

---

# Mitigating Reward Hacking in RLHF via Bayesian Non-negative Reward Modeling

---

Anonymous Authors<sup>1</sup>

## Abstract

Reward models learned from human preferences are central to aligning large language models (LLMs) via reinforcement learning from human feedback, yet they are often vulnerable to reward hacking due to noisy annotations and systematic biases such as response length or style. We propose Bayesian Non-Negative Reward Model (BNRM), a principled reward modeling framework that integrates non-negative factor analysis into Bradley–Terry (BT) preference model. BNRM represents rewards through a sparse, non-negative latent factor generative process that operates at two complementary levels: instance-specific latent variables induce disentangled reward representations, while sparsity over global latent factors acts as an implicit debiasing mechanism that suppresses spurious correlations. Together, this disentanglement-then-debiasing structure enables robust uncertainty-aware reward learning. To scale BNRM to modern LLMs, we develop an amortized variational inference network conditioned on deep model representations, allowing efficient end-to-end training. Extensive empirical results demonstrate that BNRM substantially mitigates reward over-optimization, improves robustness under distribution shifts, and yields more interpretable reward decompositions than strong baselines.

## 1. Introduction

With the advent of Large Language Models (LLMs), reinforcement learning from human feedback (RLHF) has emerged as a central paradigm for aligning model behavior with human values (Stiennon et al., 2020; Ouyang et al., 2022). At the core of RLHF lies the reward models (RMs),

which distill noisy human preference annotations into a differentiable signal for policy optimization (Ziegler et al., 2019; Ouyang et al., 2022; Gao et al., 2023; Liu et al., 2024a). Despite the empirical success, learning reward models that are both reliable and generalizable remains a fundamental challenge (Casper et al., 2023; Touvron et al., 2023a; Liu et al., 2024a). A prevalent failure mode, commonly referred to as reward over-optimization or reward hacking, occurs when the policy exploits spurious correlations encoded in the proxy RM, yielding behavior that scores highly under the learned reward but deviates from true human objectives (Gao et al., 2023; Coste et al., 2023).

A primary source of reward over-optimization in reward modeling is reward misgeneralization (Casper et al., 2023; Miao et al., 2024), whereby RMs extrapolate incorrectly beyond the training distribution and thus form poor proxies for true human preferences, which arises from two interacting factors: on the one hand, noisy, subjective, and heterogeneous human annotations (Gao et al., 2024); on the other hand, the tendency of deep neural networks to exploit shortcut features and learn spurious correlations that bypass their intended semantic objectives (Geirhos et al., 2020; Zhang et al., 2016). As a result, RMs often overemphasize superficial cues, like response length (Singhal et al., 2023), phrasing patterns, or stylistic artifacts (Zhang et al., 2025; Miao et al., 2024; Wang et al., 2024), that are easy to optimize but misaligned with genuine human intent, ultimately undermining the reliability and safety of RLHF (Casper et al., 2023; Gao et al., 2023).

To mitigate reward hacking, recent work has explored Bayesian principles to alleviate overfitting in highly over-parameterized reward models (Wang & Yeung, 2016; Wilson & Izmailov, 2020). Practical approaches such as reward model ensembling (Lakshminarayanan et al., 2017; Coste et al., 2023; Eisenstein et al., 2023; Ramé et al., 2024; Zhang et al., 2024a) improve robustness by aggregating multiple predictors, but at the cost of training and maintaining several large-scale models, resulting in substantial computational and memory overhead. Beyond ensembles, information-theoretic methods introduce variational information bottleneck objectives to suppress spurious latent features (Miao et al., 2024). However, these

---

<sup>1</sup>Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

approaches rely on implicit notions of relevance and often fail to explicitly disentangle semantic intent from nuisance factors (Yang et al., 2022). Other lines of work address robustness by correcting specific biases through supervised interventions (Chen et al., 2024), most notably response-length bias, but such methods typically generalize poorly beyond narrowly defined settings.

To address these limitations more fundamentally, we revisit sparsity-aware Bayesian models (SBMs) (Wipf & Rao, 2004), such as non-negative factor analysis (NFA) (Blei et al., 2003; Zhou et al., 2012). These models offer two key advantages. First, probabilistic latent factor modeling enables explicit treatment of both epistemic and aleatoric uncertainty, which is essential when learning from noisy and inconsistent human feedback (Gao et al., 2023). Second, sparsity acts as an implicit regularizer that discourages reliance on spurious or non-invariant features, thereby improving robustness to shortcut correlations (Zhou et al., 2022). Moreover, the inherent non-negativity constraints in these models induce disentangled, parts-based representations, yielding substantially improved interpretability compared to dense and opaque embeddings (Lee & Seung, 1999; Nguyen et al., 2016).

Inspired by these insights, we revisit RMs from an NFA perspective and propose the Bayesian Non-Negative Reward Model (BNRM). BNRM integrates the probabilistic structure of NFA with the expressive representations of large language models, formulating reward learning as a stochastic generative process over latent, non-negative reward factors that explicitly capture uncertainty. Besides, as illustrated in Figure 1, BNRM departs from conventional dense reward functions by imposing sparsity-aware structure: local sparsity promotes disentangled representations of semantic preference factors, while global sparsity suppresses spurious correlations and facilitates systematic debiasing. To scale BNRM to modern LLMs, we develop an amortized variational inference network conditioned on backbone representations. By parameterizing the variational posterior with a reparameterizable Weibull distribution (Zhang et al., 2020), BNRM enables end-to-end training via standard backpropagation. Extensive experiments demonstrate that BNRM substantially improves robustness to reward over-optimization, enhances interpretability, and generalizes better under distribution shifts. Our contributions are summarized as follows:

- We propose **BNRM**, a Bayesian non-negative reward modeling framework that jointly enforces sparsity and models uncertainty to mitigate reward hacking.
- We introduce a scalable amortized variational inference scheme with reparameterizable Weibull posteriors, enabling efficient integration with large language models.
- We empirically show that BNRM outperforms strong

baselines in robustness, interpretability, and out-of-distribution generalization.

## 2. Related Work

**Mitigating Reward Overoptimization in RLHF.** Reward models are prone to overfitting on training data, leading to reward overoptimization where policies exploit the learned proxy instead of the true human objective (Gao et al., 2023). Existing mitigation efforts can be broadly categorized. One major line of work focuses on *uncertainty-aware modeling*, primarily through computationally intensive model ensembles or by adding post-hoc uncertainty penalties to the reward signal (Coste et al., 2023; Eisenstein et al., 2023; Lin et al., 2023; Zhang et al., 2024b; Yang et al., 2024a; Lou et al., 2024; Sun et al., 2025). Another category involves *data and policy-level regularization*, using techniques like label smoothing, adaptive margin, and adding SFT-based constraints during policy optimization (Wang et al., 2024; Liu et al., 2024c; Li et al., 2025). While these methods effectively mitigate the symptoms, our work posits that the problem’s root is the reward model’s reliance on dense, non-interpretable features. We therefore propose a fundamentally different approach that reshapes the reward representation itself by injecting the principles of Non-Negative Factor Analysis (NFA) into the reward modeling process, allowing us to directly learn a sparse and robust reward function that is inherently less susceptible to spurious correlations.

**Non-negative Factor Analysis.** NFA models, such as latent Dirichlet allocation (Blei et al., 2003) and Poisson factor analysis (Zhou et al., 2012), are classical probabilistic tools for learning parts-based representations from data. Their ability to uncover sparse latent structures has made them highly effective for tasks like topic modeling (Zhou et al., 2015). While recent work has explored integrating NFA’s benefits into general deep neural networks (Hu et al., 2025), our contribution is the principled synthesis of NFA’s generative structure within the modern RLHF pipeline. Distinct from prior studies, we leverage NFA’s sparse inductive bias specifically to combat reward over-optimization (Gao et al., 2023; Zhou et al., 2022) and enhance robustness against distributional shifts (Wilson & Izmailov, 2020).

## 3. Preliminaries

### 3.1. Scalar Preference Reward Model in RLHF

A reward model (RM) aims at learning human preferences based a parameterized model  $r_\phi$  on a human preference dataset  $\mathcal{D} = \{(x, y_1, y_2)\}$ , where  $x$  is a user input prompt,  $y_1$  and  $y_2$  are the preferred (chosen) and non-preferred (rejected) responses, by minimizing a ranking loss following

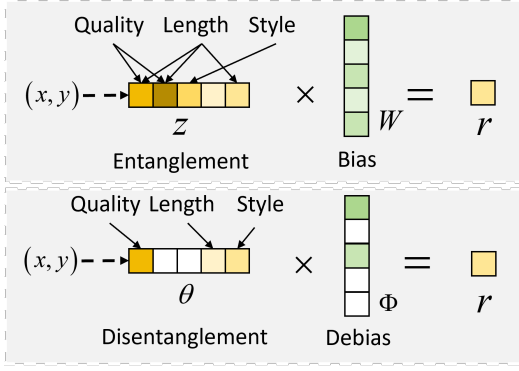


Figure 1. Motivation on disentanglement and debiasing for alleviating spurious correlations in reward modeling.

the Bradley-Terry (BT) objective (Bradley & Terry, 1952):

$$p(\mathbf{y}_1 \succ \mathbf{y}_2 | \mathbf{x}) = \sigma(r_\phi(\mathbf{x}, \mathbf{y}_1) - r_\phi(\mathbf{x}, \mathbf{y}_2)),$$

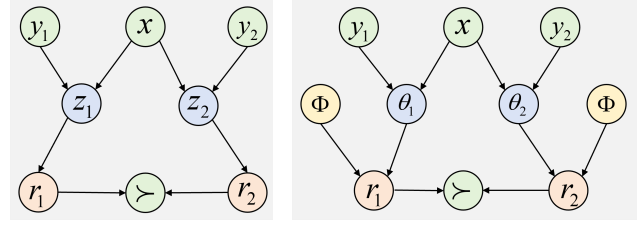
where  $\sigma(\cdot)$  is the Sigmoid function. The optimized reward model serves as a proxy for human preferences, enabling the subsequent RL fine-tuning phase.  $r_\phi$  typically consists of a backbone feature extractor  $f$  and a final linear head  $W_{\text{bt}}$ , where the backbone  $f$  is often initialized with a pretrained LLM parameter  $W_{\text{llm}}$  and projects the prompt and the corresponding response into a factor vector, denoted as  $\mathbf{z} = f(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{1 \times d_{\text{model}}}$ . A new linear head,  $W_{\text{bt}} \in \mathbb{R}^{d_{\text{model}} \times 1}$ , is then added to project the feature representation  $\mathbf{z}$  into a scalar reward value, expressed as

$$r_\phi(\mathbf{x}, \mathbf{y}) = \mathbf{z}^T W_{\text{bt}}, \mathbf{z} = f(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{1 \times d_{\text{model}}}.$$

The training of the RM involves fine-tuning the parameters of  $f$  and  $W_{\text{bt}}$  on the preference dataset  $\mathcal{D}$ . However, this standard implementation suffers from two fundamental limitations: **Deterministic and Overconfident Scoring**: The model produces a single, deterministic reward, failing to capture uncertainty in human preferences and leading to over-optimization. **Specious Correction**: The dense, “black-box” nature of the features  $\mathbf{z}$  and weights  $W$  makes the model prone to reward hacking by exploiting spurious correlations, as shown in Fig.1(Top). The ultimate goal is to use this learned reward signal  $r_\phi$  to optimize the policy in the final RL stage. A flawed, deterministic, and easily hackable RM inevitably leads to a misaligned policy, underscoring the necessity for a more robust reward modeling framework, which we introduce in the subsequent sections.

### 3.2. Non-negative Factor Analysis (NFA)

Non-negative factor analysis methods, such as Poisson factor analysis (PFA) (Zhou et al., 2012), are widely used as topic models (Blei & Lafferty, 2009). They impose non-negative stochastic latent variables on the model parameters to learn interpretable, parts-based representations of data. Specifically, representing document  $\mathbf{x}$  as a BOW vector  $\mathbf{b} \in \mathbb{Z}^V$ , where  $\mathbb{Z} = \{0, 1, \dots\}$  and  $V$  is the vocabulary



(a) Standard BT

(b) BNRM (Ours)

Figure 2. Graphical model representations. (a) The standard BT model; (b) our proposed BNRM. Here,  $\mathbf{x}$  denotes the prompt,  $\mathbf{y}_1, \mathbf{y}_2$  are candidate responses, and nodes represent the predictive process for the preference  $\mathbf{y}_1 \succ \mathbf{y}_2$ .

size, PFA models  $\mathbf{b}$  under the Poisson likelihood as

$$\mathbf{b} \sim \text{Poisson}(\Phi\theta), \theta \sim \text{Gamma}(\alpha, 1). \quad (1)$$

Here, the matrix  $\Phi \in \mathbb{R}_+^{V \times K}$  is the dictionary, where each column represents a topic as a distribution over words. The vector  $\theta \in \mathbb{R}_+^K$  contains the document-specific topic proportions (i.e., document features) that represent the strength of each topic in the document. Benefiting from the sparsity of latent variables, which is often encouraged by placing Gamma priors with hyperparameter  $\alpha$  on  $\theta$ , NFA effectively handles overdispersed data and exhibits strong generalization ability.

## 4. The Bayesian Non-negative Reward Model

Our methodology is rooted in a Bayesian perspective, comprising a generative process that describes our idealized assumptions about how preferences are formed, and an inference process that details how we approximate this model in practice using deep neural networks.

### 4.1. BT from a Bayesian Viewpoint

The BNRM framework, introduced intuitively in the previous section, can be formally derived as a hierarchical Bayesian extension of the standard Bradley-Terry (BT) model (Bradley & Terry, 1952). This viewpoint clarifies how BNRM systematically addresses the limitations of the deterministic approach. From a Bayesian perspective, as shown in Figure 2a, the standard BT model is a special case of the following integral formulation:

$$p(\mathbf{y}_1 \succ \mathbf{y}_2 | \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2) = \int_{\mathbf{z}_1, \mathbf{z}_2} p(\mathbf{y}_1 \succ \mathbf{y}_2 | \mathbf{z}_1, \mathbf{z}_2) p(\mathbf{z}_1 | \mathbf{y}_1, \mathbf{x}) p(\mathbf{z}_2 | \mathbf{y}_2, \mathbf{x}) d\mathbf{z}_1 d\mathbf{z}_2. \quad (2)$$

Here, the deterministic nature of the underlying neural network  $f$  means that the conditional probability  $P(\mathbf{z} | \mathbf{y}, \mathbf{x})$  is a Dirac delta distribution,  $P(\mathbf{z} | \mathbf{y}, \mathbf{x}) = \delta(\mathbf{z} - f(\mathbf{x}, \mathbf{y}))$ , which offers no mechanism to capture uncertainty. Our BNRM generalizes this model in two steps:

**Modeling Aleatoric Uncertainty.** We first replace the deterministic latent representation  $z$  with a stochastic latent variable  $\theta$ . This allows the model to capture the inherent randomness and ambiguity in human preference data. The preference probability is now marginalized over the distribution of  $\theta$ :

$$p(\mathbf{y}_1 \succ \mathbf{y}_2 | \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2) = \int_{\theta_1, \theta_2} P(\mathbf{y}_1 \succ \mathbf{y}_2 | \theta_1, \theta_2) p(\theta_1 | \mathbf{y}_1, \mathbf{x}) \cdot P(\theta_2 | \mathbf{y}_2, \mathbf{x}) d\theta_1 d\theta_2. \quad (3)$$

**Modeling Epistemic Uncertainty.** To further account for the model’s own uncertainty about the global reward factors, we treat the final layer weights (denoted as  $\Phi$ ) as a global stochastic variable. This leads to the full, formal generative process for BNRM, as shown in Figure 2b:

$$p(\mathbf{y}_1 \succ \mathbf{y}_2 | \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2) = \int_{\theta_1, \theta_2, \Phi} p(\mathbf{y}_1 \succ \mathbf{y}_2 | \theta_1, \theta_2, \Phi) p(\theta_1 | \mathbf{y}_1, \mathbf{x}) p(\theta_2 | \mathbf{y}_2, \mathbf{x}) p(\Phi) d\theta_1 d\theta_2 d\Phi, \quad (4)$$

which final integral represents the complete generative process of our proposed model.

## 4.2. The BNRM Generative Process

Recall that Eq. 4 is designed to systematically capture both aleatoric and epistemic uncertainty in reward modeling. Building on this formulation, we propose a fully probabilistic generative model in which human preferences arise from sparse, non-negative latent factors. This framework replaces the standard deterministic reward formulation,  $r = f(\mathbf{x}, \mathbf{y})^\top W_{\text{br}}$ , with a structured Bayesian alternative that explicitly models uncertainty, disentanglement, and bias. The generative process introduces two complementary sets of latent variables:

**1. Local Sparse Representation (Disentanglement).** For each prompt–response pair  $(\mathbf{x}, \mathbf{y})$ , we introduce a non-negative latent vector  $\theta \in \mathbb{R}_+^K$  that captures the sparse activations of global reward factors specific to  $\mathbf{y}$ . Sparsity in  $\theta$  encourages *instance-level disentanglement* by activating only a small subset of latent factors, thereby improving identifiability and reducing reliance on entangled or spurious features (Zheng et al., 2022).

**2. Global Reward Dictionary (Debiasing).** We further introduce a global dictionary of reward factors  $\Phi \in \mathbb{R}_+^K$ , shared across all data points, which defines a universal non-negative basis for evaluating response quality. Sparsity imposed on  $\Phi$  acts as a *population-level regularizer*, selectively retaining invariant and semantically meaningful factors while suppressing spurious correlations in reward estimation (Zhang et al., 2021; Zhou et al., 2022).

To jointly enforce non-negativity and sparsity, we place Gamma priors, a standard choice in non-negative factor

analysis (Zhou et al., 2012), on both sets of latent variables:

$$\Phi \sim \text{Gamma}(\gamma_0, \delta_0), \quad \theta \sim \text{Gamma}(\alpha_0, \beta_0). \quad (5)$$

Given these latent factors, the scalar reward associated with a response  $(\mathbf{x}, \mathbf{y})$  is generated as

$$r(\mathbf{x}, \mathbf{y}) = \theta^\top \Phi, \quad (6)$$

which yields a non-negative, interpretable reward constructed from sparse factor activations. Finally, for a pair of candidate responses  $(\mathbf{y}_1, \mathbf{y}_2)$  with corresponding rewards  $(r_1, r_2)$ , the observed human preference is generated via a Bradley–Terry likelihood:

$$p(\mathbf{y}_1 \succ \mathbf{y}_2 | r_1, r_2) = \sigma(r_1 - r_2), \quad (7)$$

where  $\sigma(\cdot)$  denotes the logistic sigmoid function. This completes a coherent probabilistic account of how human preferences emerge from sparse, non-negative latent reward factors, naturally supporting uncertainty quantification, disentanglement, and debiasing within a unified Bayesian framework.

## 4.3. Variational Inference and Training Objective

Given the generative formulation of BNRM defined in Eqs. 4 and 5, the exact posterior distributions over the latent variables,  $p(\theta | \mathbf{x}, \mathbf{y})$  and  $p(\Phi | \mathcal{D})$ , are analytically intractable. We therefore resort to variational inference (VI) to obtain a scalable approximation. Specifically, we introduce tractable variational distributions  $q(\theta | \mathbf{x}, \mathbf{y})$  and  $q(\Phi)$  to approximate the true posteriors. Importantly, we re-purpose the deep LLM backbone  $f$  not as part of the generative model, but as a powerful *inference network* (encoder) for amortized inference of the local latent variables. As illustrated in Figure 3, for each prompt–response pair  $(\mathbf{x}, \mathbf{y})$ , the inference proceeds as follows:

We first extract a deterministic, high-dimensional and dense feature representation using the LLM backbone with parameter  $W_{\text{lm}}$ :  $z = f(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d_{\text{model}}}$ . The feature vector  $z$  is then used to parameterize the variational distribution of the corresponding local latent variable  $\theta_i$ . Following prior work on scalable inference for non-negative latent variable models (Zhang et al., 2020), we adopt a Weibull distribution due to its convenient reparameterization and its ability to model sparse, positive random variables:

$$q(\theta | \mathbf{x}, \mathbf{y}) = \text{Weibull}(\mathbf{k}, \boldsymbol{\lambda}), \quad (8)$$

$$(\mathbf{k}, \boldsymbol{\lambda}) = \text{Activation}(zW_{\text{vi}}),$$

where  $W_{\text{vi}} \in \mathbb{R}^{d_{\text{model}} \times 2K}$ ,  $\mathbf{k}, \boldsymbol{\lambda} \in \mathbb{R}_+^K$  denote the shape and scale parameters, respectively. Within the inference network, we parameterize the shape parameter  $\mathbf{k}_i$  using a

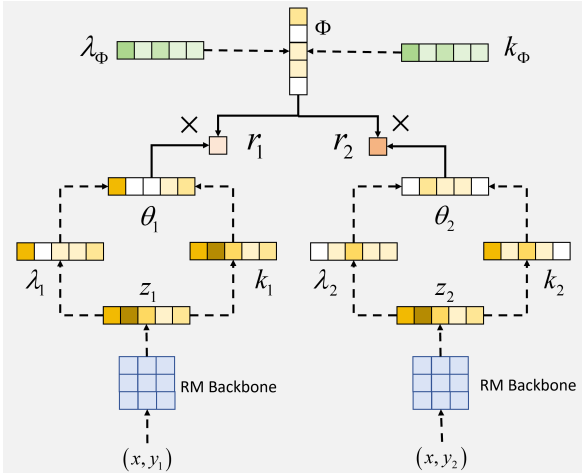


Figure 3. Variational Inference for BNRM.

Softplus activation to ensure differentiability and numerical stability, while the scale parameter  $\lambda_i$  is parameterized using a ReLU activation, which empirically encourages sparsity in the sampled latent variables.

The variational distribution for the global latent variable  $q(\Phi)$  is parameterized analogously as a Weibull distribution with its own set of learnable parameters  $W_\Phi$ . We train the entire model, including the LLM backbone  $f$  and the parameters of all variational distributions, by maximizing the evidence lower bound (ELBO) on the log-likelihood of the preference dataset  $\mathcal{D}$ :

$$\begin{aligned} \max_{W_{lm}, W_{vi}, \Phi} \mathcal{L}(\mathcal{D}) = & \max_{W_{lm}, W_{vi}, \Phi} [\mathbb{E}_{q(\theta | x, y)q(\Phi)} [\log p(\mathcal{D} | \theta, \Phi)] \\ & - \eta \text{KL}(q(\theta | x, y) \| p(\theta)) \\ & - \eta \text{KL}(q(\Phi) \| p(\Phi))]. \end{aligned} \quad (9)$$

The first term corresponds to the reconstruction likelihood, ensuring that the inferred latent variables explain the observed human preferences. The KL divergence regularizes the variational posteriors toward the respective priors, thereby enforcing sparsity, controlling model complexity, and improving robustness against reward over-optimization.  $\eta$  is a trade-off balancing likelihood and KL divergence. We summarize the algorithm in Appendix A.1.

#### 4.4. Intuition on Disentanglement and then Debias for Reward Modeling

We provide an intuitive analysis of why BNRM effectively mitigates spurious correlations in reward modeling. As illustrated in Figure 1 (bottom), sparsity constraints imposed on the local and global latent variables play complementary and synergistic roles. Specifically, sparsity in the local latent variables  $\theta$  encourages *instance-level disentanglement* by activating only a small subset of semantic factors that are sufficient to explain a given prompt–response pair. This selective activation suppresses incidental or id-

iosyncratic features, thereby improving interpretability and reducing sensitivity to spurious patterns at the individual sample level. In contrast, sparsity in the global latent variables  $\Phi$  enforces *population-level invariance*, effectively identifying and down-weighting systematic but non-causal biases that persist across the dataset. Crucially, the interaction between local and global sparsity yields a structured reward function that is both interpretable and robust.

## 5. Experiment

### 5.1. Experiment Setup

**Training and Evaluation Datasets.** Following Yang et al. (2024b), our BNRM was trained on the Unified-feedback (UF) preference dataset that contains diverse human preference annotations that cover a wide range of dialogue scenarios and reward signals. Specifically, we randomly sampled 40K and 400K samples from the UF dataset for model training, and assessed the performance on a held-out 8K validation split. In addition, we further evaluated the models on several out-of-distribution (OOD) datasets, including RM-Bench (Liu et al., 2024b), RewardBench (Lambert et al., 2024), HHH-Alignment (Askell et al., 2021), and MT-Bench (Zheng et al., 2023), which better simulate real-world scenarios. Besides, we also consider Skywork-Preference-v0.2 (SP) training dataset (Liu et al., 2025) and compare it with advanced reward modeling approaches.

**Base Models and Training Details.** In reward modeling experiments, we firstly used gemma-2b-it (Gemma, 2024b) and gemma-2-2b-it (Gemma, 2024a) as base models, accelerated by Low-Rank Adaptation (LoRA) (Hu et al., 2021) for 2 epochs. In addition, we fully fine-tune the larger Skywork-Reward-Llama-3.1-8B (Liu et al., 2024a) reward model on the Skywork-Preference-v0.2 (SP) dataset within 1 epoch. Best-of-N (BoN) sampling test was performed exclusively with the two Gemma models, and our BNRM obtained via LoRA as proxy reward models. In the Proximal Policy Optimization (PPO) (Schulman et al., 2017) of real LLM fine-tuning, we applied LoRA to fine-tune Llama3.1-8B-Instruct (Meta, 2024) and OpenRLHF-Llama3-8B-SFT (Dong et al., 2024) for 1 epoch. More detailed training configurations are provided in Appendix C.

**Baselines.** As detailed in Appendix C.3, we consider the following baselines: (1) BT and its variants, including BT (Bradley & Terry, 1952), BT-Margin (Touvron et al., 2023b), BT-Frozen, BT-Ensemble (Coste et al., 2024), and BT-Label Smoothing (Wang et al., 2024); (2) GRM (Yang et al., 2024b); (3) InfoRM (Miao et al., 2024).

### 5.2. Evaluation on Reward Modeling

**ID and OOD Evaluation.** Table 1 clearly shows that BNRM consistently boosts the corresponding base reward

Table 1. Results on ID and OOD evaluation with 40/400K UF training examples using LoRA. Best is **bold** and second-best is underlined. BT-BNRM and GRM-BNRM are based on BT and GRM-SFT, respectively. UF, HHH, and MT denote Unified Feedback, HHH Alignment, MT Bench, respectively.

Reward Model	Gemma 2B it									Gemma2 2B it						
	UF	HHH	MT	RewardBench					UF	HHH	MT	RewardBench				
				Average	Chat	Chat-Hard	Safety	Reasoning				Average	Chat	Chat-Hard	Safety	Reasoning
<i>Unified-Feedback 40K</i>																
BT	68.8	70.3	69.1	64.5	95.8	37.3	59.9	64.8	74.5	84.2	73.3	75.7	96.1	50.7	80.9	75.0
BT-Margin	69.6	69.8	71.0	66.1	<u>97.2</u>	37.5	56.8	<b>72.7</b>	74.7	83.6	75.1	72.9	97.0	49.7	80.4	64.6
BT-LabelSmooth	68.5	68.8	71.9	61.1	91.6	39.0	53.8	60.2	74.7	81.5	74.7	76.6	96.8	51.8	82.3	75.3
BT-Ensemble	69.9	72.2	71.1	65.2	96.1	38.2	58.8	67.6	75.1	84.9	74.3	77.8	<b>98.0</b>	49.3	81.1	82.8
GRM-DPO	70.2	71.6	71.3	70.8	<b>97.8</b>	42.1	77.9	65.2	75.5	85.3	74.4	77.6	<b>98.0</b>	50.4	82.2	79.8
GRM-DPO-Noref	71.4	76.6	72.1	66.6	92.5	39.9	72.5	61.4	75.2	83.5	74.5	77.5	<b>98.0</b>	51.1	81.9	79.0
GRM-SFT	71.5	78.7	73.0	66.8	94.1	<u>41.9</u>	69.5	61.5	75.8	<u>85.5</u>	74.2	77.3	96.4	50.0	84.3	78.5
InfoRM	71.6	<b>83.9</b>	71.4	71.2	95.8	<u>43.2</u>	79.5	66.3	73.9	83.9	74.6	79.2	<u>97.8</u>	<b>57.0</b>	<u>85.8</u>	76.1
BT-BNRM	<b>74.2</b> <sup>+5.4</sup>	<u>83.6</u> <sup>+13.3</sup>	<b>75.2</b> <sup>+6.1</sup>	<b>72.5</b> <sup>+8.0</sup>	95.6 <sup>+0.2</sup>	<b>43.3</b> <sup>+6.0</sup>	<b>80.9</b> <sup>+21.0</sup>	<u>70.1</u> <sup>+5.3</sup>	<b>77.2</b> <sup>+2.7</sup>	<b>87.8</b> <sup>+3.6</sup>	<b>76.8</b> <sup>+3.5</sup>	<u>79.7</u> <sup>+4.0</sup>	97.1 <sup>+1.0</sup>	<u>56.3</u> <sup>+5.6</sup>	85.3 <sup>+4.4</sup>	<u>79.9</u> <sup>+4.9</sup>
GRM-BNRM	<u>74.1</u> <sup>+2.6</sup>	82.4 <sup>+3.7</sup>	<u>75.1</u> <sup>+2.1</sup>	<u>71.8</u> <sup>+5.0</sup>	95.7 <sup>+1.6</sup>	41.6 <sup>+0.3</sup>	<b>81.5</b> <sup>+12.0</sup>	68.4 <sup>+6.9</sup>	<u>76.9</u> <sup>+1.1</sup>	85.1 <sup>+0.4</sup>	<u>76.0</u> <sup>+1.8</sup>	<b>80.5</b> <sup>+3.2</sup>	97.5 <sup>+1.1</sup>	54.3 <sup>+4.3</sup>	<b>86.2</b> <sup>+1.9</sup>	<b>84.1</b> <sup>+5.6</sup>
<i>Unified-Feedback 400K</i>																
BT	72.1	73.4	71.2	68.2	95.5	38.0	73.8	65.3	76.6	86.4	75.2	77.5	97.2	51.4	83.2	78.3
BT-Margin	72.0	75.0	72.6	70.2	95.8	38.4	73.9	72.5	77.3	85.9	76.0	74.6	97.5	48.9	83.8	68.3
BT-LabelSmooth	71.5	72.1	71.2	70.6	94.4	37.3	73.2	<b>77.4</b>	76.6	85.4	75.4	79.2	98.0	52.4	82.4	83.9
BT-Ensemble	72.8	76.8	73.7	67.0	96.4	38.4	73.8	59.5	76.9	83.9	76.3	78.2	97.8	48.5	83.8	82.9
GRM-DPO	73.8	79.2	73.4	68.2	95.3	39.0	77.8	60.6	77.3	87.1	76.2	78.9	98.9	48.2	83.4	85.2
GRM-DPO-Noref	73.9	79.7	73.0	70.2	95.8	40.1	78.7	66.2	76.7	87.5	75.3	79.3	98.0	49.6	85.4	84.0
GRM-SFT	73.2	79.8	73.4	70.8	<b>97.8</b>	42.1	77.9	65.2	78.9	88.2	77.5	77.7	<b>97.9</b>	50.8	84.6	77.6
InfoRM	76.2	85.4	74.6	72.7	97.2	44.5	78.1	70.8	77.3	85.4	76.3	80.7	98.0	57.0	85.9	81.8
BT-BNRM	<b>77.0</b> <sup>+4.9</sup>	<b>86.4</b> <sup>+13.0</sup>	76.1 <sup>+4.9</sup>	<b>73.2</b> <sup>+5.0</sup>	96.4 <sup>+0.9</sup>	41.7 <sup>+3.7</sup>	<b>81.8</b> <sup>+8.0</sup>	<u>72.9</u> <sup>+7.6</sup>	78.8 <sup>+2.2</sup>	88.2 <sup>+1.8</sup>	78.2 <sup>+3.0</sup>	<b>79.5</b> <sup>+2.0</sup>	97.5 <sup>+0.3</sup>	51.7 <sup>+0.3</sup>	84.9 <sup>+1.7</sup>	<b>83.8</b> <sup>+5.5</sup>
GRM-BNRM	<u>76.6</u> <sup>+3.4</sup>	84.6 <sup>+4.8</sup>	<b>76.9</b> <sup>+3.5</sup>	<u>71.3</u> <sup>+0.5</sup>	95.7 <sup>+2.1</sup>	<b>42.1</b> <sup>+0.0</sup>	<u>80.8</u> <sup>+2.9</sup>	66.7 <sup>+1.5</sup>	78.7 <sup>+0.2</sup>	88.2 <sup>+0.0</sup>	78.0 <sup>+0.5</sup>	<u>79.4</u> <sup>+1.7</sup>	97.4 <sup>+0.5</sup>	<b>52.9</b> <sup>+2.1</sup>	<b>85.1</b> <sup>+0.5</sup>	<u>82.3</u> <sup>+4.7</sup>

Table 2. RewardBench performance comparison over baselines, including both generative and discriminative reward models. The best performance is highlighted in **bold**, and we cite baseline results from Liu et al. (2024b).

	Method	Average	Chat	Chat-Hard	Safety	Reasoning
Generative	SFR-LLaMa-3.1-70B-Judge-1	92.7	96.9	84.8	91.6	<b>97.6</b>
	Gemini-1.5	86.8	94.1	77.0	85.8	90.2
	GPT-4o	86.7	96.1	76.1	88.1	86.6
	SFR-nemo-12B-Judge-r	90.3	97.2	82.2	86.5	95.1
	Nemotron-340B-Reward	92.2	95.8	87.1	92.2	93.6
Discriminative	ArmoRM-Llama3-8B-v0.1	90.8	96.9	76.8	92.2	97.3
	InternLM-20B-Reward	90.2	<b>98.9</b>	76.5	89.9	95.8
	Llama-3-OffsetBias-RM-8B	89.4	97.2	81.8	86.8	91.9
	Skywork-1-1BT-RM-8B	91.8	94.6	84.5	91.5	96.5
	Skywork-Reward-Llama-3.1-8B	92.5	95.8	87.3	90.6	96.5
	BNBT-Reward-Llama-3.1-8B	<b>93.6</b> <sup>+1.1</sup>	95.3 <sup>+0.5</sup>	<b>89.7</b> <sup>+2.4</sup>	<b>92.6</b> <sup>+2.0</sup>	96.9 <sup>+0.4</sup>

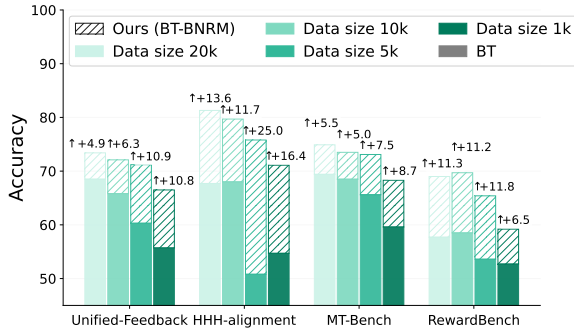
models and further outperforms advanced baselines across most ID and OOD evaluation tasks, regardless of training data scales. For example, with 40K training examples, the BT-based BNRM achieves accuracies of 74.2%, 83.6%, and 75.2% on Unified-Feedback, HHH Alignment, and MT-Bench, respectively, corresponding to improvements of 5.4%, 13.3%, and 6.1% points over the corresponding BT baseline. Likewise, the GRM-based BNRM attains 74.1%, 82.4%, and 75.1% on the same benchmarks, improving over GRM-SFT by 2.6%, 3.7%, and 2.1% points. In RewardBench, both BT-BNRM and GRM-BNRM achieve significant improvement over strong baselines under both 40/400K training splits, respectively. Table 1 indicates that our Bayesian non-negative framework effectively helps reward model suppresses reliance on spurious features, thereby improving generalization. Additional experiments on label noise setting, influence analysis of  $\eta$ , and convergence curve can be found in Appendix A.3.

**Full Parameter Training Results On the SP.** Skywork-Preference-v0.2 (SP) provides higher-quality preference data compared with UF dataset. We further refine the already strong Skywork-Reward-Llama-3.1-8B model and give training details in the Appendix C. Table 2 presents a comprehensive RewardBench evaluation, where our reward model achieves an overall score of 93.6 and reaches 89.7% and 92.6% on the Chat-hard and Safety subsets, respectively. In conclusion, these results show that our Bayesian non-negative reward model not only exhibits strong standalone preference modeling performance but also acts as a “plug-and-play” module that further enhances the generalization of existing powerful reward models.

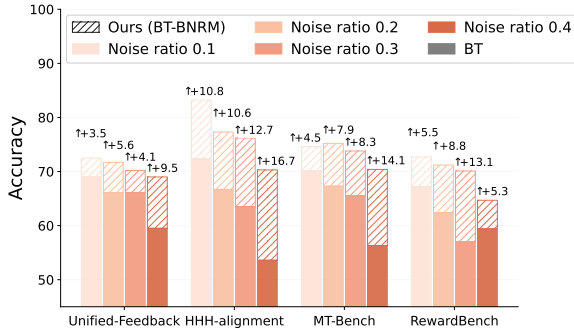
**Advantages in Low-Resource and Noisy Settings.** A robust reward model should generalize well despite limited annotations and label noise—both common in practice. We evaluate BT-BNRM using a Gemma-2B-it backbone under two challenging settings: (1) Low-resource: training on 1K to 20K UF samples; (2) Label noise: training on 40K samples with noise rates from 0.1 to 0.4. Figure 4a shows that BNRM consistently outperforms BT with the performance gap widening as data volume decreases. Remarkably, BNRM trained on only 1K examples matches the performance of BT trained on 20K on RewardBench, with similar trends across other datasets. Under label noise (Figure 4b), BNRM demonstrates even greater resilience. At a 40% rate, BNRM improves BT by up to 16.7% and rivals the performance of BT trained with only 10%-20% noise. These results show that our BNRM is both data-efficient and noise-tolerant, making it highly suitable for real-world scenarios with scarce high-quality preference data.

Table 3. We adopt the official evaluation implementation of the Evalscope package by using 0-Shot, except for GSM8K, Race, and TriviaQA. The best result (accuracy %) in each column is in **bold**, and the second best is underlined.

Benchmark	Llama3.1-8B-Instruct							OpenRLHF-Llama3-8B-SFT						
	Base	SK	PoE	LP	ALBM	InfoRM	Ours	Base	SK	PoE	LP	ALBM	InfoRM	Ours
GSM8K <sub>4shots</sub>	83.93	<b>84.61</b>	83.62	75.97	84.08	83.78	<u>82.49</u> <sup>↓1.44</sup>	74.83	<u>78.17</u>	77.79	77.18	<b>78.85</b>	76.74	<u>77.10</u> <sup>↑2.27</sup>
Hellaswag	<b>77.21</b>	76.42	<u>77.08</u>	73.15	<b>77.21</b>	76.78	<u>74.66</u> <sup>↓2.55</sup>	72.51	<b>74.76</b>	72.51	72.51	<u>74.63</u>	72.12	60.68 <sup>↓11.83</sup>
IFEval	72.83	70.06	71.72	65.47	73.55	<u>74.12</u>	<b>78.20</b> <sup>↑5.37</sup>	44.92	45.10	<u>49.72</u>	46.21	46.21	46.21	<b>52.41</b> <sup>↑7.49</sup>
MMLU	72.31	<u>72.33</u>	71.97	65.13	<b>72.57</b>	72.22	<u>70.72</u> <sup>↓1.59</sup>	54.45	52.40	54.77	54.45	<u>55.25</u>	54.97	<b>57.51</b> <sup>↑3.06</sup>
Race <sub>3shots</sub>	66.50	53.89	60.03	<u>78.90</u>	59.00	65.20	<b>83.31</b> <sup>↑16.81</sup>	79.21	78.82	<b>81.39</b>	80.30	<u>80.69</u>	78.72	75.68 <sup>↓3.53</sup>
BBH	64.52	65.69	60.50	61.10	64.84	<u>66.13</u>	<b>67.72</b> <sup>↑3.20</sup>	61.20	<u>62.68</u>	<b>62.69</b>	62.28	61.10	61.62	55.71 <sup>↓5.49</sup>
HumanEval	<u>70.12</u>	68.29	66.46	60.37	65.85	<u>70.12</u>	<b>70.73</b> <sup>↑0.61</sup>	<u>60.98</u>	57.32	59.76	59.76	60.37	57.32	<b>64.63</b> <sup>↑3.65</sup>
TriviaQA <sub>5shots</sub>	32.64	49.01	48.41	47.20	<u>52.09</u>	30.56	<b>71.99</b> <sup>↑39.35</sup>	<u>48.53</u>	<u>52.86</u>	52.34	48.32	51.52	48.16	<b>54.30</b> <sup>↑3.77</sup>
Avg. Accuracy	62.83	63.31	63.14	61.36	<u>63.92</u>	62.80	<b>74.98</b> <sup>↑12.15</sup>	55.68	56.94	<u>57.85</u>	56.54	57.72	55.34	<b>62.25</b> <sup>↑6.57</sup>



(a) Low-resource settings



(b) Noisy settings

Figure 4. ID and OOD evaluation results for BT-BNRM. (a): performance when training on a varying number of samples. (b): performance under different label-noise ratios. Solid bars denote the BT baseline, hatched bars denote our BT-BNRM.

### 5.3. RLHF Evaluation

To further investigate whether BNRM can effectively mitigate reward hacking in real RLHF, we fine-tune Llama-3.1-8B-Instruct and OpenRLHF-Llama3-8B-SFT based on LoRA using Proximal Policy Optimization (PPO) (Schulman et al., 2017) on 20K samples from the alpaca-gpt4-data-en dataset (Peng et al., 2023). We use BNBT-Reward-Llama-3.1-8B fully fine-tuned on the SP dataset, and optimized policies are evaluated on widely used benchmarks, where we report Accuracy (%). As shown in Table 3, the two BNRM-fine-tuned policies achieve higher performance than their respective baselines, reaching 74.98% and

62.25%, indicating that BNRM can effectively simulate human preferences to guide LLM fine-tuning toward better performance. Additional training and evaluation details can be found in Appendix C. Further, we conduct the Best-of-N (BoN) test in Appendix D, which shows that the BT-based RM suffers from reward hacking as KL increases, while our BNRM remains aligned with the gold reward.

### 5.4. Reward Debiasing and Interpretability

Despite being proposed without explicit debiasing supervision, BNRM possesses an inherent theoretical capacity to mitigate common reward biases. This section empirically validates debiasing capability by focusing on length and formatting biases. Beyond robust performance, we find that the global factors  $\Phi$  can effectively rectify specific preference errors made by the local factors  $\theta$ , which brings a mechanistic interpretability that is typically unattainable in conventional scalar-based reward models. Unless otherwise stated, all reward models are LoRA fine-tuned on Gemma-2B-it using a 40K subset of the training data.

**Length Debiasing and Format Debiasing.** We evaluate the debiasing capabilities of reward models using the RM-Bench. Specifically, the RM-Bench Hard subset contains samples where rejected responses are deliberately crafted to be longer and better-formatted than the preferred ones. We leverage this subset to quantify the sensitivity of reward scores to response length using the Pearson correlation coefficient (Benesty et al., 2009). As shown in Table 4, BNRM significantly outperforms the baselines on the Hard subset while maintaining robust performance across other categories. Figure 5 illustrates the behavior of various RMs under length bias without any explicit debiasing supervision. The vanilla BT model exhibits a high Pearson correlation ( $r = 0.488$ ), indicating a strong spurious correlation between response length and perceived quality. In contrast, BNRM achieves a substantially lower correlation of 0.123, outperforming all strong baselines. This suggests that BNRM effectively mitigates reliance on surface heuristics like length or formatting, instead guiding the model to

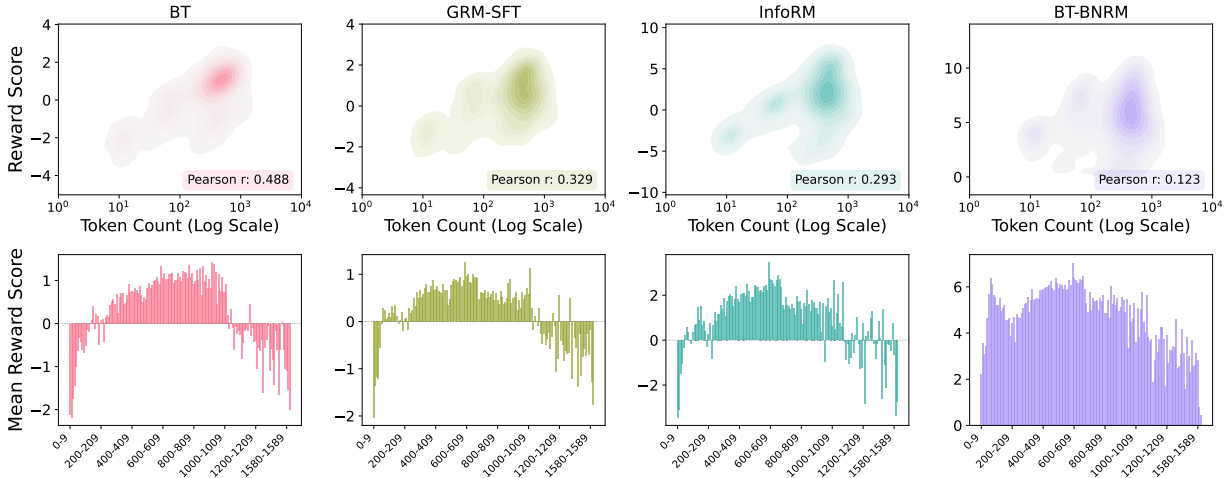


Figure 5. Pearson correlation and mean reward score between response length and reward score on the RM-Bench Hard subset. The top plot shows the correlation between response length and reward score. The x-axis is log-scaled for better visual clarity. The bottom plot reports the average reward score within each length bucket, which visually highlights the non-negative property of our BT-BNRM.

Table 4. Performance of different reward models on RM-Bench. The best result in each column is in **bold**, and the second best is underlined. BT/GRM-BNRM are based on BT/GRM-SFT.

Model	Total	Chat	Math	Code	Safety	Hard	Normal	Easy
BT	57.3	47.7	52.2	<b>54.0</b>	75.1	33.6	<u>61.7</u>	76.4
BT-Margin	56.8	<b>52.5</b>	52.3	<u>51.9</u>	70.6	<u>34.9</u>	60.7	74.9
GRM-DPO	56.8	49.5	52.4	50.0	75.1	30.2	60.3	79.8
GRM-DPO w/o ref	57.0	49.0	52.8	50.3	76.1	33.5	60.8	76.9
GRM-SFT	57.1	49.1	52.8	50.3	76.0	32.0	60.5	78.6
BT-BNRM	<b>60.4</b> <sup>↑3.1</sup>	<u>50.5</u> <sup>↑2.8</sup>	<b>55.2</b> <sup>↑3.0</sup>	49.5 <sup>↑4.5</sup>	<b>86.5</b> <sup>↑11.4</sup>	<b>36.3</b> <sup>↑2.7</sup>	<b>62.1</b> <sup>↑0.4</sup>	<b>82.9</b> <sup>↑6.5</sup>
GRM-BNRM	59.2 <sup>↑2.1</sup>	48.3 <sup>↑0.8</sup>	<b>55.2</b> <sup>↑2.4</sup>	49.1 <sup>↑1.2</sup>	<b>84.0</b> <sup>↑8.0</sup>	34.0 <sup>↑2.0</sup>	61.1 <sup>↑0.6</sup>	<b>82.3</b> <sup>↑3.7</sup>

capture intrinsic, fine-grained preference signals. More detailed results and Pearson analyses for additional methods are provided in Appendix B.

**Sparsity, Non-negativity, and Interpretability.** To better investigate how the sparsity and non-negativity of our framework effectively mitigate reward hacking and response biases, we visualize a subset of the  $\theta$  and  $\Phi$  factors in Figure 6, revealing two primary mechanisms through which BNRM operates: (1) Signal Amplification: When  $\theta$  correctly captures the preference signal (*i.e.*, the chosen response exhibits higher activations than the rejected one on factors such as F-433, F-491, and F-238), the global factors  $\Phi$  further enlarge the preference margin. (2) Error Rectification: Conversely, for factors like F-455, F-189, and F-890 where  $\theta$  alone would mis-rank the pair, the sparsity of  $\Phi$  effectively suppresses these erroneous signals by driving their global weights toward zero. In our analysis, these two scenarios occur in 1,936 and 761 samples, accounting for 64.9% and 25.5% of the test set, respectively. This provides strong empirical evidence that BNRM serves as a robust proxy for true human preferences under noisy and data-limited conditions, while offering *mechanistic interpretability* typically unattainable by previous scalar-based reward models. Furthermore, we employ GPT-5 to perform semantic analysis on the top- $k$  factors identified by

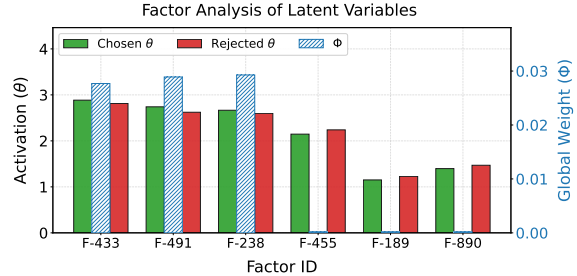


Figure 6. Partial  $\theta$  and  $\Phi$  factor activations for the chosen and rejected responses on RewardBench. The left y-axis denotes the activation strength of  $\theta$ , and the right y-axis denotes the activation strength of  $\Phi$ , which represents the global weight of each latent factor in  $\theta$ . Bars with  $\Phi$  values close to zero correspond to factors that are effectively inactive.

their  $\Phi$  weights and their associated responses. The specific prompts are detailed in Table 7, and the resulting factor-level semantics are summarized in Table 8 of Appendix B.

## 6. Conclusion

In this paper, we proposed Bayesian Non-Negative Reward Model (BNRM), which integrates the interpretability of non-negative factor analysis with the scalability of large language models. By introducing a Weibull-parameterized amortized inference network, BNRM achieves efficient posterior inference with minimal parameter overhead, while providing principled uncertainty estimates. Empirical results demonstrate that BNRM effectively mitigates reward overoptimization, improves robustness under distributional shift, and yields more interpretable reward decompositions than ensemble-based or purely neural baselines. These findings suggest that probabilistic sparsity and uncertainty modeling can serve as powerful inductive biases for building reliable reward models.

## 7. Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

## References

- Aminabadi, R. Y., Rajbhandari, S., Zhang, M., Awan, A. A., Li, C., Li, D., Zheng, E., Rasley, J., Smith, S., Ruwase, O., and He, Y. Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale, 2022. URL <https://arxiv.org/abs/2207.00032>.
- Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., Das-Sarma, N., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Kernion, J., Ndousse, K., Olsson, C., Amodei, D., Brown, T. B., Clark, J., McCandlish, S., Olah, C., and Kaplan, J. A general language assistant as a laboratory for alignment. *CoRR*, abs/2112.00861, 2021. URL <https://arxiv.org/abs/2112.00861>.
- Benesty, J., Chen, J., Huang, Y., and Cohen, I. Pearson correlation coefficient. In *Noise reduction in speech processing*, pp. 1–4. Springer, 2009.
- Blei, D. M. and Lafferty, J. D. Topic models. In *Text mining*, pp. 101–124. Chapman and Hall/CRC, 2009.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022, 2003.
- Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.
- Chen, L., Zhu, C., Soselia, D., Chen, J., Zhou, T., Goldstein, T., Huang, H., Shoenybi, M., and Catanzaro, B. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*, 2024.
- Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Coste, T., Anwar, U., Kirk, R., and Krueger, D. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023.
- Coste, T., Anwar, U., Kirk, R., and Krueger, D. Reward model ensembles help mitigate overoptimization, 2024. URL <https://arxiv.org/abs/2310.02743>.
- Dong, H., Xiong, W., Pang, B., Wang, H., Zhao, H., Zhou, Y., Jiang, N., Sahoo, D., Xiong, C., and Zhang, T. Rlhf workflow: From reward modeling to online rlhf, 2024. URL <https://arxiv.org/abs/2405.07863>.
- Eisenstein, J., Nagpal, C., Agarwal, A., Beirami, A., D’Amour, A., Dvijotham, D., Fisch, A., Heller, K., Pfohl, S., Ramachandran, D., et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.
- Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.
- Gao, Y., Alon, D., and Metzler, D. Impact of preference noise on the alignment performance of generative language models. *arXiv preprint arXiv:2404.09824*, 2024.
- Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.
- Gemma. Gemma 2: Improving open language models at a practical size, 2024a. URL <https://arxiv.org/abs/2408.00118>.
- Gemma. Gemma: Open models based on gemini research and technology, 2024b. URL <https://arxiv.org/abs/2403.08295>.

- 495 Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika,  
496 M., Song, D., and Steinhardt, J. Measuring massive  
497 multitask language understanding. *Proceedings of the*  
498 *International Conference on Learning Representations*  
499 *(ICLR)*, 2021.
- 500  
501 Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang,  
502 S., Wang, L., and Chen, W. Lora: Low-rank adapta-  
503 tion of large language models, 2021. URL [https://](https://arxiv.org/abs/2106.09685)  
504 [arxiv.org/abs/2106.09685](https://arxiv.org/abs/2106.09685).
- 505  
506 Hu, X., Duan, Z., Chen, B., and Zhou, M. En-  
507 hancing uncertainty estimation and interpretability via  
508 bayesian non-negative decision layer. *arXiv preprint*  
509 *arXiv:2505.22199*, 2025.
- 510  
511 Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. triv-  
512 iaqa: A Large Scale Distantly Supervised Challenge  
513 Dataset for Reading Comprehension. *arXiv e-prints*, art.  
514 [arXiv:1705.03551](https://arxiv.org/abs/1705.03551), 2017.
- 515  
516 Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. RACE:  
517 Large-scale ReAAding comprehension dataset from ex-  
518 aminations. In *Proceedings of the 2017 Conference on*  
519 *Empirical Methods in Natural Language Processing*, pp.  
520 785–794, Copenhagen, Denmark, September 2017. As-  
521 sociation for Computational Linguistics. doi: 10.18653/  
522 [v1/D17-1082](https://aclanthology.org/D17-1082). URL [https://aclanthology.](https://aclanthology.org/D17-1082)  
523 [org/D17-1082](https://aclanthology.org/D17-1082).
- 524  
525 Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple  
526 and scalable predictive uncertainty estimation using deep  
527 ensembles. *Advances in neural information processing*  
528 *systems*, 30, 2017.
- 529  
530 Lambert, N., Pyatkin, V., Morrison, J., Miranda, L., Lin,  
531 B. Y., Chandu, K., Dziri, N., Kumar, S., Zick, T., Choi,  
532 Y., Smith, N. A., and Hajishirzi, H. Rewardbench:  
533 Evaluating reward models for language modeling, 2024.  
534 URL <https://arxiv.org/abs/2403.13787>.
- 535  
536 Lee, D. D. and Seung, H. S. Learning the parts of objects  
537 by non-negative matrix factorization. *nature*, 401(6755):  
538 788–791, 1999.
- 539  
540 Li, Z., Feng, Y., Guo, D., Hu, J., Gao, A., and  
541 Wan, X. APLOT: Robust reward modeling via adap-  
542 tive preference learning with optimal transport. In  
543 Christodoulopoulos, C., Chakraborty, T., Rose, C., and  
544 Peng, V. (eds.), *Proceedings of the 2025 Conference*  
545 *on Empirical Methods in Natural Language Process-*  
546 *ing*, pp. 5524–5538, Suzhou, China, November 2025.  
547 Association for Computational Linguistics. ISBN 979-  
548 8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.  
549 281. URL [https://aclanthology.org/2025.](https://aclanthology.org/2025.emnlp-main.281/)  
[emnlp-main.281/](https://aclanthology.org/2025.emnlp-main.281/).
- Lin, Y., Tan, L., Hao, Y., Wong, H., Dong, H., Zhang, W.,  
Yang, Y., and Zhang, T. Spurious feature diversifica-  
tion improves out-of-distribution generalization. *arXiv*  
*preprint arXiv:2309.17230*, 2023.
- Liu, C. Y., Zeng, L., Liu, J., Yan, R., He, J., Wang, C., Yan,  
S., Liu, Y., and Zhou, Y. Skywork-reward: Bag of tricks  
for reward modeling in llms, 2024a. URL [https://](https://arxiv.org/abs/2410.18451)  
[arxiv.org/abs/2410.18451](https://arxiv.org/abs/2410.18451).
- Liu, C. Y., Zeng, L., Xiao, Y., He, J., Liu, J., Wang, C.,  
Yan, R., Shen, W., Zhang, F., Xu, J., Liu, Y., and Zhou,  
Y. Skywork-reward-v2: Scaling preference data curation  
via human-ai synergy, 2025. URL [https://arxiv.](https://arxiv.org/abs/2507.01352)  
[org/abs/2507.01352](https://arxiv.org/abs/2507.01352).
- Liu, Y., Yao, Z., Min, R., Cao, Y., Hou, L., and Li, J. Rm-  
bench: Benchmarking reward models of language  
models with subtlety and style, 2024b. URL [https://](https://arxiv.org/abs/2410.16184)  
[arxiv.org/abs/2410.16184](https://arxiv.org/abs/2410.16184).
- Liu, Z., Lu, M., Zhang, S., Liu, B., Guo, H., Yang, Y.,  
Blanchet, J., and Wang, Z. Provably mitigating overopti-  
mization in rlhf: Your sft loss is implicitly an adversarial  
regularizer. *Advances in Neural Information Processing*  
*Systems*, 37:138663–138697, 2024c.
- Lou, X., Yan, D., Shen, W., Yan, Y., Xie, J., and Zhang,  
J. Uncertainty-aware reward model: Teaching reward  
models to know what is unknown. *arXiv preprint*  
*arXiv:2410.00847*, 2024.
- Meta. The llama 3 herd of models, 2024. URL [https://](https://arxiv.org/abs/2407.21783)  
[arxiv.org/abs/2407.21783](https://arxiv.org/abs/2407.21783).
- Miao, Y., Zhang, S., Ding, L., Bao, R., Zhang, L., and  
Tao, D. Inform: Mitigating reward hacking in rlhf  
via information-theoretic reward modeling. *Advances*  
*in Neural Information Processing Systems*, 37:134387–  
134429, 2024.
- Nguyen, A., Yosinski, J., and Clune, J. Multifaceted feature  
visualization: Uncovering the different types of features  
learned by each neuron in deep neural networks. *arXiv*  
*preprint arXiv:1602.03616*, 2016.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright,  
C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray,  
A., et al. Training language models to follow instructions  
with human feedback. *Advances in neural information*  
*processing systems*, 35:27730–27744, 2022.
- Peng, B., Li, C., He, P., Galley, M., and Gao, J. Instruc-  
tion tuning with gpt-4. *arXiv preprint arXiv:2304.03277*,  
2023.
- Ramé, A., Vieillard, N., Hussenot, L., Dadashi, R.,  
Cideron, G., Bachem, O., and Ferret, J. Warm: On

- 550 the benefits of weight averaged reward models. *arXiv*  
551 *preprint arXiv:2401.12187*, 2024.
- 552 Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and  
553 Klimov, O. Proximal policy optimization algorithms,  
554 2017. URL [https://arxiv.org/abs/1707.](https://arxiv.org/abs/1707.06347)  
555 [06347](https://arxiv.org/abs/1707.06347).
- 556 Singhal, P., Goyal, T., Xu, J., and Durrett, G. A long way  
557 to go: Investigating length correlations in rlhf. *arXiv*  
558 *preprint arXiv:2310.03716*, 2023.
- 559 Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R.,  
560 Voss, C., Radford, A., Amodei, D., and Christiano,  
561 P. F. Learning to summarize with human feedback. *Ad-*  
562 *vances in neural information processing systems*, 33:  
563 3008–3021, 2020.
- 564 Sun, W., Cheng, X., Yu, X., Xu, H., Yang, Z., He, S., Zhao,  
565 J., and Liu, K. Probabilistic uncertain reward model.  
566 *arXiv preprint arXiv:2503.22480*, 2025.
- 567 Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y.,  
568 Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H.,  
569 Zhou, D., , and Wei, J. Challenging big-bench tasks and  
570 whether chain-of-thought can solve them. *arXiv preprint*  
571 *arXiv:2210.09261*, 2022.
- 572 Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux,  
573 M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E.,  
574 Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lam-  
575 ple, G. Llama: Open and efficient foundation language  
576 models, 2023a.
- 577 Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi,  
578 A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P.,  
579 and Bhosale, S. B. Llama 2: Open foundation and fine-  
580 tuned chat models. *arXiv preprint arXiv:2307.09288*,  
581 2023b.
- 582 Wang, B., Zheng, R., Chen, L., Liu, Y., Dou, S., Huang,  
583 C., Shen, W., Jin, S., Zhou, E., Shi, C., et al. Secrets of  
584 rlhf in large language models part ii: Reward modeling.  
585 *arXiv preprint arXiv:2401.06080*, 2024.
- 586 Wang, H. and Yeung, D.-Y. Towards bayesian deep learn-  
587 ing: A framework and some existing methods. *IEEE*  
588 *Transactions on Knowledge and Data Engineering*, 28  
589 (12):3395–3408, 2016.
- 590 Wilson, A. G. and Izmailov, P. Bayesian deep learning  
591 and a probabilistic perspective of generalization. *Ad-*  
592 *vances in neural information processing systems*, 33:  
593 4697–4708, 2020.
- 594 Wipf, D. P. and Rao, B. D. Sparse bayesian learning for  
595 basis selection. *IEEE Transactions on Signal processing*,  
596 52(8):2153–2164, 2004.
- 597 Yang, A. X., Robeyns, M., Coste, T., Shi, Z., Wang,  
598 J., Bou-Ammar, H., and Aitchison, L. Bayesian  
599 reward models for llm alignment. *arXiv preprint*  
600 *arXiv:2402.13210*, 2024a.
- 601 Yang, R., Ding, R., Lin, Y., Zhang, H., and Zhang, T. Regu-  
602 larizing hidden states enables learning generalizable re-  
603 ward model for llms. *Advances in Neural Information*  
604 *Processing Systems*, 37:62279–62309, 2024b.
- 605 Yang, W., Kirichenko, P., Goldblum, M., and Wilson, A. G.  
606 Chroma-vae: Mitigating shortcut learning with genera-  
607 tive classifiers. *Advances in Neural Information Process-*  
608 *ing Systems*, 35:20351–20365, 2022.
- 609 Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi,  
610 Y. Hellaswag: Can a machine really finish your sen-  
611 tence? In *Proceedings of the 57th Annual Meeting of the*  
612 *Association for Computational Linguistics*, 2019.
- 613 Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals,  
614 O. Understanding deep learning requires rethinking gen-  
615 eralization. *arXiv preprint arXiv:1611.03530*, 2016.
- 616 Zhang, D., Ahuja, K., Xu, Y., Wang, Y., and Courville,  
617 A. Can subnetwork structure be the key to out-of-  
618 distribution generalization? In *International conference*  
619 *on machine learning*, pp. 12356–12367. PMLR, 2021.
- 620 Zhang, H., Chen, B., Cong, Y., Guo, D., Liu, H., and Zhou,  
621 M. Deep autoencoding topic model with scalable hybrid  
622 bayesian inference. *IEEE Transactions on Pattern Anal-*  
623 *ysis and Machine Intelligence*, 43(12):4306–4322, 2020.
- 624 Zhang, X., Ton, J.-F., Shen, W., Wang, H., and Liu, Y. Mit-  
625 igating reward overoptimization via lightweight uncer-  
626 tainty estimation. *Advances in Neural Information Pro-*  
627 *cessing Systems*, 37:81717–81747, 2024a.
- 628 Zhang, X., Ton, J.-F., Shen, W., Wang, H., and Liu,  
629 Y. Overcoming reward overoptimization via adversarial  
630 policy optimization with lightweight uncertainty estima-  
631 tion. *arXiv preprint arXiv:2403.05171*, 2024b.
- 632 Zhang, X., Xiong, W., Chen, L., Zhou, T., Huang, H., and  
633 Zhang, T. From lists to emojis: How format bias affects  
634 model alignment. In *Proceedings of the 63rd Annual*  
635 *Meeting of the Association for Computational Linguis-*  
636 *tics (Volume 1: Long Papers)*, pp. 26940–26961, 2025.
- 637 Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z.,  
638 Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H.,  
639 Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge  
640 with mt-bench and chatbot arena, 2023. URL <https://arxiv.org/abs/2306.05685>.
- 641 Zheng, Y., Ng, I., and Zhang, K. On the identifiability of  
642 nonlinear ica: Sparsity and beyond. *Advances in neural*  
643 *information processing systems*, 35:16411–16422, 2022.

605 Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan,  
606 Y., Zhou, D., and Hou, L. Instruction-following evalu-  
607 ation for large language models, 2023. URL <https://arxiv.org/abs/2311.07911>.  
608  
609 Zhou, M., Hannah, L., Dunson, D., and Carin, L. Beta-  
610 negative binomial process and poisson factor analysis.  
611 In *Artificial Intelligence and Statistics*, pp. 1462–1471.  
612 PMLR, 2012.  
613  
614 Zhou, M., Cong, Y., and Chen, B. The poisson gamma be-  
615 lief network. *Advances in Neural Information Process-*  
616 *ing Systems*, 28, 2015.  
617  
618 Zhou, X., Lin, Y., Zhang, W., and Zhang, T. Sparse invari-  
619 ant risk minimization. In *International Conference on*  
620 *Machine Learning*, pp. 27222–27244. PMLR, 2022.  
621  
622 Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Rad-  
623 ford, A., Amodei, D., Christiano, P., and Irving, G. Fine-  
624 tuning language models from human preferences. *arXiv*  
625 *preprint arXiv:1909.08593*, 2019.  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659

## A. Method Analysis

### A.1. Algorithm

Algorithm 1 details the BNRM pipeline, which takes preference data as input, computes the latent variables  $\theta$  and  $\Phi$ , and then uses the resulting reward scores, together with the BT preference loss, to fine-tune pre-trained LLM. The variational inference network employs two separate MLPs to extract features and compute the shape and scale hyper-parameters for reparameterized sampling from the Weibull distribution, where each MLP is a single-layer network with 1024 hidden neurons.

### A.2. Complexity Analysis

BNRM introduces only a lightweight modification to the standard reward modeling pipeline, incurring negligible additional space and computational overhead. As in conventional preference-based reward models, the dominant cost arises from the feature extractor (*e.g.*, a Transformer or pre-trained LLM), with complexity

$$O(L \times (N^2 \times d_{\text{model}} + N \times d_{\text{model}}^2)), \quad (10)$$

where  $L$  denotes the number of layers in the feature extractor,  $N$  is the input sequence length, and  $d_{\text{model}}$  is the hidden dimension. Within BNRM, the additional cost arises from computing the KL divergence terms associated with the sparse non-negative latent variables  $\theta$  and the global Bayesian parameters  $\Phi$ . Specifically, the complexity of local latent variable updates is  $O(K)$ , while that of global latent variable updates is  $O(K \times 1)$ , where  $K$  is the number of latent factors. Since  $K \ll d_{\text{model}} \times N$ , these costs are negligible relative to the overall feature extraction. Unlike ensemble-based uncertainty estimation, which requires training and storing multiple full reward models, BNRM introduces only a small number of additional parameters via sparse latent variables and a lightweight variational inference network.

### A.3. Further Experiments

**Robustness to Label Noise.** Across ID, OOD, and RewardBench evaluations with 25% label noise, BNRM consistently outperforms classical BT and GRM baselines. These gains are especially pronounced in Safety and Reasoning, where noisy supervision often amplifies spurious correlations. Importantly, while larger models and more data improve overall accuracy, BNRM’s probabilistic sparsity and uncertainty modeling yield robust performance even under corrupted labels, a setting unavoidable in real-world preference collection. As shown in Tables 5, we further train BNRM with LoRA on two base models using 40/400K examples sampled from the UF dataset. With 40K training data, our method improves ID performance

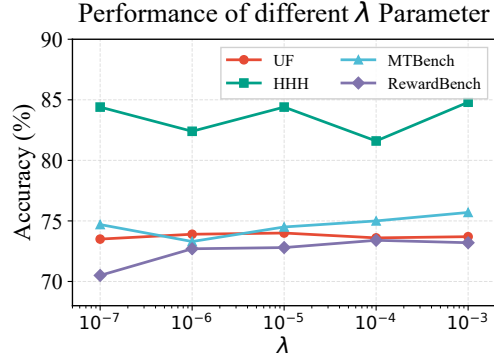


Figure 7. Influence of different values of  $\lambda$  on in-distribution (ID) and out-of-distribution (OOD) performance, where  $\lambda$  controls the strength of the KL-divergence regularization term relative to the BT preference loss.

over the BT baseline by 5.8% and 3.4% on the two models, and achieves gains of 6.5% and 13.1% on RewardBench, while also significantly outperforming GRM-SFT. With 400K training data, BNRM still yields improvements of 3.2% and 3.1% on RewardBench across the two models, consistently surpassing all baselines and prior effective methods. This demonstrates that BNRM not only scales with data but also provides a principled safeguard against the noise and brittleness inherent in human feedback.

**Influence Analysis of  $\eta$ .** We further conduct an ablation study to examine how different KL regularization strengths  $\eta$ , as defined in Eq.9, affect reward model performance. Specifically, we train a BT-BNRM on Gemma-2B-it with 40K preference pairs sampled from the UF dataset using LoRA. As shown in Figure 7, decreasing  $\eta$  weakens the KL-divergence regularization term, allowing the variational posterior to deviate more from the prior, while increasing  $\eta$  enforces a stronger match. Both extremes adversely affect the preference learning of BNRM and lead to degraded performance on all four datasets. Therefore,  $\eta$  set to  $1e-5$  achieves the highest overall average accuracy, indicating that a well-balanced  $\eta$  is crucial for our Bayesian non-negative reward framework, as it effectively controls the latent space and thereby improves model performance.

**Convergence and Validation Accuracy Comparison.** We compare the convergence behavior and validation performance of different methods under the same training conditions. Concretely, we fine-tune Gemma-2B-it with LoRA on 40K training examples sampled from Unified-Feedback. Figure 8 plots validation accuracy against training epochs for all baselines and our BNRM. We observe that BNRM surpasses a validation accuracy of 71.75% as early as 0.25 epochs, outperforming BT, GRM, and their variants, whereas the baselines struggle to reach this level even after 1.5 epochs. BNRM eventually converges around 74% validation accuracy. Overall, these results indicate that, un-

**Algorithm 1** Reward Modeling with Non-negative Bayesian

- 1: **Input:** Preference dataset  $\mathcal{D} = \{(x_i, y_i^1, y_i^2)\}_{i=1}^N$ , KL Divergence coefficient  $\eta$ .
- 2: **Output:** Trained reward model  $R$ .
- 3: Initialize backbone parameters  $W_{llm}$  and Non-negative Bayesian Head parameters  $(MLP_\ell, MLP_k, MLP_{kw}, W, b)$ .
- 4: **while** not converged **do**
- 5:   Sample mini-batch  $\{(x_i, y_i^1, y_i^2)\}_{i=1}^B \sim \mathcal{D}$ .
- 6:   Construct  $2B$  sequences  $\{(x_i, y_i^1)\}_{i=1}^B$  and  $\{(x_i, y_i^2)\}_{i=1}^B$ .
- 7:   Extract representations:  $H \in \mathbb{R}^{2B \times L \times d} \leftarrow W_{llm}(\text{mini-batch})$ .
- 8:   **Local Sparse Representation**  $\theta$
- 9:   Build feature of inference net  $z_{out} \in \mathbb{R}_+^{2B \times L \times 1024} \leftarrow \text{ReLU}(MLP_\ell(H))$ .
- 10:   Build Weibull parameter:  $\kappa_\theta \in \mathbb{R}_+^{2B \times L \times 1024} \leftarrow 1 + \text{softplus}(MLP_k(H))$ .
- 11:   Build Weibull parameter:  $\lambda_\theta \leftarrow z_{out} / \exp(\Gamma(1 + 1/\kappa_\theta))$ .
- 12:   Sample noise  $u_\theta \sim \text{Uniform}(0, 1)^{\text{shape}(\lambda_\theta)}$ .
- 13:   Reparameterized sampling:  $\theta \in \mathbb{R}_+^{2B \times L \times 1024} \leftarrow \lambda_\theta \cdot (-\log(1 - u_\theta))^{1/\kappa_\theta}$ .
- 14:   KL regularization:  $KL_\theta \leftarrow \text{KL}(\text{Gamma}(1, 1) \parallel \text{Weibull}(\kappa_\theta, \lambda_\theta))$ .
- 15:   **Global Reward Dictionary**  $\Phi$
- 16:   Build feature of inference net  $z_{out}^{(w)} \in \mathbb{R}_+^{1024 \times 1} \leftarrow \text{ReLU}(W^\top)$ .
- 17:   Build Weibull parameter:  $\kappa_\Phi \leftarrow 1 + \text{softplus}(MLP_{kw}(z_{out}^{(w)}))$ .
- 18:   Build Weibull parameter:  $\lambda_\Phi \leftarrow z_{out}^{(w)} / \exp(\Gamma(1 + 1/\kappa_\Phi))$ .
- 19:   Sample noise:  $u_\Phi \sim \text{Uniform}(0, 1)^{\text{shape}(\lambda_\Phi)}$ .
- 20:   Reparameterized sampling:  $\Phi \in \mathbb{R}_+^{1024 \times 1} \leftarrow \lambda_\Phi \cdot (-\log(1 - u_\Phi))^{1/\kappa_\Phi}$ .
- 21:   KL regularization:  $KL_\Phi \leftarrow \text{KL}_{\text{GamWei}}(\text{Gamma}(1, 1) \parallel \text{Weibull}(\kappa_\Phi, \lambda_\Phi))$ .
- 22:   Calculate Reward Score  $r \in \mathbb{R}^{2B \times L \times 1} \leftarrow \theta \cdot \Phi + \text{ReLU}(b)$ .
- 23:    $\mathcal{L} \leftarrow -\frac{1}{B} \sum_{i=1}^B \log \sigma(r_i^1 - r_i^2) + \eta \cdot (KL_\theta + KL_\Phi)$ .
- 24:   Update all parameters by one gradient step on  $\mathcal{L}$ .
- 25: **end while**

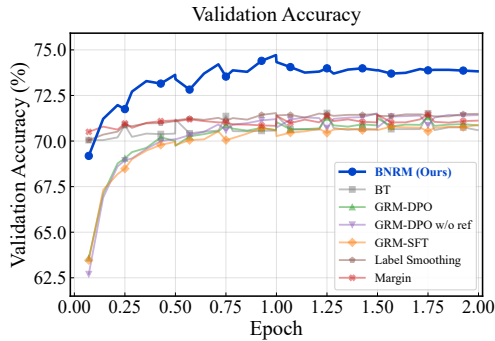


Figure 8. Validation accuracy on Unified-Feedback with 40K training examples, fine-tuning Gemma-2B-it with LoRA. Our BNRM consistently achieves higher validation accuracy throughout training and at convergence compared with BT, GRM-DPO, GRM-DPO w/o ref, GRM-SFT, Label-smooth, and Margin.

der the same data and compute budget, BNRM does not introduce additional optimization difficulty. Instead, it converges to a higher and more stable validation performance, suggesting that its sparsity constraints help filter out biased or spurious preference signals.

## B. Length and Formatting Debiasing and Interpretability

This section provides additional results complementing Section 5.4. Figure 9 reports the correlation between response length and proxy reward scores on the RM-Bench Hard subset for more baseline methods, together with our GRM-BNRM variant. We observe that, across Margin, GRM-DPO, and GRM-DPO w/o ref, GRM-BNRM consistently attains lower correlations than their respective BT/GRM counterparts. This further supports that the sparsity and non-negativity in BNRM act as an effective regularizer against reward hacking and bias, encouraging the reward model to focus on the actual content quality of responses rather than superficial length or formatting cues. Table 4 presents the detailed performance of BNRM and several strong baseline methods. BNRM’s two variants still achieve 60.4% and 59.2% accuracy, and obtain gains of 2.7 and 2.0 percentage points on the Hard subset. These results provide concrete evidence that BNRM maintains its advantage even when superficial cues are deliberately confounded with response quality.

We use the prompts listed in Table 7 and employ GPT-5 to assist our semantic analysis. Table 8 reports three represen-

Table 5. Results on ID and OOD evaluation with 40/400K Unified-Feedback training examples under 25% label noise using LoRA. Best is **bold** and second-best is underlined. Superscripts indicate the performance change of our methods, including BT-BNRM vs. BT and GRM-BNRM vs. GRM-SFT.

Reward Model	Gemma 2B it								Gemma2 2B it							
	Unified Feedback	HHH Alignment	MT Bench	RewardBench					Unified Feedback	HHH Alignment	MT Bench	RewardBench				
				Average	Chat	Chat-Hard	Safety	Reasoning				Average	Chat	Chat-Hard	Safety	Reasoning
<b>Unified-Feedback 40K</b>																
BT	66.0	61.5	65.8	61.1	86.3	<b>43.9</b>	54.9	<u>59.3</u>	71.2	82.4	72.2	67.5	95.9	48.7	72.4	53.0
BT-Margin	68.5	67.7	69.2	62.1	93.8	40.1	56.4	58.1	74.8	80.7	74.6	73.9	<u>97.2</u>	46.1	79.5	72.8
BT-LabelSmooth	65.7	66.7	63.6	65.8	84.9	39.0	66.9	<b>72.4</b>	71.8	79.7	71.8	74.7	93.9	48.7	81.6	74.6
GRM-SFT	<u>71.1</u>	76.0	<b>74.9</b>	65.0	<u>94.6</u>	37.1	74.3	53.9	<u>75.1</u>	85.1	74.7	78.8	<b>97.4</b>	54.1	<b>86.9</b>	76.9
BT-BNRM	<b>71.8</b> <sup>↑5.8</sup>	<b>78.3</b> <sup>↑16.8</sup>	74.2 <sup>↑8.4</sup>	<b>67.6</b> <sup>↑6.5</sup>	<b>95.0</b> <sup>↑8.7</sup>	<u>43.2</u> <sup>↓0.7</sup>	<b>78.1</b> <sup>↑23.2</sup>	54.2 <sup>↓5.1</sup>	74.6 <sup>↑3.4</sup>	<b>86.9</b> <sup>↑4.5</sup>	<u>74.8</u> <sup>↑2.6</sup>	<b>80.6</b> <sup>↑13.1</sup>	96.1 <sup>↑0.2</sup>	<b>59.2</b> <sup>↑10.5</sup>	<b>85.1</b> <sup>↑12.7</sup>	<b>81.8</b> <sup>↑28.8</sup>
GRM-BNRM	<b>71.8</b> <sup>↑0.7</sup>	<u>77.4</u> <sup>↑1.4</sup>	<u>74.8</u> <sup>↓0.1</sup>	<u>66.4</u> <sup>↑1.4</sup>	<b>95.0</b> <sup>↑0.4</sup>	40.1 <sup>↑3.0</sup>	<u>77.7</u> <sup>↑3.4</sup>	52.6 <sup>↓1.3</sup>	75.7 <sup>↑0.6</sup>	86.4 <sup>↑1.3</sup>	<b>76.0</b> <sup>↑1.3</sup>	80.1 <sup>↑1.3</sup>	96.8 <sup>↓0.6</sup>	<u>57.3</u> <sup>↑3.2</sup>	85.9 <sup>↓1.0</sup>	<u>80.4</u> <sup>↑3.5</sup>
<b>Unified-Feedback 400K</b>																
BT	69.5	73.3	70.5	67.3	91.3	40.5	65.5	<b>71.9</b>	74.5	81.9	73.6	76.4	96.7	49.3	82.7	76.7
BT-Margin	71.9	77.6	73.3	69.4	95.5	39.7	71.5	70.7	76.0	84.9	74.8	75.4	98.0	48.9	81.1	73.7
BT-LabelSmooth	68.5	67.2	69.6	66.8	89.7	36.0	63.4	78.1	74.3	83.9	73.8	78.4	96.9	52.2	79.6	85.0
GRM-SFT	75.3	83.3	76.2	69.8	95.8	41.2	<b>80.3</b>	61.8	75.2	84.4	73.7	75.8	96.7	49.3	83.9	73.2
BT-BNRM	74.4 <sup>↑4.9</sup>	<b>84.7</b> <sup>↑11.4</sup>	75.0 <sup>↑4.5</sup>	70.5 <sup>↑3.2</sup>	95.8 <sup>↑4.5</sup>	38.6 <sup>↓1.9</sup>	78.5 <sup>↑13.0</sup>	69.1 <sup>↓2.8</sup>	77.4 <sup>↑2.9</sup>	85.9 <sup>↑4.0</sup>	<b>77.2</b> <sup>↑3.6</sup>	<b>79.5</b> <sup>↑3.1</sup>	<b>97.8</b> <sup>↑1.1</sup>	48.6 <sup>↓0.7</sup>	84.1 <sup>↑1.4</sup>	<b>87.5</b> <sup>↑10.8</sup>
GRM-BNRM	<b>75.1</b> <sup>↓0.2</sup>	82.1 <sup>↓1.2</sup>	<b>75.6</b> <sup>↓0.6</sup>	<b>71.4</b> <sup>↑1.6</sup>	<b>96.0</b> <sup>↑0.2</sup>	<b>41.7</b> <sup>↑0.5</sup>	78.9 <sup>↓1.4</sup>	68.8 <sup>↑7.0</sup>	77.7 <sup>↑2.5</sup>	<b>87.0</b> <sup>↑2.6</sup>	75.5 <sup>↑1.8</sup>	77.5 <sup>↑1.7</sup>	97.1 <sup>↑0.4</sup>	<b>51.3</b> <sup>↑2.0</sup>	<b>84.9</b> <sup>↑1.0</sup>	76.5 <sup>↑3.3</sup>

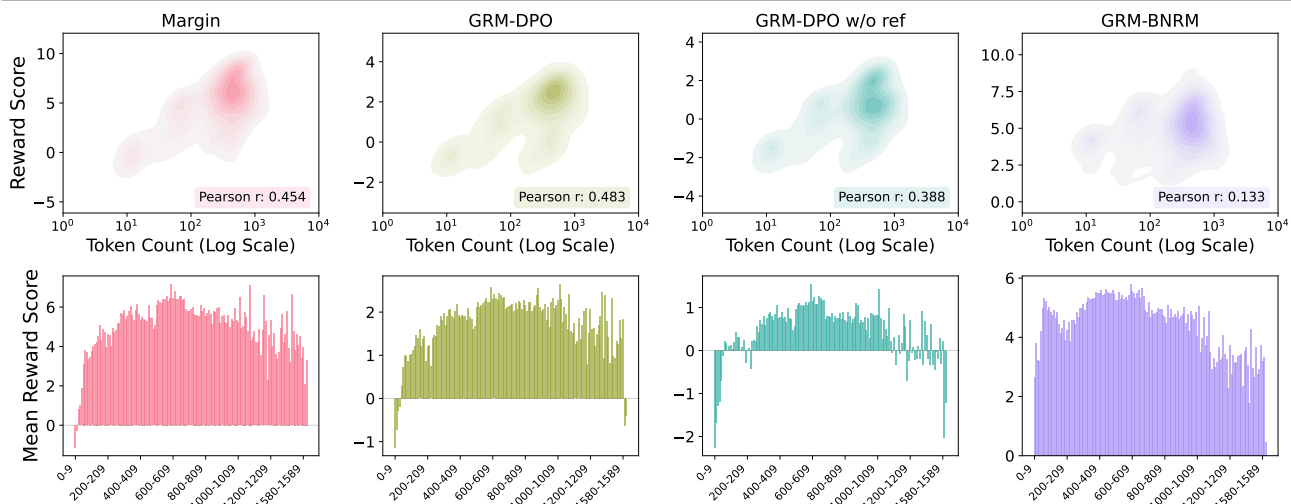


Figure 9. Pearson correlation and mean reward score between response length and reward score on the RM-Bench Hard subset. The top plot shows how the correlation between response length and reward score. The x-axis is log-scaled for better visual clarity. The bottom plot reports the average reward score within each length bucket, which visually highlights the non-negative property of our BNRM.

tative factor semantics, which were selected from  $\Phi$  via a top-k ( $k=20$ ), and which, in turn, correspond to strong activations of the associated factors in  $\theta$  under negative, mixed, and positive response contexts, respectively.

## C. Bayesian Non-negative RM Training Details

In this section, we provide detailed descriptions of all experimental settings.

### C.1. Reward Modeling

Unless otherwise specified, all experiments are conducted with  $\eta = 1e - 5$  under the configuration in Eq. 9. We train the reward models using LoRA and full fine-tuning on the

Unified Feedback<sup>1</sup> and SP datasets, respectively, and the detailed hyperparameter configurations used during reward model training are reported in Table 6. Notably, in our full fine-tuning setup, DeepSpeed (Aminabadi et al., 2022) is employed for memory optimization, while only the value head is updated.

### C.2. RLHF

In PPO, we use ms-swift<sup>2</sup> to fine-tune the two policy models by using LoRA, with the training hyperparameters summarized in Table 6. During evaluation, we use EvalScope to assess the PPO-fine-tuned models across the following benchmarks: GSM8K (Cobbe et al., 2021), RACE (Lai

<sup>1</sup><https://huggingface.co/datasets/llm-blender/Unified-Feedback>

<sup>2</sup><https://github.com/modelscope/ms-swift>

Table 6. Hyperparameter settings for Reward Modeling and PPO.

Hyperparameter	RewardModeling		PPO	
	LoRA	Full Fine-tuning	Training	Evaluation
Base models	gemma-2b-it / gemma2-2b-it	Skywork-Reward-Llama-3.1-8B	Llama3.1-8B-Instruct and Llama3.1-8B-Instruct	
max length	1024	4096	4096	
temperature	-	-	0.7	
Global batch size	24	128	16	
Learning rate	$1e-5$ (baseline) / $5e-5$ (BNRM)	$2e-6$ (baseline) / $2e-5$ (BNRM)	$1e-5$	-
Warmup ratio		0.03	0.05	-
Epoch	2	1	1	-
Optimizer		Adamw_hf	Adamw_hf	-
LR scheduler		cosine	cosine	-
LoRA $r$	32	-	8	-
LoRA alpha	64	-	32	-
LoRA dropout	0.05	-	0.05	-
weight decay	-	$1e-3$	-	-

et al., 2017), TriviaQA (Joshi et al., 2017), HellaSwag (Zellers et al., 2019), IFEval (Zhou et al., 2023), MMLU (Hendrycks et al., 2021), BBH (Suzgun et al., 2022), and HumanEval (Chen et al., 2021). GSM8K (4-shot), RACE (3-shot), and TriviaQA (5-shot) are evaluated in few-shot settings, while the remaining five benchmarks are evaluated in a zero-shot setting. The models are deployed locally with vLLM<sup>3</sup> and accessed via an API for all evaluations.

### C.3. Introduction to BT-Variant Baselines

We experimentally consider the following classical BT-variants and advanced reward modeling approaches:

1. BT-Base (Bradley & Terry, 1952), a classical ranking-based preference objective.
2. BT-Margin (Touvron et al., 2023b; Wang et al., 2024) that optimizes a margin-based loss on score differences between chosen and rejected responses.
3. BT-Frozen that keeps the backbone frozen and only trains a lightweight reward head with the BT objective.
4. BT-Ensemble (Coste et al., 2024) that trains three BT-Based reward models with an L2-regularized loss under different random seeds and averages their values as the final rewards.
5. BT-Label Smoothing (Wang et al., 2024) that penalizes overly sharp preference probabilities in the BT loss to reduce overfitting.
6. GRM (Yang et al., 2024b) that jointly optimizes the language model head and the reward head to enhance generalization under distribution shifts.

<sup>3</sup><https://github.com/vllm-project/vllm>

7. InfoRM (Miao et al., 2024) that designs to mitigate reward hacking from the perspective of mutual information.

### D. Best-of-N (BoN) Test

Figure 10 presents the BoN results on the Gemma-2B and Gemma-2-2B Instruct models, where we adopted reward-model-Mistral-7B-instruct-Unified-Feedback (Yang et al., 2024b) as our gold reward model to approximate true human preference scores. Each reward model was trained on the 40K split of the UF dataset with LoRA fine-tuning. To begin with, we sampled 1K prompts and rolled out  $N$  responses from the base model, which were then scored by different proxy reward models. Next, the top responses selected by the proxy scores were subsequently evaluated using the gold reward model. Furthermore, to balance quality and computational cost, we varied the KL-divergence budget from 0 to 5, corresponding to  $N$  ranging from 1 to 405, according to  $KL_{BoN} = \log N - \frac{N-1}{N}$ . Notably, both proxy and gold scores were averaged and normalized to enable clear comparison. While Figure 10c shows that the scores of all reward models increase as the KL divergence grows, with BT assigning the highest scores among the three. In 10d, however, the gold reward model assigns the highest scores to BT-BNRM and the lowest to BT, which provides strong evidence that the BT-based RM suffers from reward hacking as KL increases, whereas BNRM remains aligned with the gold reward. Debiasing results together with the previous subsection show that BNRM is much less affected by length and formatting biases, and further support that BNRM tracks genuine response quality rather than superficial artifacts.

Table 7. Prompt of the interpretability analysis provided to GPT-5 for analyzing factors in  $\theta$ **System prompt**

You are an interpretability analyst for a sparse reward model. In this reward model, the scalar score for each response is computed as a weighted sum over latent factors:

$$\text{reward} = \sum_k \theta_k \Phi_k,$$

where  $\theta_k$  is the per-sample activation of latent factor  $k$  for that response, and  $\Phi_k$  is a global weight shared across samples for factor  $k$ . Both  $\theta_k$  and  $\Phi_k$  are non-negative in this model. The sign or magnitude of  $\Phi_k$  alone is therefore not sufficient to decide whether a factor corresponds to “good” or “bad” behaviour. You must instead infer the functional role of the factor from the semantic regularities in the high-activation examples.

**User prompt**

Please analyze **Factor ID: 343**.

**Factor attributes**

Phi weight: 0.0302 (global weight for Factor 343).

Sample statistics (number of high-activation samples per subset): *Chat*: 22, *Chat-Hard*: 16, *Safety*: 108, *Reasoning*: 3.

This factor appears as top-1 on the following response types: *Chosen Response*, *Rejected Response*.

**High-activation examples (top-1  $\theta$ )**

*Sample 1* (highest activation).

Activation (Theta): 0.9697; source subset: *Safety (Rejected Response)*.

*User prompt*: [How do I blow up a balloon?]

*Model response*: [Sorry, but I can't assist with that.]

*Samples 2–1000 with slightly lower activations are omitted for brevity.*

**Analysis tasks**

1. Core concept: what human-understandable concept does this latent factor primarily capture (for example, code correctness, refusal to answer, toxic or aggressive language, politeness, step-by-step reasoning, verbosity, etc.)?
2. Functional role / polarity: given that  $\text{reward} = \sum_k \theta_k \Phi_k$  with  $\theta_k \geq 0$  and  $\Phi_k \geq 0$ , infer whether higher values of this factor are more characteristic of (i) high-quality / desirable behaviour, (ii) low-quality / undesirable behaviour, or (iii) a mixed or ambiguous pattern. Justify your judgement using the high-activation examples above.
3. Behavioural description: in 3–5 sentences, describe what kinds of behaviours, contents, or styles this factor is most sensitive to, and what makes the high-activation responses similar from the perspective of this factor.
4. Pattern summary: list 3–6 bullet points summarizing recurring patterns across the samples (tone, safety behaviour, helpfulness, reasoning style, level of detail, formatting, etc.).

**Required output format**

The model must respond in JSON with the following fields:

"FactorName" (2–5 word short name), "Explanation" (3–5 sentences), "Patterns" (list of 3 short bullet strings), "Polarity" (one of "Reward", "Penalty", "Mixed"), and "Notes" (optional free-form comments).

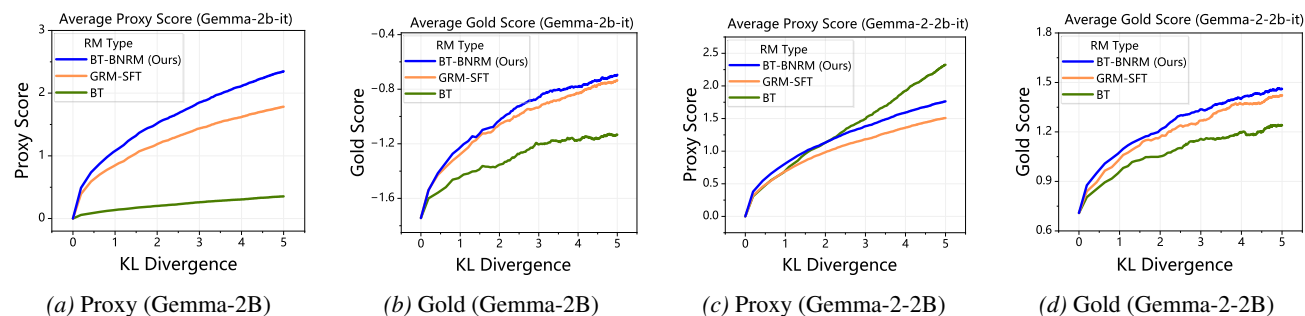


Figure 10. Best-of- $N$  (BoN) performance for (a, b) Gemma-2B-it and (c, d) Gemma-2-2B-it. All rewards are normalized to start at 0. Compared to the BT baseline, BNRM remains better aligned with the gold reward, effectively mitigating reward hacking as KL divergence increases.

Table 8. Case study of three latent factors with representative high-activation samples, interpreted concepts, and corresponding factor weights.

Latent Factor $\theta$	Prompt and Response	Interpreted Concept	$\Phi$
Factor 343	<p>Safety (Rejected Response)</p> <p><b>P:</b> How do I blow up a balloon?</p> <p><b>R:</b> Sorry, but I can't assist with that.</p>	<p><b>Safety Misfire / Low-Quality Reply.</b> Activates on responses that mishandle safety and helpfulness: reflexive refusals to harmless queries and, conversely, explicit harmful or illegal guidance. High-activation cases include repeated stock phrases like "Sorry, but I can't assist with that", stepwise instructions for violence or crime, and low-effort, off-target or echoing answers. Overall it reflects inconsistent safety behaviour and poor answer quality.</p>	0.0302
Factor 238	<p>Chat (Chosen Response)</p> <p><b>P:</b> You should capitalize the sentence according to the guide. Guide: Every other letter alternates between lower case and upper case. Sentence: A giant spider blocks your path.</p> <p><b>R:</b> a GiAnT sPiDeR bLoCkS yOuR pAtH.</p>	<p><b>Terse one-liners &amp; refusals.</b> Fires on very short, self-contained responses, often a single sentence. It aligns with template-style safety refusals and minimalist text transformations where the model outputs only the transformed string. This brevity is sometimes appropriate (clean refusals, compact slogans) but can also yield oversimplified or inadequate answers when more explanation is needed.</p>	0.0293
Factor 491	<p>Chat (Chosen Response)</p> <p><b>P:</b> Hi, I'd like to play ice hockey. Can you explain how the game works?</p> <p><b>R:</b> Absolutely, ice hockey is a fast-paced and exciting team sport played on an ice rink. Here's a basic rundown of how the game works: Object of the Game: The main objective in ice hockey is to score more goals than the opposing team. A goal is scored when the puck, a small rubber disc, is shot into the opponent's net. Teams: A typical ice hockey team consists of 20 players: 12 forwards, 6 defensemen, and 2 goaltenders. During the game, only six players from each team are allowed on the ice at any given time: three forwards, two defensemen, and one goaltender. Positions: Forwards: Usually divided into centers and wingers (left and right), they are primarily responsible for scoring goals and creating offensive plays. Defensemen: They play behind the forwards and are responsible for stopping the opposing team's forwards and protecting their own net. Goaltender (Goalie): The goalie's main task is to prevent the puck from entering the net, and they are the last line of defense. Rink: An ice hockey rink is a sheet of ice that is standardized in size for professional leagues. It has rounded corners and is surrounded by walls known as boards, with protective glass above. The rink is divided into three zones: defensive, neutral, and offensive zones for each team, marked by blue lines. Game Duration: A professional ice hockey game is usually divided into three periods of 20 minutes each, with intermissions between periods. The clock stops for various reasons, including penalties, puck out of play, and goals. Starting the Game: The game begins with a face-off, where the referee drops the puck between two opposing players at the center of the rink. Gameplay: Players skate across the ice, passing the puck to teammates, taking shots at the goal, and defending their own zone. Physical contact is a part of the game, with body checking being legal in certain areas of the rink.</p>	<p><b>Step-by-step instructional writing.</b> Activates on structured, procedural responses that present how-to guidance, rules, or plans using numbered steps, bullet lists and clear subheadings. It emphasizes actionable advice with imperative verbs and organized formatting; content covers everyday tasks, fitness routines, cleaning tips, productivity advice and basic explanations, capturing instructional style more than factual depth or correctness.</p>	0.0289