal increase is justified by the substantial performance improvements.

F The Use of Large Language Models

In preparing this manuscript, we used a large language model (LLM) solely for polishing the writing style and improving the clarity of the manuscript. The LLM was not used for generating research ideas, designing experiments, conducting analyses, or deriving results. All scientific contributions, including the conceptualization, methodology, experiments, and conclusions, were developed entirely by the authors.

Intensity (β)	AIME24	AIME25	AMC	MATH-500	Minerva	Olympiad	Avg.
0.5	32.60	11.98	61.94	80.60	41.54	42.96	**45.27**
1.0	31.25	12.92	62.01	78.60	34.56	41.93	43.54
1.5	31.77	10.42	57.57	81.80	37.87	42.81	43.71
2.0	28.23	12.93	60.39	85.60	38.60	44.74	45.08
2.5	27.92	12.60	60.77	76.60	37.50	43.26	43.11
3.0	33.44	11.56	60.28	79.40	39.71	42.07	44.41

Table 5: Sensitivity analysis of Shaping Intensity (β) on six math benchmarks. We observe that $\beta = 0.5$ yields the best aggregated performance.

Coeff. (λ)	AIME24	AIME25	AMC	MATH-500	Minerva	Olympiad	Avg.
0.1	19.27	12.19	58.74	84.40	38.97	45.78	43.22
0.3	31.04	13.23	61.26	81.40	42.28	43.11	**45.39**
0.5	28.23	12.93	60.39	85.60	38.60	44.74	45.08
0.7	32.50	10.94	59.45	80.60	40.44	41.78	44.28
0.9	25.94	13.75	62.58	80.40	35.66	44.00	43.72

Table 6: Sensitivity analysis of Balancing Coefficient (λ). The model achieves peak performance at $\lambda = 0.3$, balancing exploration and exploitation effectively.

Model	In-Distribution							Out-of-Distribution		
	AIME24	AIME25	AMC	MATH500	Minerva	Olympiad	**Avg.**	ARC-c	MMLU-Pro	**Avg.**
Qwen-2.5-Math-1.5B	7.71	3.54	26.43	32.40	8.82	22.52	16.90	4.10	2.35	3.23
GRPO	11.56	8.33	42.43	75.20	23.90	35.70	32.85	24.49	24.74	24.62
+ VANE (RSD+SPD)	14.06	6.15	46.08	74.20	28.68	34.81	34.00	37.03	25.52	31.28
+ VANE (SSS+SPD)	15.31	7.29	46.08	71.60	30.88	37.33	34.75	27.65	25.76	26.71
Qwen-2.5-7B-Instruct	11.98	6.46	43.56	71.80	32.72	38.22	34.12	86.95	55.92	71.44
GRPO	13.44	12.60	52.03	74.60	32.35	41.93	37.83	86.95	58.60	72.78
+ VANE (RSD+SPD)	15.83	8.85	52.56	77.20	36.76	43.11	39.05	86.69	59.22	72.97
+ VANE (SSS+SPD)	14.69	9.06	50.41	78.40	36.76	43.56	38.81	86.86	59.10	72.98
Qwen-2.5-14B	5.42	5.63	26.36	34.00	12.50	20.15	17.34	58.19	32.69	45.44
GRPO	10.42	7.19	54.67	78.40	42.68	45.19	39.75	91.04	63.96	77.50
+ VANE (RSD+SPD)	12.50	15.63	54.82	81.80	43.01	44.89	42.11	91.21	64.84	78.03
+ VANE (SSS+SPD)	15.52	16.15	53.88	76.80	42.28	46.37	41.83	92.41	64.93	78.67

Table 7: Performance comparison of Qwen 2.5 variants (1.5B Math, 7B Instruct, and 14B) across varying scales. The first row of each block shows the base model performance. Subsequent rows show the GRPO baseline and VANE improvements (denoted with "+").

Method & Metric	AIME24	AIME25	AMC	MATH-500	Minerva	Olympiad	Avg.
GRPO (Baseline)							
Pass@1	25.42	10.83	63.06	81.80	36.40	42.81	43.38
Pass@4	44.79	18.43	76.20	89.40	46.32	55.70	55.14
Pass@8	50.00	24.06	81.73	90.40	48.90	60.74	59.26
Pass@16	55.72	29.69	86.48	93.20	53.30	64.74	63.85
VANE (RSD+SPD)							
Pass@1	28.23	12.93	60.39	85.60	38.60	44.74	45.08
Pass@4	47.50	21.88	73.64	89.60	49.63	55.41	56.28
Pass@8	53.02	26.77	79.41	90.40	55.15	61.78	61.09
Pass@16	59.38	31.35	82.72	92.40	57.35	65.93	64.85
VANE (SSS+SPD)							
Pass@1	32.29	12.08	60.81	81.40	40.44	46.07	45.51
Pass@4	48.02	21.35	74.06	88.40	48.53	57.93	56.38
Pass@8	52.29	25.94	79.25	91.40	53.68	63.56	61.02
Pass@16	55.83	31.25	85.39	93.20	55.88	66.67	64.70

Table 8: Pass@k performance scaling of GRPO and VANE variants (RSD+SPD, SSS+SPD) across math benchmarks. We evaluate sampling consistency at $k = \{1, 4, 8, 16\}$.

Model	In-Distribution (Math)							Out-of-Distribution (General)		
	AIME24	AIME25	AMC	MATH-500	Minerva	Olympiad	**Avg.**	ARC-c	MMLU-Pro	**Avg.**
Qwen2.5-Math-7B-GRPO	25.42	10.83	63.06	81.80	36.40	42.81	43.38	62.97	41.82	52.40
Single Component										
+ RSD-only	28.65	11.67	60.96	80.60	39.34	44.00	44.20	66.30	48.20	57.25
+ SSS-only	31.56	10.42	61.07	79.40	39.71	41.93	44.01	40.10	41.55	40.83
+ SPD-only	24.90	8.86	59.75	79.80	40.81	42.81	42.82	70.39	45.69	58.04
Combined Components										
+ RSD + SPD	28.23	12.93	60.39	85.60	38.60	44.74	45.08	69.20	49.48	59.34
+ SSS + SPD	32.29	12.08	60.81	81.40	40.44	46.07	45.51	47.70	46.55	47.13
Ablation (Ignore)										
+ RSD + SPD (Ignore)	27.81	13.44	59.22	78.40	36.03	43.11	43.00	63.22	45.94	54.58
+ SSS + SPD (Ignore)	26.15	12.08	56.89	80.60	26.47	41.04	40.54	64.93	45.54	55.24
Llama-3.1-8B-Instruct-GRPO	8.44	0	25.72	50.00	31.62	16.74	22.09	83.79	49.76	66.78
Combined Components										
+ RSD + SPD	7.50	0.31	31.63	49.80	29.78	18.52	22.92	84.90	50.12	67.51
+ SSS + SPD	8.75	0	27.67	53.20	28.68	20.44	23.12	84.98	48.94	66.96
Ablation (Ignore)										
+ RSD + SPD (Ignore)	5.94	0.20	26.05	47.80	24.63	14.81	19.91	78.24	46.56	62.40
+ SSS + SPD (Ignore)	5.42	0	28.84	46.00	26.10	16.44	20.47	83.44	45.85	64.65

Table 9: Detailed ablation results across all individual benchmarks for Qwen2.5-Math-7B and Llama3.1-8B-Instruct.

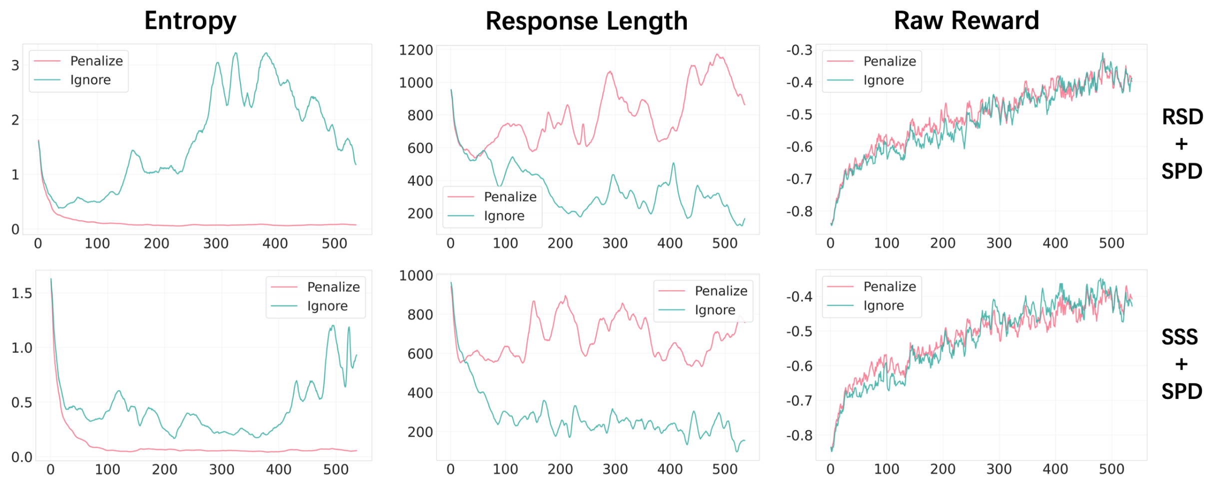


Figure 9: Training dynamics (536 steps) of VANE on Llama3.1-8B-Instruct comparing Penalize and Ignore settings. Despite achieving comparable raw rewards, the Ignore setting exhibits degeneration, where response length collapses and entropy rises to anomalous levels.

Question and Ground Truth

Question: A two-digit number has the property that the square of its tens digit plus ten times its units digit equals the square of its units digit plus ten times its tens digit. Determine all two-digit numbers which have this property, and are prime numbers.

Ground Truth: 11, 19, 37, 73

GRPO Response — *Fail: Premature Division*

The model correctly sets up the initial equation for the number $10a + b$:

$$a^2 + 10b = b^2 + 10a \implies a^2 - b^2 = 10(a - b)$$

It factors this into $(a - b)(a + b) = 10(a - b)$.
Step 1: Assumption & Division It explicitly states: "If $a \neq b$, we can divide both sides by $a-b$." This leads to the single constraint $a + b = 10$.
Step 2: Checking Candidates The model iterates through pairs (a, b) that sum to 10:

- $(1, 9) \to 19$: Prime (Keep).
- $(2, 8) \to 28$: Even, Composite.
- $(3, 7) \to 37$: Prime (Keep).
- $(4, 6) \to 46$: Even, Composite.
- $(6, 4) \ldots (9, 1)$: Only 73 is Prime.

[Error Step] Although it briefly notes that $a = b$ makes the equation trivially true, it concludes the final answer based *only* on the $a \neq b$ branch, thereby missing 11.
Answer: $\boxed{19, 37, 73}$

Analysis: GRPO performs a rigorous check on the $a \neq b$ branch but fails to integrate the edge case $a = b$ (11) into the final solution set.

VANE+Ignore Response — *Fail: Hallucination*

The model derives $(a - b)(a + b) = 10(a - b)$ and correctly identifies two cases: 1. $a + b = 10$ (yielding primes 19, 37, 73). 2. $a - b = 0$ (yielding candidate 11).
[Logic Error: Contradictory Rejection] The model explicitly checks 11 but rejects it with flawed logic: "Finally, we need to check if 11 is a prime number, which it is. **However, it does not satisfy** $a + b = 10$." The model incorrectly implies that solutions must satisfy *both* conditions simultaneously, confusing the logical OR with AND.
[Severe Hallucination] The generation degrades at the very end, appending an unrelated proper noun to the final answer box:

"… final answer is $\boxed{19, 37, 73}$ Giuliani"

Analysis: VANE+Ignore demonstrates logic collapse by rejecting a valid case it found, followed by text hallucination, indicating severe instability.

VANE Response — **Success: Robust Reasoning**

To solve this problem, we need to find all two-digit primes $10a+b$ (where $a \in \{1..9\}, b \in \{0..9\}$) that satisfy the property:

$$a^2 + 10b = b^2 + 10a$$

Step 1: Algebraic Derivation Rearranging the terms to group variables:

$$a^2 - b^2 - 10a + 10b = 0$$

Factor difference of squares and factor out -10:

$$(a - b)(a + b) - 10(a - b) = 0$$

Factor out the common term $(a - b)$:

$$(a - b)(a + b - 10) = 0$$

This equation holds true if either factor is zero. Thus, we have two distinct cases to investigate.

Step 2: Analyzing Case 1 ($a - b = 0$) If $a - b = 0$, then $a = b$. The number is of the form $10a + a = 11a$. We check primality for digits $a = 1..9$:

- $a = 1 \implies 11$: **Prime**.
- $a = 2 \implies 22$: Divisible by 2.
- $a = 3..9$: All are multiples of 11 (e.g., 33, 44...), thus composite.

Valid solution from Case 1: **11**.

Step 3: Analyzing Case 2 ($a + b - 10 = 0$) If $a + b = 10$, we check all pairs (a, b) summing to 10. Note that we must check primality for the resulting number $10a + b$:

- $(1, 9) \to 19$: **Prime**.
- $(2, 8) \to 28$: Even, composite.
- $(3, 7) \to 37$: **Prime**.
- $(4, 6) \to 46$: Even, composite.
- $(5, 5) \to 55$: Divisible by 5 (also $a = b$, covered in Case 1).
- $(6, 4) \to 64$: Even, composite.
- $(7, 3) \to 73$: **Prime**.
- $(8, 2) \to 82$: Even, composite.
- $(9, 1) \to 91$: Composite (7×13).

Valid solutions from Case 2: **19, 37, 73**.

Step 4: Final Consolidation We combine the valid primes found in both branches of the solution space:

$$S = \{11\} \cup \{19, 37, 73\}$$

Final Answer: $\boxed{11, 19, 37, 73}$

Analysis: VANE demonstrates superior reasoning diversity. Unlike the other models, it rigorously explores both branches of the algebraic factorization $(a - b)(a + b - 10) = 0$, ensuring the edge case '11' is retained in the final solution set.

Figure 10: **Case study on algebraic cases.** We compare reasoning paths for a number theory problem. The **GRPO** baseline misses the solution "11" due to premature pruning (assuming $a \neq b$), while the **Ignore** setting exhibits high-entropy instability (hallucinating "Giuliani"). In contrast, **VANE** correctly identifies the zero-product property, exploring both branches to retrieve the complete solution set.

Question and Ground Truth

Question: 9 players choose ice cream flavors (Chocolate C, Vanilla V, Strawberry S). The conditions are:

1. At least one player chooses each flavor ($C, V, S \geq 1$).
2. $C > V > S$.

Let N be the number of different assignments. Find $N \pmod{1000}$.
Ground Truth: 16 (Total $N = 2016$)

GRPO Response *Fail: Mode Collapse*

The model generates a Python script loop:

```
for c in range(1, 9):
    for v in range(1, c):
        for s in range(1, v):
            if c + v + s == 9:
                # Calculates factorials...
```

Execution Result The code finds the first valid partition $C = 6, V = 2, S = 1$ and calculates:

$$\frac{9!}{6!2!1!} = 252$$

[Error Step] It stops searching after this first match, ignoring $(5, 3, 1)$ and $(4, 3, 2)$.
Answer: $\boxed{252}$

*Analysis: GRPO suffers from **Mode Collapse**. It correctly identifies the first valid solution (6, 2, 1) but fails to explore the remaining solution space.*

VANE+Ignore Response *Fail: Calculation Error*

The model sets up the correct loops for s, v, c.
Code Execution Attempt It attempts to sum multinomial coefficients:

```
if c > v:
    sum += multinomial(9, c, v, s)
```

[Error Step: Hallucination] Despite correct logic, the model outputs an arbitrary number:

$$\text{Output} = 288$$

This number (288) is neither a valid partial sum nor the total.
Answer: $\boxed{288}$

Analysis: VANE+Ignore exhibits a severe calculation hallucination. The output 288 is factually incorrect and cannot be derived from the logic provided.

VANE Response **Success: Full Coverage**

To solve this, we must find all integer solutions (C, V, S) to the equation $C + V + S = 9$ subject to $C > V > S \geq 1$.

Step 1: Systematic Enumeration We iterate based on the smallest value S.

- **Case** $S = 1$: We need $C + V = 8$ and $C > V > 1$.
  - If $V = 2 \implies C = 6$. Valid: $(6, 2, 1)$.
  - If $V = 3 \implies C = 5$. Valid: $(5, 3, 1)$.
  - If $V = 4 \implies C = 4$. Invalid ($C \not> V$).
- **Case** $S = 2$: We need $C + V = 7$ and $C > V > 2$.
  - If $V = 3 \implies C = 4$. Valid: $(4, 3, 2)$.
  - If $V = 4 \implies C = 3$. Invalid ($C < V$).
- **Case** $S \geq 3$: Min sum $4 + 3 + 3 > 9$. Stop.

Step 2: Calculate Assignments (Detailed) We compute the multinomial coefficient $\frac{9!}{C!V!S!}$ for each valid case found above. Note that $9! = 362,880$.

1. Partition (6, 2, 1):

$$\binom{9}{6,2,1} = \frac{362,880}{(720)(2)(1)} = \frac{362,880}{1440} = 252$$

2. Partition (5, 3, 1):

$$\binom{9}{5,3,1} = \frac{362,880}{(120)(6)(1)} = \frac{362,880}{720} = 504$$

3. Partition (4, 3, 2):

$$\binom{9}{4,3,2} = \frac{362,880}{(24)(6)(2)} = \frac{362,880}{288} = 1260$$

Step 3: Aggregation and Modulo We now sum the number of ways from all three distinct partitions:

$$N = 252 + 504 + 1260$$
$$N = 756 + 1260$$
$$N = 2016$$

Finally, find the remainder modulo 1000:

$$2016 \pmod{1000} = 16$$

Answer: $\boxed{16}$

*Analysis: VANE demonstrates superior **Diversity of Thought** by systematically covering the entire solution space. Unlike GRPO which stopped at the first valid case, VANE correctly identified and summed all three distinct partitions.*

Figure 11: **Case study on combinatorial counting.** The task requires partitioning 9 items into groups (C, V, S) satisfying $C > V > S \geq 1$. **VANE** correctly enumerates the three valid integer partitions ($(6, 2, 1), (5, 3, 1), (4, 3, 2)$) and computes the multinomial sum to find $2016 \equiv 16 \pmod{1000}$. In contrast, both **GRPO** and the **Ignore** setting rely on Python scripts but fail to retrieve the correct answer due to execution hallucinations (outputting 252 and 288, respectively), highlighting the risks of relying solely on tool use without reasoning verification.

Question and Ground Truth

Question: A die with faces $\{1, 2, 3, 4, 6, 8\}$ is rolled.

- If **Odd**: Odd numbers doubled.
- If **Even**: Even numbers halved.

Find the probability that a **2** appears on the 2nd roll.

Ground Truth: $\frac{2}{9}$

GRPO Response *Fail: Invalid Assumption*

The model starts by incorrectly assuming a standard die distribution ($P(\text{odd}) = P(\text{even}) = 0.5$), ignoring the specific faces.

Step 1: Analyze Scenarios

- **Odd Case:** Assumes faces become $\{2, 2, 6, 4, 16, 8\}$. Calculates $P(2|\text{odd}) = 1/3$.
- **Even Case:** Assumes faces become $\{1, 1, 3, 2, 4, 4\}$. Calculates $P(2|\text{even}) = 1/6$.

[Error Step: Prior Probability] "The probability of rolling an odd number ... is $\frac{3}{6} = \frac{1}{2}$... even is $\frac{1}{2}$."

Step 2: Final Calculation Using this wrong prior:

$$P = \frac{1}{3} \cdot \frac{1}{2} + \frac{1}{6} \cdot \frac{1}{2} = \frac{1}{6} + \frac{1}{12} = \frac{1}{4}$$

Answer: $\boxed{\frac{1}{4}}$

Analysis: The model fails to count the actual faces $(1, 2, 3, 4, 6, 8)$. There are only 2 odd numbers $(1, 3)$, so $P(\text{odd})$ should be $1/3$, not $1/2$. This is a ***Prior Assumption Error****.*

VANE+Ignore Response *Fail: Hallucination*

Scenario 1 (Odd): Correctly calculates $P(\text{Odd}) = 1/3$. Die changes to $\{2, 4, 6, 8, 2, 8\}$. It calculates $P(2|\text{Odd}) = 2/6 = 1/3$. Subtotal: $1/3 \times 1/3 = 1/9$.

Scenario 2 (Even): Die becomes $\{1, 1, 3, 4, 6, 4\}$. **[Error Step]** "Probability of rolling a 2 ... is 0 (since 2 is not on the die)." The model performs the halving but fails to realize that $4 \div 2 = 2$.

[Hallucination] The response ends with unrelated text:

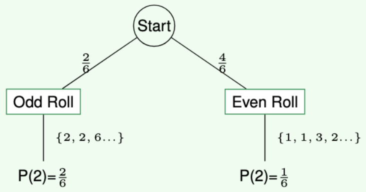
"... $\boxed{\frac{1}{9}}$ **[Unrelated Text: Cruise Ship]**"

Answer: $\boxed{\frac{1}{9}}$

Analysis: VANE+Ignore makes a calculation error (claiming no '2' exists) and ends with a textual hallucination, indicating severe generation instability.

VANE Response **Success: Correct Logic**

To solve this, we track the specific state of the die faces conditioned on the outcome of the first roll.



Step 1: Initial Probabilities

- **Odd Faces**: $\{1, 3\}$ (Count: 2).

$$P(\text{Odd}) = \frac{2}{6} = \frac{1}{3}$$

- **Even Faces**: $\{2, 4, 6, 8\}$ (Count: 4).

$$P(\text{Even}) = \frac{4}{6} = \frac{2}{3}$$

Step 2: Determine New Faces

- **If Odd**: Odd faces double ($1 \to 2, 3 \to 6$). Evens stay. New Set: $\{2, 2, 6, 4, 12, 8\}$. (Two 2s).

$$P(2|\text{Odd}) = \frac{2}{6} = \frac{1}{3}$$

- **If Even**: Even faces halve ($2 \to 1, 4 \to 2$, etc). New Set: $\{1, 1, 3, 2, 3, 4\}$. (One 2).

$$P(2|\text{Even}) = \frac{1}{6}$$

Step 3: Total Probability Calculation Using the Law of Total Probability:

$$P = P(2|\text{Odd})P(\text{Odd}) + P(2|\text{Even})P(\text{Even})$$

$$P = \left(\frac{1}{3} \times \frac{1}{3}\right) + \left(\frac{1}{6} \times \frac{2}{3}\right) = \frac{1}{9} + \frac{1}{9} = \frac{2}{9}$$

Step 4: Code Verification Next writes Python code to verify:

```
# Case 1: Odd (Prob 1/3) -> P(2) = 1/3
p1 = Fraction(1,3) * Fraction(1,3)
# Case 2: Even (Prob 2/3) -> P(2) = 1/6
p2 = Fraction(2,3) * Fraction(1,6)
print(p1 + p2)
```

Answer: $\boxed{\frac{2}{9}}$

Analysis: VANE correctly identifies that the die is non-standard, assigning the correct weights $(1/3$ vs $2/3)$ to the initial branches, unlike GRPO.

Figure 12: **Case study on probability with dynamic state tracking.** We analyze reasoning paths for a non-standard die problem. The **GRPO** baseline defaults to a standard prior ($P = 0.5$), ignoring specific face counts. The **Ignore** setting exhibits instability, resulting in calculation errors and textual hallucinations (e.g., "Cruise Ship"). In contrast, **VANE** correctly models the conditional state changes to derive the exact solution.