# Representation-Aware Advantage Estimation: Your Reward Model Provides More Than A Scalar Output

**Guozheng Li**[*]
Southeast University
gzli@seu.edu.cn

**Xiyan Fu**[*]
Nanyang Technological University
xiyan.fu@ntu.edu.sg

**Yiwen Guo**[†]
Independent Researcher
guoyiwen89@gmail.com

## Abstract

Current reinforcement learning from human feedback (RLHF) methods primarily rely on scalar rewards from a trained reward model (RM). While effective, scalar rewards are often noisy and fail to capture fine-grained preference differences, whereas RM hidden states encode richer semantic and preference information. We introduce the representation-aware advantage estimation, which leverages RM hidden states and models them as auxiliary signals for better advantage estimation. Specifically, we propose the Graph-based Advantage Estimation (GraphAE), treat each sampled group as a graph, where nodes correspond to responses and edges capture their similarity in the RM hidden space. Then advantages are computed via graph propagation, enabling each sample to incorporate contextual information from its neighbors. GraphAE is lightweight and can be seamlessly integrated into existing group-based RL algorithms. We apply GraphAE to GRPO, GSPO and RLOO, and conduct extensive experiments on different models and benchmarks. Empirical results show consistent improvements across three benchmarks, with gains of up to + 6.3 on Arena-Hard-v0.1, + 8.27 on AlpacaEval 2.0, and + 0.22 on MT-Bench. These results demonstrate that leveraging RM representations leads to more sample efficient and robust RLHF [1].

## 1 Introduction

Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017) has emerged as a cornerstone for aligning Large Language Models (LLMs) with human preferences (Bai et al., 2022; Ganguli et al., 2022). The standard RLHF pipelines rely on a scalar reward produced by a reward model (RM) to guide policy updates via policy gradient methods (Sutton et al., 1999). Many policy optimization methods have been developed to improve the stability and efficiency of RLHF, including PPO (Schulman et al., 2017), group-based variants such as GRPO (Shao et al., 2024), as well as direct preference optimization methods such as DPO (Rafailov et al., 2023). These algorithms have been widely applied to tasks such as instruction following (Ouyang et al., 2022), dialogue (Zhang et al., 2020), and reasoning (Wei et al., 2022).

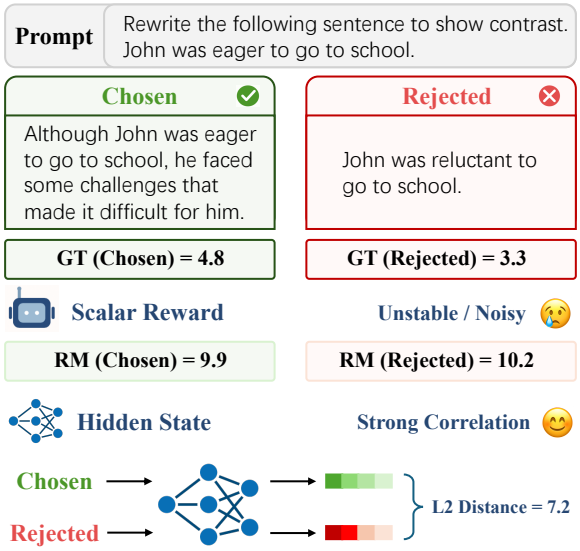


Figure 1: The same RM may assign similar or even reversed scalar scores to opposite responses, while its hidden representations clearly separate them. GT denotes ground truth and RM denotes reward model.

Although these methods have achieved strong empirical performance, they heavily rely on the scalar output of the RM. However, we find different responses may receive identical or very similar scores despite exhibiting clear semantic differences, as shown in Figure 1. That is, a scalar

[*]Equal contribution.
[†]Corresponding author.
[1]https://github.com/1iguozheng/graphae



*Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, pages 34839–34851
October 24-29, 2026 ©2026 Association for Computational Linguistics

reward is often insufficient to capture fine-grained differences between responses. As a result, the advantage estimates computed in many online RL algorithms based on such reward signals can be inaccurate or noisy (Zhang et al., 2025).

To address this limitation, we aim to identify a more expressive signal that can better capture fine-grained differences among responses. Recent studies on interpretable RMs suggest that intermediate representations encode rich, fine-grained information related to preference and reward signals that is not fully reflected in the final scalar output (Wang et al., 2024c; Zhang et al., 2026). We therefore utilize the RM's hidden state representations prior to the final scalar projection for advantage estimation. Building on these representations, we move beyond scalar advantage estimation and introduce a Representation-Aware Advantage Estimation.

In specific, we propose the Graph-based Advantage Estimation (GraphAE), a structured advantage estimator built upon RM representations. Given a set of sampled responses, we construct a similarity graph based on their RM representations, where edge weights reflect pairwise similarity. We then estimate advantages by solving a graph-regularized objective that balances fidelity to the original scalar rewards with smoothness over the graph (Smola and Kondor, 2003; Zhu et al., 2003). This yields a closed-form solution that can be efficiently solved in practice. The resulting rewards integrate both individual reward signals and local structural information, leading to a more informative and robust estimator. It is lightweight and can be seamlessly integrated into existing group-based RL algorithms by replacing the original advantage estimator, such as GRPO (Shao et al., 2024), GSPO (Zheng et al., 2025), and RLOO (Ahmadian et al., 2024).

We evaluate GraphAE on widely used benchmarks including Arena-Hard-v0.1 (Li et al., 2024), AlpacaEval 2.0 (Dubois et al., 2024), and MT-Bench (Zheng et al., 2023). GraphAE consistently improves performance over strong baselines on different RL algorithms and backbones, with gains of up to +6.3 on Arena-Hard, +8.27 on AlpacaEval, and +0.22 on MT-Bench, demonstrating improved alignment quality and robustness. Moreover, GraphAE reduces within-group reward variance and accelerates convergence. GraphAE is robust across hyperparameter choices and adds minimal computation during advantage estimation, showing that leveraging RM representations leads to more stable and efficient RLHF.

## 2 Related Work

### 2.1 Reinforcement Learning for LLMs

Reinforcement learning has emerged as a prevalent method for fine-tuning LLMs, with PPO (Schulman et al., 2017) and its variations (Ramamurthy et al., 2023; Wu et al., 2023) being the most prevalent methods. These techniques largely adhere to the actor-critic paradigm (Sutton et al., 1998), which significantly increases training complexity and computational resource consumption. Many recent studies have sought to mitigate this limitation. Methods such as DPO (Rafailov et al., 2023) and its variants (Meng et al., 2024; Ethayarajh et al., 2024; Hong et al., 2024) bypass reward modeling and the actor-critic learning framework by directly learning from preferences. However, existing research indicates that due to their offline nature, these approaches exhibit a performance gap compared to online RL (Xu et al., 2024). In contrast to actor-critic and offline methods, recent work explores *group-based policy optimization* (Shao et al., 2024; Yu et al., 2025; Liu et al., 2025b), which removes the need for a value function by leveraging multiple responses per prompt. For example, GRPO (Shao et al., 2024) computes advantages via group-relative rewards, avoiding critic training while remaining efficient. GSPO (Zheng et al., 2025) improves stability by moving from token- to sequence-level optimization, and RLOO (Ahmadian et al., 2024) uses a leave-one-out baseline for variance reduction. Despite these differences, they all rely on scalar rewards and treat samples independently, ignoring their structural relationships.

### 2.2 Reward Models

Reward models (RMs) are central to RLHF, serving as learned proxies for human preferences and providing scalar rewards for policy optimization (Christiano et al., 2017). The standard formulation follows the Bradley–Terry model (Bradley and Terry, 1952), which assigns scalar scores based on pairwise preference data (Bai et al., 2022; Ganguli et al., 2022). Recent work extends this paradigm with more fine-grained supervision, such as process reward models (PRMs) that provide

step-level signals (Lightman et al., 2023). To address the limitations of raw rewards, a related line of work explores reward shaping techniques that modify or augment reward signals to improve training stability, credit assignment, and robustness to reward hacking (Kwon et al., 2023; Fu et al., 2025).

Despite these efforts, most RLHF pipelines ultimately rely on a single scalar reward derived from the final hidden state, compressing rich internal representations into a point estimate (Wang et al., 2024a). This overlooks the information encoded in RM representations, which capture semantic and preference-aligned relationships among samples (Yang et al., 2024; Wang et al., 2024b). In this work, we instead leverage RM hidden states as reward features to refine advantage estimation.

## 3 Preliminary

Consider a large language model (LLM) $\pi_\theta$ parameterized by $\theta$, which takes an input sequence $x$ and generates an output sequence $y$ according to the conditional distribution $\pi_\theta(\cdot \mid x)$. Reinforcement learning from human feedback (RLHF) aims to optimize $\pi_\theta$ to maximize expected rewards aligned with human preferences. In practice, such rewards are provided by a reward model (RM) trained from human preference data. Specifically, given a dataset of preference comparisons $\mathcal{D}_{\text{pref}} = \{(x, y_w, y_l)\}$, where $y_w \succ y_l$ indicates that $y_w$ is preferred over $y_l$ for prompt $x$, preferences are commonly modeled using the Bradley–Terry (BT) model (Bradley and Terry, 1952) with an underlying ground-truth reward function $r^\star(\cdot, \cdot)$:

$$P(y_w \succ y_l \mid x) = \sigma\big(r^\star(x, y_w) - r^\star(x, y_l)\big), \quad (1)$$

where $\sigma(z) = 1/(1 + e^{-z})$ denotes the sigmoid function. A parametric RM $r(x, y)$ can then be learned via maximum likelihood estimation:

$$\mathcal{L}_{\text{RM}}(r) = \mathbb{E}_{\mathcal{D}_{\text{pref}}}\left[-\log \sigma\big(r(x, y_w) - r(x, y_l)\big)\right]. \quad (2)$$

In RLHF, the RM provides a scalar reward for each response: $r_i = r(x, y_i)$, where $y_i \sim \pi_\theta(\cdot \mid x)$. Based on these rewards, group-based RL algorithms such as GRPO (Shao et al., 2024) estimate advantages based on a global baseline:

$$A_i = r_i - \bar{r}, \quad \text{where} \quad \bar{r} = \frac{1}{G}\sum_{j=1}^{G} r_j, \quad (3)$$

for a group of $G$ sampled completions. Then RLHF formulates policy optimization as maximizing the expected advantages under the policy while constraining deviation from a reference policy $\pi_{\text{ref}}$:

$$\max_\theta \ \mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\big[A(x, y)\big] - \beta\, \mathbb{E}_{x\sim\mathcal{D}}\big[D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})\big]. \quad (4)$$

where $\beta > 0$ controls the strength of KL regularization. This objective is typically optimized using policy gradient methods (Sutton et al., 1999).

## 4 Representation-Aware Advantage Estimation

We introduce a representation-aware advantage estimation that leverages reward model representations to refine advantage signals via graph regularization. We formalize the estimation objective in §4.1 and derive its closed-form solution in §4.2. Finally, we show that the refined advantages can be seamlessly integrated into existing group-based RL algorithms in §4.3.

### 4.1 Graph-Regularized Advantage Formulation

Existing advantage estimation relies on independent scalar rewards. However, prior work (Zhang et al., 2025) suggests that reward signals are inherently noisy, as responses with similar quality can still receive noticeably different scores. This issue causes high-variance advantage estimates. To address this issue, we refine the reward signal by introducing two complementary objectives:

**Representation-aware smoothness.** We encourage similar responses to have consistent reward estimates by enforcing smoothness in the reward model representation space. To this end, we construct a similarity graph over sampled responses. Each node corresponds to a response $y_i$, represented by:

$$z_i = f_\phi(x, y_i),$$

where $z_i$ is the hidden representation before the reward head. Edges encode pairwise similarity between responses, with weights defined by a Gaussian kernel:

$$K_{ij} = \exp\left(-\frac{\|z_i - z_j\|^2}{2\sigma^2}\right). \quad (5)$$

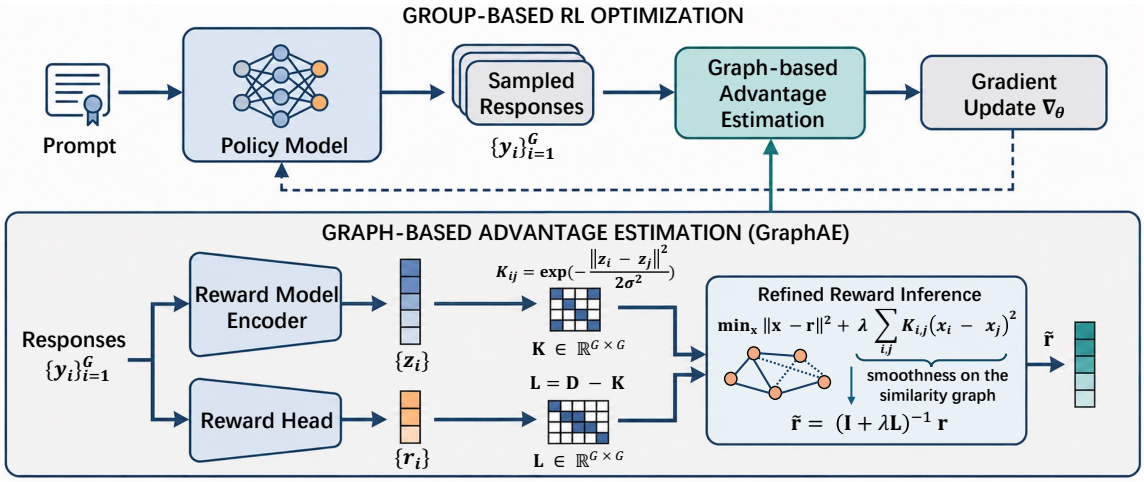


Figure 2: Overview of GraphAE. GraphAE refines advantages by leveraging RM representation structures, acting as a structured control variate that preserves the reward signal while reducing noise and improving learning efficiency.

This yields a fully-connected weighted graph that captures local relationships in the representation space, and induces a smoothness constraint $\sum_{i,j} K_{ij}(x_i - x_j)^2$ that penalizes discrepancies between similar responses.

**Fidelity to the original reward.** To prevent over-smoothing, the refined reward should remain close to the original reward signal produced by the reward model. We therefore impose a fidelity constraint of the form $\|\mathbf{x} - \mathbf{r}\|^2$, which anchors the optimized rewards to the original values.

Combining the above two objectives, we obtain the refined reward $\tilde{\mathbf{r}}$ as:

$$\tilde{\mathbf{r}} = \arg\min_{\mathbf{x}} \underbrace{\|\mathbf{x} - \mathbf{r}\|^2}_{\text{fidelity}} + \lambda \underbrace{\sum_{i,j} K_{ij}(x_i - x_j)^2}_{\text{smoothness}} \tag{6}$$

where $\lambda$ controls the trade-off between fidelity and smoothness.

### 4.2 Solution and Properties

The objective in Eq. 6 corresponds to a standard graph Laplacian regularization problem. Let $L = D - K$ denote the graph Laplacian, where $D_{ii} = \sum_j K_{ij}$. The optimization problem can be rewritten as:

$$\tilde{\mathbf{r}} = \arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{r}\|^2 + \lambda \mathbf{x}^\top L \mathbf{x}. \tag{7}$$

Taking derivatives with respect to $\mathbf{x}$ and setting to zero gives:

$$2(\mathbf{x} - \mathbf{r}) + 2\lambda L \mathbf{x} = 0, \tag{8}$$

which yields the linear system:

$$(\mathbf{I} + \lambda L)\mathbf{x} = \mathbf{r}. \tag{9}$$

Since $\mathbf{I} + \lambda L \in \mathbb{R}^{G \times G}$ is symmetric positive definite, the solution is unique and can be computed efficiently. In RLHF, the group size $G$ is typically small (e.g., $G \leq 32$), making direct solvers both efficient and numerically stable. The computational cost is $O(G^3)$ and negligible in practice. Therefore, the optimal solution admits a closed form:

$$\tilde{\mathbf{r}} = (\mathbf{I} + \lambda L)^{-1} \mathbf{r}. \tag{10}$$

This solution can be interpreted as graph-based smoothing of the reward signal, where each entry aggregates information from similar responses while remaining anchored to its original value.

### 4.3 Compatibility with Group-based RL

Our method can be seamlessly integrated into a wide range of group-based RL algorithms. Specifically, we replace the original reward $r_i$ with the refined reward $\tilde{r}_i$ in the advantage estimator. For example, in GRPO (Shao et al., 2024) :

$$A_i^{\text{GRPO}} : \quad \frac{r_i - \mu_r}{\sigma_r + \epsilon} \rightarrow \frac{\tilde{r}_i - \mu_{\tilde{r}}}{\sigma_{\tilde{r}} + \epsilon}. \tag{11}$$

This modification only affects the advantage estimation step, while leaving the policy optimization procedure unchanged.

## 5 Experimental Setup

### 5.1 Datasets and Evaluation

We evaluate the performance of GraphAE on three widely adopted RLHF benchmarks: Arena-Hard-v0.1 (Li et al., 2024), AlpacaEval 2.0 (Dubois et al., 2024), and MT-Bench (Zheng et al., 2023). These benchmarks require the model to generate answers to a wide range of open-ended questions.

We follow the standard evaluation protocol of each benchmark. For Arena-Hard-v0.1, we report the win rate (WR) against GPT-4-0314 with the 95% confidence interval (CI), using GPT-4.1 as the judge. For AlpacaEval 2.0, we report both WR and length-controlled win rate (LC WR) against GPT-4-Turbo. For MT-Bench, we report two turns and average scores using GPT-4-Turbo as the judge. More details are provided in the Appendix A.

### 5.2 Baselines

GraphAE is designed as a plug-in advantage estimator for group-based RLHF. We apply it to three representative group-based RL algorithms, including GRPO (Shao et al., 2024), GSPO (Zheng et al., 2025), and RLOO (Ahmadian et al., 2024). For each method, we replace the original scalar reward based advantage estimator with GraphAE while keeping all other components unchanged.

We compare GraphAE against RL methods that use scalar rewards for advantage estimation. To avoid the effect of reward regularization, we further include reward smoothing baselines of the form $\tilde{r}_i = (1-\alpha)r_i + \alpha\bar{r}$, where $\bar{r}$ denotes the mean reward within each rollout group and $\alpha$ controls the smoothing strength.[2] For a more comprehensive evaluation, we also include classical baselines such as PPO (Schulman et al., 2017) and the original instruction-tuned models without RL training.

### 5.3 Implementation Details

**Reward Model.** We adopt a pre-trained reward model Skywork-Reward-V2-Llama-3.1-8B (Liu et al., 2025a) given its superior performance across multiple benchmarks. The pre-trained model allows us to isolate the contribution of representation advantage estimation without confounding factors from training. In GraphAE, we extract the representation of the last token from the final layer for graph-based advantage estimation.

**Policy Training.** We conduct RL training on two widely used instruction-tuned LLMs: Qwen2.5-7B-Instruct (Yang et al., 2025) and Llama-3-8B-Instruct (Grattafiori et al., 2024). Specifically, we use a learning rate of 2e-6 and set the maximum generated sequence length to 1024. We employ a batch size of 1024 and a mini-batch size of 8, with 1 epoch and 128 gradient accumulation steps. The sampling temperature is set to 0.7 and the KL coefficient is set to 0.01. We roll out 8 responses for each prompt. We set the smoothing coefficient $\lambda$ to 1 for Qwen2.5-7B-Instruct and 0.5 for Llama-3-8B-Instruct. For the Gaussian kernel bandwidth $\sigma$, we adaptively set it to the median of all non-zero pairwise distances within each group, allowing it to adjust to different group scales.

## 6 Results and Analysis

### 6.1 Main Results

**Overall Results.** Table 1 summarizes the main results across three evaluation benchmarks. Compared with the original group-based RL algorithms, GraphAE consistently improves GRPO, GSPO, and RLOO across both model backbones and all evaluation benchmarks. This demonstrates the effectiveness and generalization ability of GraphAE, compared to original scalar based advantage estimation methods. We also observe that the uniform reward smoothing baseline (+Smooth) yields only marginal or inconsistent improvements, suggesting that the gains of GraphAE are not attributable to reward regularization, but rather to better exploitation of preference information encoded in RM representations. Moreover, all RL methods substantially outperform the w/o RL baselines, while PPO achieves only moderate gains despite requiring an additional value network.

GraphAE achieves its largest gains on Arena-Hard and AlpacaEval, improving over GRPO by up to +6.3 and +8.27 points, respectively, while the gains on MT-Bench are comparatively smaller. This pattern is consistent with the characteristics of the benchmarks. In Arena-Hard and AlpacaEval, the quality differences between responses are often subtle, making scalar rewards more susceptible to

[2] The $\alpha$ is set to 0.3 across all RL settings based on experiments in Appendix C

| Method | Qwen2.5-7B-Instruct | | | | | | | Llama-3-8B-Instruct | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Arena-Hard | | AlpacaEval | | MT-Bench | | | Arena-Hard | | AlpacaEval | | MT-Bench | | |
| | WR | CI | LC WR | WR | T1 | T2 | Avg | WR | CI | LC WR | WR | T1 | T2 | Avg |
| w/o RL | 45.7 | -2.1/+1.6 | 33.74 | 33.41 | 8.23 | 6.96 | 7.59 | 19.0 | -1.7/+1.5 | 22.09 | 22.82 | 7.19 | 6.05 | 6.62 |
| PPO | 52.9 | -1.7/+1.9 | 35.60 | 35.10 | 8.30 | 7.15 | 7.73 | 33.2 | -1.7/+1.9 | 34.60 | 34.20 | 7.35 | 6.95 | 7.15 |
| GRPO | 51.5 | -1.9/+1.7 | 36.26 | 35.94 | **8.66** | 7.01 | 7.84 | 31.1 | -1.8/+1.9 | 34.85 | 35.53 | 7.64 | 6.63 | 7.13 |
| GRPO + Smooth | 51.0 | -2.0/+1.7 | 36.40 | 36.12 | 8.61 | 7.02 | 7.82 | 30.8 | -1.9/+1.9 | 34.91 | 35.62 | 7.60 | 6.62 | 7.11 |
| GRPO + GraphAE | **57.8** | -1.9/+2.2 | **42.85** | **44.21** | 8.60 | **7.10** | **7.85** | **33.9** | -1.6/+1.8 | **37.56** | **36.03** | **7.68** | **6.74** | **7.21** |
| GSPO | 55.5 | -2.4/+2.1 | 35.76 | 36.23 | 8.50 | 7.23 | 7.86 | 31.6 | -1.5/+1.8 | 36.47 | 36.98 | 7.49 | 6.53 | 7.01 |
| GSPO + Smooth | 54.4 | -2.1/+2.2 | 35.93 | 36.40 | 8.48 | 7.21 | 7.85 | 30.4 | -1.7/+1.9 | 36.51 | 37.01 | 7.46 | 6.52 | 6.99 |
| GSPO + GraphAE | **59.5** | -1.9/+1.8 | **40.63** | **42.05** | **8.60** | **7.43** | **8.01** | **34.2** | -2.1/+1.8 | **37.85** | **37.92** | **7.64** | **6.83** | **7.23** |
| RLOO | 50.8 | -1.9/+2.2 | 35.90 | 35.20 | 8.55 | 7.05 | 7.80 | 30.5 | -1.8/+1.7 | 34.23 | 35.81 | 7.55 | 6.60 | 7.05 |
| RLOO + Smooth | 50.8 | -1.7/+2.0 | 35.81 | 35.42 | 8.50 | 7.04 | 7.77 | 30.1 | -1.6/+1.8 | 34.31 | 35.94 | 7.51 | 6.58 | 7.05 |
| RLOO + GraphAE | **55.2** | -1.8/+2.1 | **39.50** | **41.10** | **8.60** | **7.20** | **7.98** | **32.7** | -1.9/+1.7 | **36.13** | **36.81** | **7.62** | **6.75** | **7.16** |

Table 1: Main results on Qwen2.5-7B-Instruct and Llama-3-8B-Instruct across Arena-Hard-v0.1, AlpacaEval 2.0, and MT-Bench. Smooth denotes the uniform reward smoothing baseline with $\alpha = 0.3$.

noise and leading to less reliable advantage estimates. GraphAE alleviates this issue by leveraging representation aware advantage estimation, which provide richer and more stable signals. In contrast, MT-Bench exhibits a narrower score range among strong models, which limits the headroom for further improvement.

**Feature Ablation.** In this section, we explore whether the improvements of GraphAE come from the preference relevant information encoded in RM hidden states, rather than merely from introducing a graph structure. To examine this, we conduct a feature ablation by keeping the underlying Qwen + GRPO optimization pipeline unchanged and varying only the feature source used for graph construction. We compare GraphAE with three controlled variants: i) *Scalar-Graph*, which constructs graph edges from scalar reward differences; ii) *Policy-Graph*, which uses the last-token hidden states of the policy model; and iii) *Shuffled RM-Graph*, which randomly permutes the correspondence between responses and RM features.

As shown in Figure 3, *Scalar-Graph* performs similarly to vanilla GRPO, with slight drops on Arena-Hard and only marginal gains on AlpacaEval. This suggests that graph construction based solely on scalar rewards provides little additional information beyond the reward signal already used by GRPO. *Policy-Graph* achieves moderate improvements, indicating that policy-model representations capture some local semantic structure among responses. However, the remaining gap to

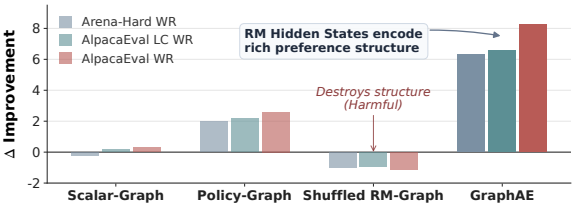


Figure 3: Ablation of graph construction strategies.

GraphAE suggests that semantic similarity alone is insufficient for constructing an effective preference graph. In contrast, *Shuffled RM-Graph* degrades performance below GRPO across all metrics, demonstrating that an incorrect neighborhood structure can actively harm optimization. Overall, these results show that the gains of GraphAE primarily stem from the preference-relevant local geometry encoded in RM hidden states, rather than from graph connectivity, feature dimensionality, or the use of a graph structure alone.

## 6.2 Mechanism Analysis

To better understand why GraphAE improves performance, we conduct a series of analyses on its underlying behavior:

**Variance Reduction.** We compare training steps of GRPO under the same training settings. For each training step, we compute the standard deviation of rewards within each rollout group and track its moving average throughout training. A lower value means that rewards within the same rollout group are less variable. As illustrated in Figure 4, we observe that GraphAE reduces the

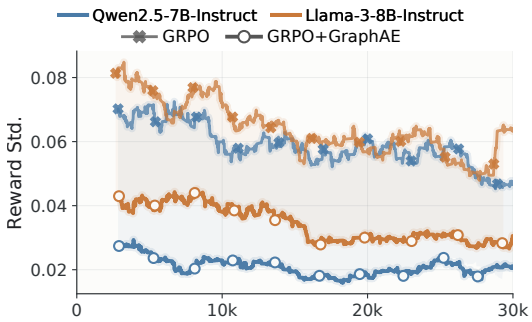


Figure 4: Same budget training dynamics of group reward std. over the first 30k training steps on GRPO.

within-group reward standard deviation, compared with the original scalar reward based optimization. This suggests that GraphAE provides smoother and less noisy rewards, enhancing the training stability.

**Denoising Without Losing Preference Structure.** After observing more stable within-group rewards during training, we further ask whether GraphAE preserves the underlying preference structure while reducing reward noise. We analyze 120 rollout groups with RM features. For each group, we compare the original reward signal with the GraphAE signal from Eq. 10 using i) centered-signal standard deviation, $\mathrm{Std}(s) = \left(\frac{1}{G}\sum_{i=1}^{G}(s_i - \bar{s})^2\right)^{1/2}$, where $\bar{s} = \frac{1}{G}\sum_{i=1}^{G} s_i$; and ii) graph total variation, $\mathrm{GTV}(s) = \frac{1}{G^2}\sum_{i=1}^{G}\sum_{j=1}^{G} K_{ij}(s_i - s_j)^2$, where $K_{ij}$ denotes the RM hidden-state graph edge weight. We report both metrics after normalization by the corresponding value of the original scalar reward signal. Figure 5 shows ten randomly sampled results, and the average value across 120 groups.

GraphAE substantially reduces both quantities, showing that it effectively suppresses local reward noise. However, the denoised signal still remains highly aligned with the original reward structure: the average pairwise sign-flip rate is only 6.25%, and the cosine alignment between the original rewards and GraphAE remains 0.905. These results suggest that GraphAE reduces noisy local fluctuations without collapsing the preference geometry.

**Stronger Benefits for High-Variance Group.** Since GraphAE reduces local reward fluctuations without destroying preference structure, we next examine where this denoising effect brings the largest benefit. We bucket matched rollout groups by their baseline reward standard deviation under

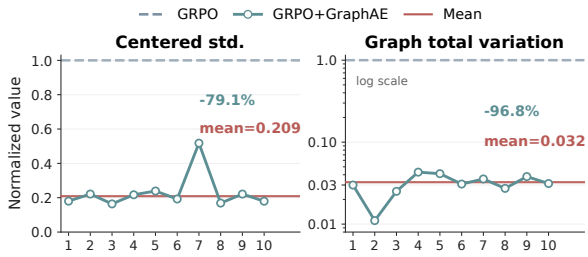


Figure 5: Denoising effect of GraphAE under GRPO.

| Baseline Reward std. | Reward Gain | Positive Rate |
|---|---|---|
| <2.5 | -1.4 | 38.1% |
| 2.5∼5.0 | -1.0 | 45.4% |
| 5.0∼7.5 | +0.6 | 54.0% |
| >7.5 | **+10.4** | **70.8%** |

Table 2: Reward gains under different variance levels. Positive rate is the fraction of groups in each bucket where GraphAE increases the group mean reward.

vanilla GRPO, and measure the change in group mean reward after applying GRPO + GraphAE on the same prompts. Table 2 shows that GraphAE's gain is concentrated in high-variance groups. Low-variance groups receive limited or slightly negative gains, while the highest-variance bucket shows a clear positive improvement. This suggests that GraphAE is most useful when scalar rewards are noisy and credit assignment is unreliable, rather than uniformly smoothing all groups.

### 6.3 Robustness and Efficiency

**Reward Model Generalization.** We next test whether the preference geometry exploited by GraphAE is specific to one RM. To this end, we conduct experiments using two additional RMs, *Skywork-Reward-V2-Qwen3-8B* (Liu et al., 2025a) and *Eurus-RM-7B* (Yuan et al., 2024) and further report results across multiple group-based RL algorithms, including GRPO, GSPO, and RLOO. In addition to downstream performance, we measure the correlation between RM representation distances and reward differences, $\rho(d_h, |\Delta r|)$, as a proxy for how strongly preference structure is encoded in the representation space.

As shown in Table 3, hidden state geometry consistently correlates with reward gaps across all RMs, indicating that preference structure is broadly encoded in RM representations. Importantly, the strength of this correlation closely aligns with downstream gains: models with better represen-

| RM | $\rho(d_h, \lvert\Delta r\rvert)$ | GRPO | GSPO | RLOO |
|---|---|---|---|---|
| Llama-3.1-8B | 0.873 | +6.3 / +8.27 | +4.0 / +5.82 | +4.4 / +7.50 |
| Qwen3-8B | 0.861 | +5.8 / +7.52 | +4.2 / +6.05 | +4.1 / +6.88 |
| Eurus-RM-7B | 0.712 | +2.9 / +4.73 | +2.1 / +3.66 | +2.0 / +3.90 |

Table 3: Reward model generalization across different RMs and RL algorithms.

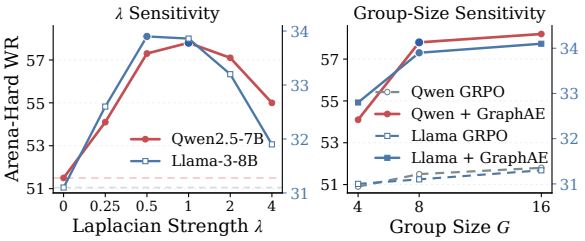


Figure 6: Robustness to the main hyperparameters of GraphAE on Arena-Hard under GRPO.

tations yield larger improvements under GraphAE, while weaker RMs lead to reduced but still positive gains. We further observe that GraphAE improves performance consistently across GRPO, GSPO, and RLOO, suggesting that the benefit is not tied to a specific optimization algorithm. Combined with the consistent trends across different RMs, these results indicate that GraphAE exploits a general and model-agnostic structural property of RM representations rather than relying on particular RMs or RL formulations.

**Robustness to Hyperparameters.** We evaluate the robustness of GraphAE to its main hyperparameters on Arena-Hard under GRPO. Figure 6 varies the Laplacian strength $\lambda$, and varies the rollout group size $G$ for two backbones. First, GraphAE is not sensitive to a narrow choice of $\lambda$. Both backbones improve substantially once $\lambda > 0$ and remain strong over a broad range. The best value is attained near $\lambda = 1$ for Qwen and near $\lambda = 0.5$ for Llama, but performance around these optima is very similar, indicating that GraphAE does not rely on a sharp sweet spot. Second, GraphAE is also robust to group size. Larger groups bring slightly stronger improvements, where richer local neighborhoods make graph-based credit assignment more effective. These results suggest that GraphAE is stable across reasonable hyperparameter choices rather than relying on fragile tuning.

**Overhead.** Table 4 shows the estimator overhead. Computing the GraphAE on 4096-dimensional rollout embeddings yields 0.80 ms latency per group at

| Group Size | Compute Latency | Scratch Memory |
|---|---|---|
| $G = 8$ | 0.80 ms | 1.34 KB |
| $G = 16$ | 3.29 ms | 5.19 KB |
| $G = 32$ | 15.27 ms | 20.38 KB |

Table 4: Compute overhead of the GraphAE.

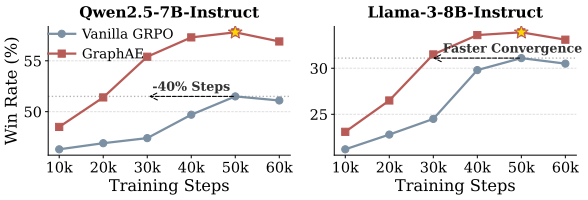


Figure 7: Checkpoint sample efficiency analysis.

$G = 8$, and 15.27 ms at $G = 32$, while the scratch memory stays below 21 KB per group. Since the estimator only solves a small $G \times G$ linear system after reward computation, its cost is negligible relative to policy and RM forward passes. Taken together, these results show that GraphAE is both more sample efficient and lightweight in practice.

**Checkpoint Efficiency.** To investigate whether GraphAE accelerates policy convergence, we save checkpoints every 10k training steps and evaluate them on Arena-Hard. As shown in Figure 7, GraphAE improves alignment quality throughout the entire training trajectory: it robustly outperforms vanilla GRPO at every evaluated checkpoint. For Qwen, GraphAE reaches the final GRPO peak score of 51.5 at merely 30k steps, whereas vanilla GRPO requires 50k steps to reach this level, corresponding to a 40% reduction in optimization steps. This accelerated convergence pattern generalizes remarkably to Llama, where GraphAE attains the GRPO peak in fewer than 30k steps and eventually climbs to an unprecedented 33.9. Moreover, both models show that our method improves both final performance and convergence speed under the same compute budget.

## 7 Conclusion

We propose to leverage reward model hidden states as graph signals to compute advantages. By propagating rewards over graphs, our method GraphAE captures local correlations, reduces variance, and generalizes group-based RL algorithms. It bridges the gap between scalar reward estimation and structured inference, offering a more information rich, stable, and effective RLHF paradigm.

## Limitations

GraphAE relies on the quality of the reward model representations to construct the similarity graph. When the RM hidden states contain weak or noisy preference structure, the effectiveness of graph-based propagation may be reduced. In addition, GraphAE introduces a small amount of extra computation for graph construction and matrix solving within each rollout group. Although this overhead is lightweight in our experiments, scaling to substantially larger group sizes may require more efficient approximations in future work.

## References


Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of ACL*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Proceedings of NeurIPS*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. In *Proceedings of ICML*.

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

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Jiayi Fu, Xuandong Zhao, Chengyuan Yao, Heng Wang, Qi Han, and Yanghua Xiao. 2025. Reward shaping to mitigate reward hacking in rlhf. *arXiv preprint arXiv:2502.18770*.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

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

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. In *Proceedings of EMNLP*.

Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. 2023. Reward design with language models. *arXiv preprint arXiv:2303.00001*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Karl Cobbe, Mark Chen, Heewoo Jun, John Schulman, and Ilya Sutskever. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, et al. 2025a. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*.

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*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Proceedings of NeurIPS*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Proceedings of NeurIPS*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of NeurIPS*.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *Proceedings of ICLR*.

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

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

Alex Smola and Risi Kondor. 2003. Kernels and regularization on graphs. In *Proceedings of COLT*.

Richard S Sutton, Andrew G Barto, et al. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of NeurIPS*.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. 2024a. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Chenglong Wang, Yang Gan, Yifu Huo, Yongyu Mu, Qiaozhi He, Murun Yang, Tong Xiao, Chunliang Zhang, Tongran Liu, and Jingbo Zhu. 2024b. Lrhp: Learning representations for human preferences via preference pairs. *arXiv preprint arXiv:2410.04503*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024c. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of EMNLP*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of NeurIPS*.

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2023. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment. *arXiv preprint arXiv:2310.00212*.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

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

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. Regularizing hidden states enables learning generalizable reward model for llms. In *Proceedings of NeurIPS*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. In *Proceedings of NeurIPS*.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. 2024. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*.

Chuheng Zhang, Wei Shen, Li Zhao, Xuyun Zhang, Xiaolong Xu, Wanchun Dou, and Jiang Bian. 2025. Policy filtration for rlhf to mitigate noise in reward models. In *Proceedings of ICML*.

Shuyi Zhang, Wei Shi, Sihang Li, Jiayi Liao, Tao Liang, Hengxing Cai, and Xiang Wang. 2026. Interpretable reward model via sparse autoencoder. In *Proceedings of AAAI*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of ACL*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of NeurIPS*.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of ICML*.

## A Datasets and Evaluation

We follow each benchmark's evaluation protocol. For Arena-Hard-v0.1 (Li et al., 2024), we report the win rate (WR), and CI denotes the 95% confidence interval of the Arena-Hard win rate, which measures the statistical uncertainty of the evaluation result. We use the official library [3], adopting the default decoding configuration and comparing the WR against GPT-4-0314, using GPT-4.1 as the judge. For AlpacaEval 2.0 (Dubois et al., 2024), we report the WR and length-controlled (LC) WR. We follow the default setting [4], evaluating the WR against GPT-4-Turbo using weighted-alpaca-eval-gpt4-turbo as annotator. For MT-Bench (Zheng et al., 2023), we report the scores on two turns and their average. We follow an open codebase [5] and update the chat format for compatibility with the Llama-3.1-8B-Instruct chat template. We use GPT-4-Turbo as the judge to rate the quality of responses with scalar scores ranging from 1 to 10.

## B Prompt Construction for RL Training

For RL training, we use the prompts from the UltraFeedback (Cui et al., 2024) training split. Each training example contains either a structured conversation field or a plain prompt field. When the structured messages are available, we directly use the original role-content messages from the dataset. Otherwise, we convert the plain prompt into a single user message. For both Qwen2.5-7B-Instruct (Yang et al., 2025) and Llama-3-8B-Instruct (Grattafiori et al., 2024), we prepend the same system instruction: "*You are a helpful, honest, and harmless assistant.*" The resulting conversation is then formatted with the corresponding model-specific chat template, with the generation prompt enabled, and used as the policy input for rollout generation. After the policy samples a response, we append this response as the assistant message to the same conversation and format it with the reward model chat template. The reward model then produces both the scalar reward and the hidden-state representation used by GraphAE. This ensures that policy generation and reward evaluation are based on the same underlying conversation context, while each model still follows its own official chat format.

[3] https://github.com/lmarena/arena-hard-auto

[4] https://github.com/tatsu-lab/alpaca_eval

[5] https://github.com/fanqiwan/FuseAI/tree/main/FuseChat-3.0/FuseEval/IF-Eval/MT-Bench

## C Effect of Generic Reward Smoothing

To further distinguish GraphAE from generic reward regularization, we study the effect of the smoothing coefficient $\alpha$ in the reward smooth baseline ($\tilde{r}_i = (1-\alpha)r_i + \alpha\bar{r}$) on final performance. The Qwen + GRPO results are shown in Table 5. We observe that a small amount of smoothing brings only limited benefit, while stronger smoothing consistently hurts final performance. As $\alpha$ increases, the scalar rewards within each rollout group become less distinguishable, which weakens useful preference contrasts among responses. This suggests that indiscriminately suppressing reward variance is not sufficient, and can even be harmful when the original reward differences contain meaningful preference information.

In contrast, GraphAE improves performance substantially over both vanilla GRPO and all reward smoothing variants. The key difference is that GraphAE does not collapse all rewards toward the group mean. Instead, it propagates reward information locally according to the similarity graph induced by RM hidden states. Therefore, responses that are close in the RM representation space can share information, while unrelated responses are not forced to receive similar rewards. This selective, structure-aware denoising allows GraphAE to reduce noisy local fluctuations without destroying useful preference distinctions, explaining why it outperforms generic reward smoothing.

## D Generalization of Variance Reduction

We further examine whether the variance-reduction effect of GraphAE generalizes beyond GRPO. Specifically, we compare the training dynamics of GSPO and RLOO under the same training settings, and track the moving average of the within-group reward standard deviation throughout training. As shown in Figure 8 and Figure 9, GraphAE consistently reduces the reward standard deviation compared with the original scalar-reward based optimization. This trend holds for both Qwen2.5-7B-Instruct and Llama-3-8B-Instruct, indicating that the stabilizing effect is not tied to a particular group-based optimizer. The reduction is especially clear in the early and middle stages of

| Method | Arena-Hard | AlpacaEval LC | AlpacaEval WR | MT-Bench T1 | MT-Bench T2 | MT-Bench Avg |
|---|---|---|---|---|---|---|
| Qwen2.5-7B-Instruct | 45.7 | 33.74 | 33.41 | 8.23 | 6.96 | 7.59 |
| GRPO | 51.5 | 36.26 | 35.94 | **8.66** | 7.01 | 7.84 |
| Reward Smooth ($\alpha = 0.1$) | 52.3 | 36.11 | 35.72 | 8.61 | 7.00 | 7.81 |
| Reward Smooth ($\alpha = 0.3$) | 51.0 | 36.40 | 36.12 | 8.61 | 7.02 | 7.82 |
| Reward Smooth ($\alpha = 0.5$) | 49.9 | 35.12 | 34.67 | 8.52 | 6.95 | 7.74 |
| Reward Smooth ($\alpha = 0.7$) | 47.7 | 34.51 | 33.98 | 8.40 | 6.88 | 7.64 |
| GraphAE | **57.8** | **42.85** | **44.21** | 8.60 | **7.10** | **7.85** |

Table 5: Comparison between GraphAE and generic reward smoothing with different coefficient $\alpha$.

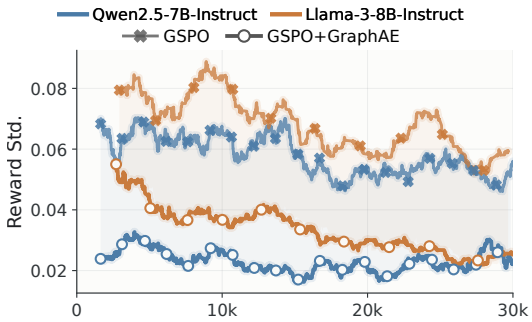


Figure 8: Same budget training dynamics of group reward std. over the first 30k training steps on GSPO.

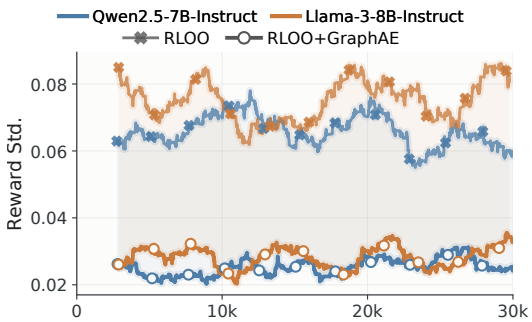


Figure 9: Same budget training dynamics of group reward std. over the first 30k training steps on RLOO.

training, where vanilla GSPO and RLOO exhibit larger fluctuations in group rewards. By propagating rewards over the RM representation graph, GraphAE produces smoother within-group reward signals while keeping the underlying optimization procedure unchanged. These results complement the GRPO analysis and suggest that GraphAE provides a general variance-reduction mechanism for group-based RLHF algorithms.

## E Case Study

We further provide several qualitative examples to show how GraphAE changes individual rewards. Table 6 reports four Qwen + GRPO rollout cases. Each case contains a target response, its nearest neighbor in the RM hidden space, the original scalar RM reward, and the refined GraphAE reward. These examples reveal a clear pattern. The scalar RM sometimes gives a reward that is inconsistent with nearby responses. In the first two cases, the target response receives a relatively high scalar reward, but its nearest neighbors suggest that this reward is over-estimated. GraphAE therefore pulls the reward down. In the last two cases, the target response receives a very low scalar reward, although it is close to higher-reward responses in the RM hidden space. GraphAE then raises its reward toward the local neighborhood.

This shows why GraphAE works. It does not simply smooth all rewards toward the group mean. Instead, it uses the RM hidden space to identify which responses should influence each other. If a response is close to strong neighbors, GraphAE can recover an under-estimated reward. If a response is close to weaker or more cautious neighbors, GraphAE can suppress an over-estimated reward. In this way, GraphAE reduces local reward noise while keeping the preference structure encoded by the RM representations. The examples therefore provide direct evidence that RM hidden states contain useful information beyond the scalar reward, and that GraphAE can use this information to produce more reliable advantage estimates.

| Qualitative examples of GraphAE reward correction under Qwen2.5-7B-Instruct + GRPO | |
|---|---|
| **Case 1: Over-rewarded scientific claim is pulled down** | |
| **Prompt** | Why do we absorb information faster when our brain wave frequency is four-eight Hz (Hertz)? |
| **Target response** | The idea that we absorb information faster when our brain wave frequency is in the 4–8 Hz range is related to the theta brainwave state. Theta waves are typically associated with deep relaxation, dream states, and creative thinking. Here is a more detailed explanation: Theta waves are commonly observed in the brain during deep relaxation, meditation, and early stages of sleep. This state is often associated with enhanced creativity, problem-solving, and deep learning. |
| **Nearest neighbor** | The idea that brainwave frequencies specifically at 4–8 Hz can enhance information absorption is not supported by current scientific research. Brainwave frequencies in this range are typically associated with relaxed states or drowsiness, but this does not imply faster information absorption. |
| **Correction** | RM reward: 1.760 $\rightarrow$ GraphAE reward: 1.629 ($\Delta = -0.130$). The nearest neighbor has RM reward 1.577, hidden distance $d_h = 0.058$, and edge weight $w = 0.839$. |
| **Takeaway** | Prompt asks why information is absorbed faster at 4–8 Hz brain waves. The target response confidently explains this as a theta-wave effect. Neighbor response notes that such a learning-speed claim is not supported by current scientific evidence. The scalar RM gives a high reward to an over-confident scientific explanation. A close hidden-space neighbor is more cautious and better calibrated. GraphAE uses this local neighbor evidence to pull the reward down. |
| **Case 2: Incorrect cumulative-sum table is pulled down** | |
| **Prompt** | Given the natural number 5, compute the summation of all natural numbers from 1 up to and including 5. Please display the output in a tabular format with two columns: the first column should contain the individual natural numbers and the second column should display their corresponding sum. Additionally, please provide the source code for your solution. |
| **Target response** | Sure! Here is the tabular format you requested: Natural Number / Sum of Numbers from 1 to 5: 1 / 15, 2 / 15, 3 / 15, 4 / 15, 5 / 15. To compute the sum of all natural numbers from 1 to 5, we can use the formula for the sum of the first $n$ natural numbers. |
| **Nearest neighbor** | Certainly! Below is the tabular format as requested, followed by the source code in Python. Number / Sum: 1 / 1, 2 / 3, 3 / 6, 4 / 10, 5 / 15. The Python code computes the summation of all natural numbers from 1 to 5 and displays them in the tabular format. |
| **Correction** | RM reward: 1.400 $\rightarrow$ GraphAE reward: 1.361 ($\Delta = -0.039$). The nearest neighbor has RM reward 1.322, hidden distance $d_h = 0.039$, and edge weight $w = 0.903$. |
| **Takeaway** | Prompt asks for cumulative sums from 1 to 5. The target table incorrectly sets every row to 15. Neighbor response gives the correct cumulative sums 1, 3, 6, 10, 15. The target response contains a clear table error: every cumulative-sum row is set to 15. A close neighbor provides the correct cumulative sequence. GraphAE therefore reduces the reward of the locally flawed response. |
| **Case 3: Under-rewarded evidence response is recovered** | |
| **Prompt** | What is the evidence and quotes on the impact the founders' well-being has on the chances of success of their startup? |
| **Target response** | The well-being of startup founders can significantly impact the success of their ventures. This impact is supported by various studies, anecdotal evidence, and quotes from successful entrepreneurs. Here are some key points and quotes: mental health and decision-making, stress and burnout, resilience, and leadership quality. |
| **Nearest neighbor** | The well-being of founders can indeed have a significant impact on the success of their startups. Here are some key points, evidence, and quotes that support this idea, including evidence from research, founder mental health, decision-making, team morale, and long-term startup performance. |
| **Correction** | RM reward: 0.874 $\rightarrow$ GraphAE reward: 1.474 ($\Delta = +0.600$). The nearest neighbor has RM reward 1.701, hidden distance $d_h = 0.031$, and edge weight $w = 0.757$. |
| **Takeaway** | Prompt asks for evidence and quotes about how founder well-being affects startup success. The target gives structured evidence and quotes but receives a low scalar reward. Neighbor response has a similar evidence-oriented structure and receives a high scalar reward. The target response is structurally close to high-reward evidence-based responses but receives an anomalously low scalar reward. GraphAE recovers this under-estimated response by propagating information from nearby high-reward neighbors. |
| **Case 4: Under-rewarded narrative response is recovered** | |
| **Prompt** | Write a first-person narrative about a young woman who discovers she can generate illusions that seem real. However, due to early onset schizophrenia, she struggles to differentiate between her illusions and reality. Incorporate elements of psychological and supernatural tension. |
| **Target response** | I wake up in a fog, the same way I always do. My name is Elara, and I am twenty-two. The first thing I notice is the cold, damp air against my skin. I open my eyes to a room I cannot quite place, with walls painted in a sickly shade of green. It is dim, and the only light comes from a flickering bulb hanging from the ceiling. The room feels off. |
| **Nearest neighbor** | I wake up in the middle of the night to the sound of my own breathing. My heart races, and I blink furiously as I try to make sense of the dizzying chaos that surrounds me. The walls of my apartment seem to shift and twist, becoming a kaleidoscope of colors and shapes. |
| **Correction** | RM reward: 0.763 $\rightarrow$ GraphAE reward: 1.151 ($\Delta = +0.389$). The nearest neighbor has RM reward 1.296, hidden distance $d_h = 0.068$, and edge weight $w = 0.879$. |
| **Takeaway** | Prompt asks for a first-person narrative about a young woman with realistic illusions and schizophrenia. The target is coherent and matches the requested style but receives a low scalar reward. Neighbor response has a similar first-person psychological narrative style. The target response is close to other first-person psychological narratives in RM hidden space, but its scalar reward is much lower. GraphAE raises the reward toward the local neighborhood, correcting a likely scalar-reward outlier. |

Table 6: Qualitative examples of GraphAE reward correction under Qwen2.5-7B-Instruct + GRPO. Each case shows a real prompt, a target response excerpt, its nearest neighbor in RM hidden space, and the reward change from the scalar RM reward to the GraphAE reward.