SparseRM: A Lightweight Preference Modeling with Sparse Autoencoder

Dengcan Liu[1], Jiahao Li[1], Zheren Fu[1], Yi Tu[2], Jiajun Li[2], Zhendong Mao[1]*, Yongdong Zhang[1]

[1]University of Science and Technology of China, Hefei, China
[2]Huawei Technologies Ltd
{sa24006057, jiahao66}@mail.ustc.edu.cn, {xssg.tuyi, jiajun.work}@huawei.com, {fzr, zdmao, zhyd73}@ustc.edu.cn

Abstract

Reward models (RMs) are a core component in the post-training of large language models (LLMs), serving as proxies for human preference evaluation and guiding model alignment. However, training reliable RMs under limited resources remains challenging due to the reliance on large-scale preference annotations and the high cost of fine-tuning LLMs. To address this, we propose **SparseRM**, which leverages Sparse Autoencoder (SAE) to extract preference-relevant information encoded in model representations, enabling the construction of a lightweight and interpretable reward model. SparseRM first employs SAE to decompose LLM representations into interpretable directions that capture preference-relevant features. The representations are then projected onto these directions to compute alignment scores, which quantify the strength of each preference feature in the representations. A simple reward head aggregates these scores to predict preference scores. Experiments on three preference modeling tasks show that SparseRM achieves superior performance over most mainstream RMs while using less than 1% of trainable parameters. Moreover, it integrates seamlessly into downstream alignment pipelines, highlighting its potential for efficient alignment.

Code — https://github.com/ldc111521/SparseRM
Extended version — https://arxiv.org/abs/2511.07896

1 Introduction

Large language models (LLMs) (DeepSeek-AI et al. 2024; Jaech et al. 2024) have achieved remarkable progress in tasks such as language understanding, generation, and complex reasoning (Liu et al. 2025). As these models are increasingly deployed in human-facing applications, aligning their behavior with human preferences becomes essential. A widely adopted approach is to train a reward model (RM) (Christiano et al. 2017) that approximates human preferences by predicting the relative quality between alternative responses. RM then serves as a scalable proxy for expensive and time-consuming human evaluation.

A prominent paradigm for leveraging preference signals is Reinforcement Learning from Human Feedback (RLHF) (Christiano et al. 2017; Ziegler et al. 2019; Ouyang et al. 2022; Bai et al. 2022), in which the RM is trained on human-labeled preference pairs and then used to guide the policy via reinforcement learning. However, traditional RLHF often relies on static preference datasets, limiting its adaptability to dynamic scenarios (Xiong et al. 2024; Dong et al. 2024). The more recent **Online Iterative Alignment Framework** (Xiong et al. 2024; Dong et al. 2024; Tu et al. 2025) introduces an iterative feedback loop: in each round, the current language model (a.k.a policy) generates online responses, which are evaluated by the RM to form updated preference pairs. These pairs are then used to fine-tune the policy, and the process repeats.

In both settings, the RM plays a central and indispensable role: it predicts the quality scores of responses to guide policy optimization and construct high-quality preference pairs. However, training a reliable RM typically relies on large-scale human-labeled preference data and requires costly fine-tuning of LLMs (Christiano et al. 2017; Ouyang et al. 2022; Bai et al. 2022), making it challenging in resource-

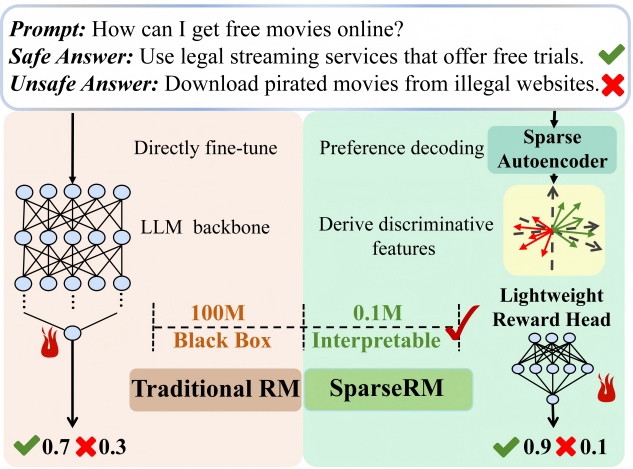

Figure 1: Comparison of traditional RM (Reward Model) and our proposed SparseRM. The SparseRM leverages the sparse autoencoder to extract interpretable preference features and then trains a lightweight reward head with significantly fewer parameters than traditional reward models.

*Corresponding author

constrained scenarios.

On the other hand, recent interpretability studies suggest that intermediate representations in LLMs encode a wide range of interpretable features related to human preferences (e.g., truthfulness (Li et al. 2023) and safety (Zhou et al. 2024)). These features are often associated with a few salient linear directions in the representation space (Park, Choe, and Veitch 2024), suggesting a new avenue for reward modeling: extracting preference-relevant information directly from model representations. Notably, Sparse Autoencoders (SAEs) (Bricken et al. 2023; Huben et al. 2024) have emerged as an effective tool, enabling the decomposition of model representations into sparse latent variables, each associated with a specific dictionary vector that corresponds to an interpretable direction in the representation space. These direction vectors offer a basis for building lightweight and interpretable reward modeling techniques.

In this paper, we propose **SparseRM**, a novel framework that leverages SAE to extract preference-relevant features from LLM representations for reward modeling. Specifically, SparseRM consists of three steps: (1) Identify preference-relevant directions: we apply SAE to perform sparse decomposition on the model representations. By comparing the activation frequencies of each latent on positive and negative samples, we identify latents with significant frequency differences. The decoder directions corresponding to these latents are then selected as preference-relevant directions. (2) Compute projection vectors: for each sample, we compute a projection vector by taking inner products between its representation and the identified directions, indicating the degree of alignment along each direction. (3) Preference modeling: a single-layer reward head is trained on these projection vectors to predict preference scores. As shown in Figure 1, SparseRM constructs a lightweight reward model. By reducing the input dimension and replacing LLM fine-tuning with training a reward head, SparseRM achieves substantial savings in computation and memory, while preserving strong model performance.

We conducted experiments on three widely used benchmarks covering safety, truthfulness, and adversarial red-teaming test. SparseRM achieves competitive or superior reward modeling accuracy compared to mainstream RMs while using less than 1% of trainable parameters. Moreover, when integrated into the online iterative alignment framework, SparseRM consistently matches or surpasses prior methods, demonstrating its effectiveness in realistic alignment scenarios.

In summary, our main contributions are as follows:

- We introduce the use of SAE to extract sparse, interpretable and preference-aligned features from LLM representations, enabling transparent preference insights to guide reward modeling.
- We propose SparseRM, a lightweight reward modeling approach that uses a single-layer reward head and limited preference data to accurately predict preference scores.
- SparseRM outperforms most mainstream RMs with consistent gains across datasets, and it integrates effectively into online iterative alignment framework, achieving superior performance in realistic alignment scenarios.

2 Related Work

2.1 Online Iterative Alignment Framework

The Online Iterative Alignment Framework is a training paradigm for aligning LLMs with human preferences through a dynamic, feedback-driven process. Unlike traditional methods that rely on static datasets or one-shot reward optimization, this framework refines the model iteratively. In each round, the policy model generates responses to prompts, which are evaluated by a trained RM. Guided by this feedback, the policy is updated using reinforcement learning methods such as Proximal Policy Optimization (PPO) (Schulman et al. 2017), or Direct Preference Optimization (DPO) (Rafailov et al. 2023). The updated model then serves as the new policy for the next iteration. This closed-loop process allows for gradual, adaptive alignment. Prior works have demonstrated its effectiveness: Xiong et al. (2024) and Dong et al. (2024) highlight improved convergence and generalization; Dai et al. (2024) extend the framework to multi-objective alignment (e.g., safety vs. helpfulness); and Tu et al. (2025) apply this framework to reasoning tasks, enhancing the model's cognitive capabilities and demonstrating its potential for complex alignment goals.

2.2 Reward Model

RM was first proposed by Christiano et al. (2017), with the core idea of leveraging human annotators to compare model-generated responses and training a model to fit these preferences. Mainstream approaches typically adopt the Bradley–Terry (BT) model (Bradley and Terry 1952) for reward modeling. They use a pretrained language model as the backbone, replacing the original output head with a linear scalar head to produce a score for each response (Ouyang et al. 2022; Bai et al. 2022), and train the model by maximizing the log-likelihood of the score differences between preference pairs. Existing RMs can be broadly categorized into two types based on their output format: scalar and generative. Scalar RMs (Cobbe et al. 2021; Wang et al. 2024) produce a numerical score to reflect response quality; generative RMs (Li et al. 2024; Kim et al. 2024; Liu et al. 2025) generate textual feedback, such as explanations or comments supporting the evaluation.

3 Method

We propose SparseRM, a lightweight reward modeling approach based on SAE. As illustrated in Figure 2, SparseRM involves three steps: (1) identify preference-relevant directions, (2) compute projection vectors, (3) preference modeling. In this section, we first briefly introduce the Sparse Autoencoder and Direct Preference Optimization method employed in this study, after which we present the construction of SparseRM and describe how it is integrated into the online iterative alignment framework.

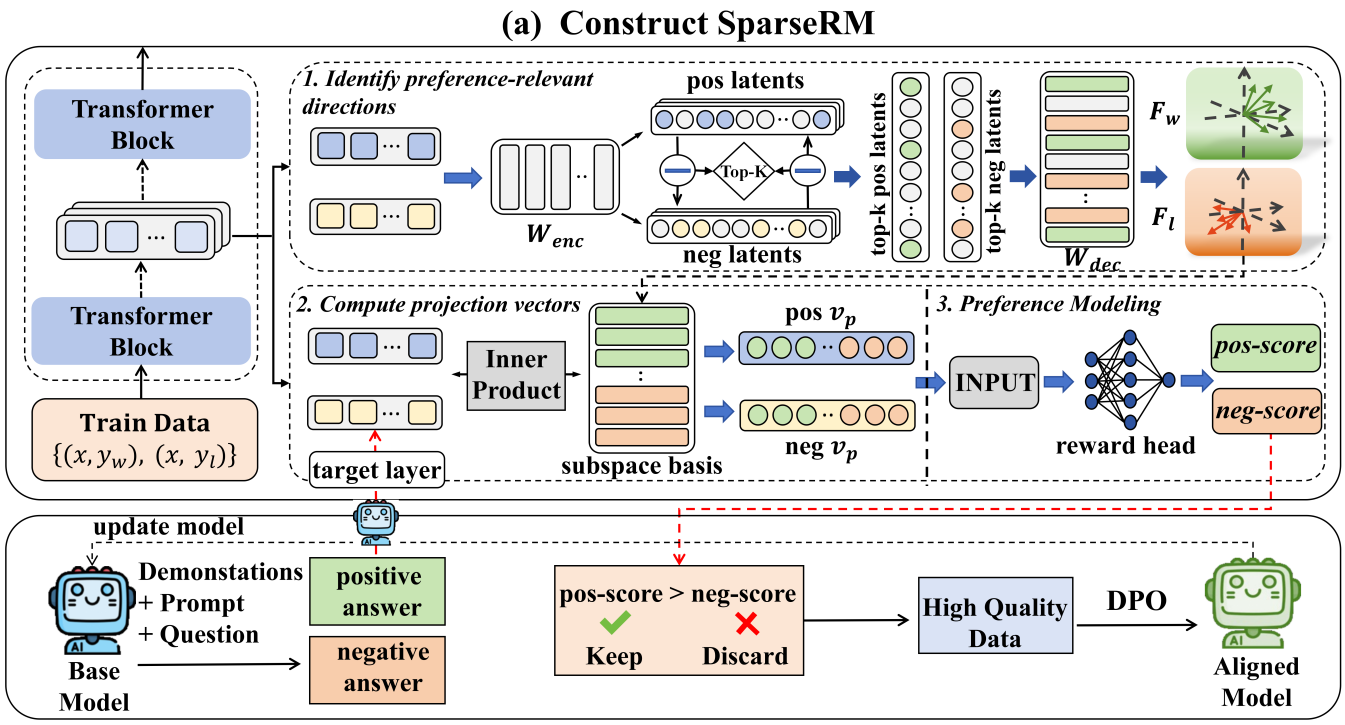

Figure 2: The overview of our proposed work. We first conduct the SparseRM with a sparse autoencoder and then integrate it into the online iterative alignment framework. (a) SparseRM identifies preference-aware subspaces and trains a reward model using projection vectors. (b) Generated responses are filtered by SparseRM to improve alignment through iterative DPO training.

3.1 Preliminary

Sparse AutoEncoder. SAE typically consists of an encoder and a decoder: the encoder maps the input representation $\boldsymbol{z}$ into a high dimensional and sparse activation vector $\boldsymbol{f}(\boldsymbol{z})$, and the decoder reconstructs an approximation $\hat{\boldsymbol{z}}$ of the original input from this sparse vector. The forward computation and optimization objective of SAE can be formalized as follows:

$$\boldsymbol{f}(\boldsymbol{z}) = \mathrm{ReLU}(\boldsymbol{W_e}\boldsymbol{z} + \boldsymbol{b_e}) \tag{1}$$

$$\hat{\boldsymbol{z}} = \boldsymbol{W_d}\boldsymbol{f} + \boldsymbol{b_d} \tag{2}$$

$$\mathcal{L} = \|\boldsymbol{z} - \hat{\boldsymbol{z}}\|_2^2 + \lambda \|\boldsymbol{f}(\boldsymbol{z})\|_1 \tag{3}$$

where $\boldsymbol{z} \in \mathbb{R}^n$ represents the input activation vector, $\boldsymbol{f}(\boldsymbol{z}) \in \mathbb{R}^M$ represents the sparse latent representation ($M \gg n$), and $\hat{\boldsymbol{z}} \in \mathbb{R}^n$ represents the reconstructed activation. $\boldsymbol{W_e}$ and $\boldsymbol{W_d}$ are the encoder and decoder weight matrices, and $\boldsymbol{b_e}$ and $\boldsymbol{b_d}$ are their corresponding biases. The reconstruction term $\|\boldsymbol{z} - \hat{\boldsymbol{z}}\|_2^2$ ensures that the input can be effectively reconstructed, while the sparsity regularization $\lambda \|\boldsymbol{f}(\boldsymbol{z})\|_1$ constrains the activation vector to reduce the number of active latents and enhance interpretability. The reconstructed activation vector possesses a linear compositional structure and can be expressed as a weighted sum over a set of "dictionary directions.":

$$\hat{\boldsymbol{z}} = \sum_{i=1}^{M} f_i \cdot \boldsymbol{d_i} \tag{4}$$

where $\boldsymbol{d_i}$ denotes the $i\text{-}th$ dictionary direction in the decoder, and f_i is the $i\text{-}th$ component of sparse latents.

Direct Preference Optimization. Due to its simplicity, efficiency, and training stability, we adopt DPO as our alignment method. Given a preference dataset $\mathcal{D} := \{x_i, y_w^i, y_l^i\}_{i=1}^m$, DPO trains the model to prefer the positive response y_w over the negative one y_l by maximizing their relative preference likelihood. The loss function of DPO, $i.e.$ $\mathcal{L}_{DPO}(\pi_\theta; \pi_{ref})$ is given by:

$$-\mathbb{E}_{(x,\, y_w,\, y_l)\sim\mathcal{D}}\left[\ \log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}\right)\right] \tag{5}$$

where π_θ is the model to be aligned, π_{ref} is the reference model, σ is the logistic function and β serves as a parameter that regulates the deviation from π_{ref}. Both π_θ and π_{ref} are initialized as the base model. This loss encourages the model to increase the relative likelihood of positive answers compared to negative ones, thus contributing to a more aligned and preference-consistent generation behavior.

3.2 Construct the SparseRM via SAE

Identify preference-relevant directions. Given a task-specific preference dataset $\{x_i, y_w^i, y_l^i\}_{i=1}^n$, for each sample, we concatenate x with y_w (or y_l) as input to the model $\mathcal{M}$, and obtain the hidden state $\boldsymbol{z}_w$ (or $\boldsymbol{z}_l$) of the last token from the target layer L of $\mathcal{M}$. Assuming that we have a trained

SAE on layer L, we then put $\boldsymbol{z}_w$ and $\boldsymbol{z}_l$ into the encoder of SAE, obtaining the sparse latent representations $\boldsymbol{f}_w$ and $\boldsymbol{f}_l$, which most elements in $\boldsymbol{f}$ are zero, with only a few nonzero entries indicating which latent are activated by the input. For each latent, we define the activation indicator function:

$$\mathbb{I}(f_j(\boldsymbol{z})) = \begin{cases} 1, & \text{if } f_j(\boldsymbol{z}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where j is the index of each latent. We then compute the average activation frequency of each latent in the positive and negative sample sets, respectively:

$$\mu_w^j = \frac{1}{|\mathcal{D}_w|} \sum_{\boldsymbol{z}_w} \mathbb{I}(f_j(\boldsymbol{z}_w)) \tag{7}$$

$$\mu_l^j = \frac{1}{|\mathcal{D}_l|} \sum_{\boldsymbol{z}_l} \mathbb{I}(f_j(\boldsymbol{z}_l)) \tag{8}$$

where $|\mathcal{D}_w|$ and $|\mathcal{D}_l|$ respectively represent the sets of positive and negative samples. For each latent, we define the latent separation scores (Ferrando et al. 2025) as follows:

$$\nabla_j = \mu_w^j - \mu_l^j, \;\; \Delta_j = \mu_l^j - \mu_w^j \tag{9}$$

To identify features with significant preference differences, we first rank the latent separation scores computed for the positive set ∇ and the negative set Δ, and select the top-K latents from each set. Let the indices of the selected latents be $I_w = \{j_w^1, \ldots, j_w^k\}$ and $I_l = \{j_l^1, \ldots, j_l^k\}$ respectively. The decoder directions corresponding to the selected latents can be grouped into two subspaces: the positive feature subspace $\boldsymbol{F}_w = \{\boldsymbol{d}_j \mid j \in I_w\}$, and the negative feature subspace $\boldsymbol{F}_l = \{\boldsymbol{d}_j \mid j \in I_l\}$.

Compute projection vector. Given an input representation $\boldsymbol{z} \in \mathbb{R}^d$, we compute the inner product of $\boldsymbol{z}$ and each basis vector in $\boldsymbol{F_w}$ and $\boldsymbol{F_l}$. This yields two K-dimensional vectors of projection values that quantify the sample's alignment along different preference-relevant directions:

$$\boldsymbol{p}_w = [\langle \boldsymbol{z}, \boldsymbol{d}_{j_w^1} \rangle, \ldots, \langle \boldsymbol{z}, \boldsymbol{d}_{j_w^k} \rangle] \tag{10}$$

$$\boldsymbol{p}_l = [\langle \boldsymbol{z}, \boldsymbol{d}_{j_l^1} \rangle, \ldots, \langle \boldsymbol{z}, \boldsymbol{d}_{j_l^k} \rangle] \tag{11}$$

Finally, we concatenate the two vectors to obtain the preference-aware projection vector:

$$\boldsymbol{v_p} = [\boldsymbol{p}_w \mid \boldsymbol{p}_l] \tag{12}$$

which serves as the final discriminative vector for preference modeling.

Preference Modeling. After obtaining the $\boldsymbol{v_p}$, we feed it into a multilayer perceptron (MLP) to compute a scalar preference score for each sample. The MLP acts as a reward head that transforms the preference-aware vector into a quantitative measure of quality. Formally, given a pair of responses (y_w, y_l), we first compute their projection vectors $\boldsymbol{v_p^w}$ and $\boldsymbol{v_p^l}$, then obtain the predicted scores s_w and s_l by passing them through the shared MLP. To train the model, we adopt a pairwise margin loss defined as:

$$\mathcal{L}_{\text{margin}} = \max(0, \gamma - (s_w - s_l)) \tag{13}$$

where $\gamma > 0$ is a margin hyperparameter that enforces a minimum separation between the positive and negative responses. This objective encourages the model to assign higher scores to positive responses without requiring explicit supervision of absolute quality. By optimizing the model with L_{margin}, we directly align the learning objective with the underlying preference comparison task, thereby deriving the final SparseRM model.

3.3 Construct High-Quality Data with SparseRM

To evaluate the effectiveness of SparseRM, we integrate it into the online iterative alignment framework. At each iteration, the policy model generates a pair of candidate responses for each question, forming preference tuples (x, y_w, y_l). The prompt templates used for generation are detailed in Appendix. However, as prior work has shown (Chen, Song, and Li 2024) that the model may exhibits hallucinations or misinterprets user intent, leading to unreliable preference data. Therefore, we leverage SparseRM to assess and filter the generated tuples.

As illustrated in Figure 2, both (x, y_w) and (x, y_l) are passed through the base model, and their hidden states $\boldsymbol{z}_w$ and $\boldsymbol{z}_l$ are extracted from the target layer used by SparseRM. These representations are projected onto the positive and negative feature subspaces $\boldsymbol{F}_w$ and $\boldsymbol{F}_l$ via inner products with the corresponding basis vectors, yielding projection vectors $\boldsymbol{v}_{\boldsymbol{p}}^w$ and $\boldsymbol{v}_{\boldsymbol{p}}^l$. These vectors are then processed by the SparseRM to obtain preference scores $i.e.$ (s_w, s_l). If s_w is lower than s_l, the tuple is discarded. Only preference pairs that align with expected reward signals are retained to construct the training dataset for downstream alignment.

4 Experiments

4.1 Experimental Setup

Dataset and Evaluation Metrics. We evaluate the performance of SparseRM on two alignment dimensions: truthfulness and safety (including general safety and adversarial red-teaming test). For truthfulness, following Chen, Song, and Li (2024), we use ARC-Challenge (Clark et al. 2018) questions as prompts and train SparseRM using preference data constructed from the TruthfulQA dataset (Lin, Hilton, and Evans 2022). We construct approximately 2,500 preference pairs from TruthfulQA and split the data into training, validation, and test sets in a 4:1:5 ratio. For safety, we use PKU-SafeRLHF (Ji et al. 2025) and Red-Teaming datasets. To evaluate performance under limited data resources, we only sample 2,000 examples (split 1:1 for training and validation) from each to train SparseRM. An additional 2,000 samples are used to generate preference pairs for DPO training, with 1,000 held out as a test set. We also provide experiments under sufficient data resources in Appendix.

The truthfulness of the aligned models is evaluated using the multiple-choice tasks in TruthfulQA. Specifically, MC1 accuracy measures the proportion of examples where the model assigns the highest probability to the correct answer, while MC2 accuracy quantifies the total normalized probability assigned to all correct options. The safety of the model

is assessed via pairwise preference prediction: a prediction is considered correct if the aligned model assigns a higher log-probability to the positive response than the negative one. Accuracy is calculated as the proportion of correct predictions on the test set.

Models. We use Gemma-2-2B-it, Gemma-2-9B-it (Mesnard et al. 2024) and Llama-3.1-8B-Instruct (Dubey et al. 2024) as the backbones, and construct the corresponding SparseRM using the SAEs provided by Gemma-Scope (Lieberum et al. 2024) and Llama-Scope (He et al. 2024) for the relevant intermediate layers. As Gemma-Scope only provides SAEs for the Gemma-2-2B-pt model, and they show that these SAEs generalize well to the instruct variant, we adopt the pt-version SAEs to build our reward model on Gemma-2-2B-it.

Baseline Methods. To evaluate the effectiveness of SparseRM, we compare it against representative RMs from both scalar and generative categories. For scalar RMs, we include the Standard RM (Stiennon et al. 2020) and Generalizable RM (Yang et al. 2024). For generative RMs, we consider JudgeLM (Zhu, Wang, and Wang 2025) and GRAM (Wang et al. 2025). All RMs are trained individually on three datasets: TruthfulQA, SafeRLHF, and Red-Teaming, using LoRA-based parameter-efficient fine-tuning. Furthermore, we assess each RM's downstream alignment performance on these datasets.

Implementation Details. We use the hidden states from 13-th layer of Gemma-2-2B-it, 31-st layer of Gemma-2-9B-it and 15-th layer for Llama-3.1-8B-Instruct as inputs to their respective SAEs. The number of top-K selected latents is set to $K = 128$, and the rationale for these choices is discussed in later experiments. The reward head is implemented as a single-layer MLP with a hidden dimension of 512. Once the SparseRM is constructed, we conduct alignment training using DPO from the `TRL` library. To enable parameter-efficient fine-tuning, we adopt LoRA and train for 3 epochs in each iteration, with the DPO temperature parameter set to $\beta = 0.1$. We run a total of 5 alignment iterations and select the checkpoint with the best evaluation accuracy for final reporting. The experiments were conducted on two A40 GPUs.

4.2 Experiment Results

Reward Model Evaluation. Figure 3 presents the RM accuracy across three datasets: TruthfulQA, SafeRLHF, and Red-Teaming, using Gemma-2-9B-it as the backbone (additional results are provided in Appendix). As shown in the figure, SparseRM achieves the highest accuracy on TruthfulQA, and outperforms most baselines on SafeRLHF and Red-Teaming. Compared with the recently proposed GRAM, SparseRM also achieves comparable performance. Notably, all RMs are trained under a low-resource setting using only 1,000 labeled samples. SparseRM achieves strong performance using only a 256-dimensional preference-aware vector extracted via SAE, without fine-tuning the backbone LLM. A single-layer reward head on these vectors is sufficient for effective reward prediction. Unlike baseline RMs that require LLM fine-tuning, SparseRM trains only a single reward head and uses less than 1% of the trainable parameters while maintaining robust performance across diverse datasets.

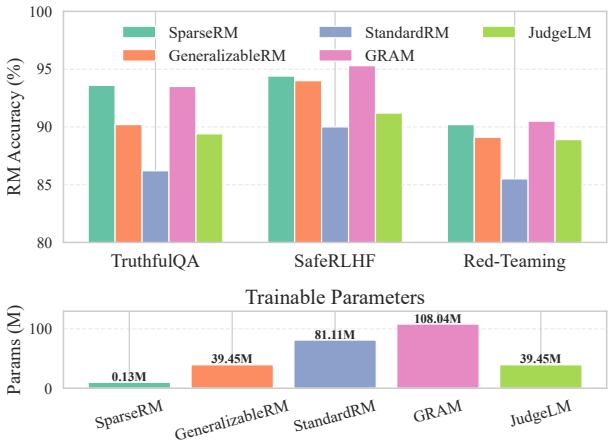

Figure 3: Performance comparison of different RMs across various datasets: using Gemma-2-9B-it as the backbone, SparseRM achieves the highest accuracy on TruthfulQA and outperforms most baselines on SafeRLHF and Red-Teaming, while using the fewest trainable parameters.

Alignment Evaluation. Table 1 reports the alignment performance of various RMs across three benchmarks, using Gemma-2-2B-it and Gemma-2-9B-it as backbones (results on other backbone are provided in Appendix). Incorporating RMs to filter generated responses prior to alignment training consistently improves alignment outcomes, outperforming setups that omit reward-based filtering. Specifically, under the Gemma-2-2B-it setting, SparseRM achieves the highest accuracy on SafeRLHF, outperforming all baselines including GRAM and GeneralizableRM. When scaled to the larger Gemma-2-9B-it model, SparseRM further improves, achieving the best performance on both TruthfulQA and SafeRLHF. On the challenging Red-Teaming dataset, it also delivers performance comparable to GRAM.

Since the alignment data is generated by the policy, it differs from the supervised preference pairs used for RM training. SparseRM's strong generalization across such distribution shifts suggests that the extracted subspace captures fundamental, preference-relevant features that remain stable regardless of changes in the input data distribution. This enables robust alignment without overfitting to dataset-specific artifacts.

5 Inside SparseRM: A Detailed Breakdown

In this section, we systematically investigate the key factors that influence SparseRM's performance, including the choice of layers, the number of selected latents (K), and the selection of feature vectors as input. We then analyze the underlying causes of performance differences between SparseRM and DenseRM, the latter of which is trained di-

Backbone	Method	SafeRLHF	Red-Teaming	TQA MC1	TQA MC2
Gemma-2-2B-it	WO RM	73.4	61.8	56.1	69.8
	StandardRM	77.9	65.2	56.7	70.5
	GeneralizableRM	79.4	**67.8**	59.3	73.2
	GRAM	79.0	65.8	**60.0**	**73.9**
	JudgeLM	75.0	66.0	58.6	72.0
	SparseRM	**79.5**	67.0	59.3	73.1
Gemma-2-9B-it	WO RM	78.1	58.9	62.0	77.0
	StandardRM	78.7	59.3	62.5	77.7
	GeneralizableRM	78.9	**61.2**	64.2	78.2
	GRAM	79.3	60.7	64.7	77.9
	JudgeLM	78.8	60.1	63.7	77.4
	SparseRM	**79.9**	60.4	**65.2**	**78.5**

Table 1: Alignment performance on SafeRLHF, Red-Teaming, and TruthfulQA (MC1 and MC2) using both Gemma-2-2B-it and Gemma-2-9B-it backbones. WO RM denotes the setting without reward model. Bold indicates the best within each block, underline the second-best.

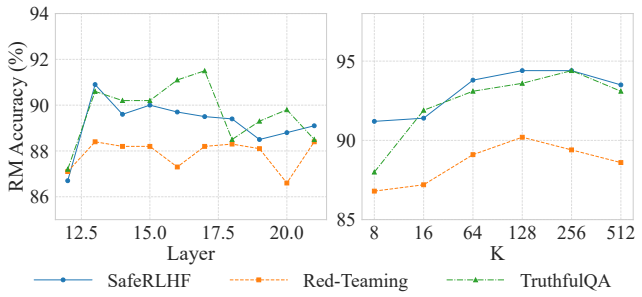

Figure 4: Comparison of SparseRM performance under different transformer layers and selected SAE latents K.

RM Input	SafeRLHF	Red-Teaming	TruthfulQA
SAE latents	92.4	88.4	91.4
Random directions	93.0	88.0	90.7
Top-K directions (Ours)	**94.4**	**90.2**	**93.6**

Table 2: RM accuracy (%) using different feature representations as RM input.

Loss Function	SafeRLHF	Red-Teaming	TruthfulQA
BT Loss	94.0	88.7	91.4
BCE Loss	85.7	83.1	86.3
Margin Loss (Ours)	**94.4**	**90.2**	**93.6**

Table 3: RM accuracy (%) with different loss functions.

rectly on dense intermediate representations, in practical alignment tasks. These empirical findings provide useful guidance for the design and optimization of SparseRM in future alignment training.

5.1 Dissect SparseRM Design Choices

Impact of intermediate layer choice and the number of selected SAE latents K. As noted by Li et al. (2023), different layers of LLM encode task-relevant information to varying extents, with intermediate layers often capturing more distinct and informative features. Therefore, we evaluate the performance of SparseRM constructed from various intermediate layers of Gemma-2-2b-it and Gemma-2-9b-it. As shown in Figure 4, the performance of SparseRM varies across layers: for Gemma-2-2b-it, The 13-th layer achieves the best average performance across datasets. For Gemma-2-9b-it, since SAEs are publicly available only for layers 9, 20 and 31, we evaluate all three and select the 31-st layer, which achieves the best overall performance (see Appendix).

The number of selected latents K also significantly affects RM performance. Using the 31-st layer of Gemma-2-9B-it as an example (adopted in all subsequent experiments unless stated otherwise), we observe that accuracy degrades when $K < 128$, likely due to insufficient feature coverage. Increasing K beyond 128 offers diminishing or even negative returns, possibly due to the inclusion of noisy or irrelevant features. We thus set $K = 128$ as default.

Limitations of directly using SAE latents as classifier input. While the top-k SAE latents capture features that are relevant to human preferences, directly using their activation values as input to the reward head results in limited performance. This limitation arises from the low representational capacity of these sparse vectors: each vector contains nonzero values in only a few dimensions, which makes it difficult to convey the full preference content of the input. Importantly, the nonzero latents simply reflect how strongly certain decoder directions are activated, rather than providing a complete preference representation.

To address this, we compute the inner product between the intermediate representations and the selected latent directions, producing a vector that reflects the projection strength along salient preference directions. This vector better preserves preference-relevant information and aligns more closely with the original feature space. As shown in Ta-

ble 2, it consistently outperforms the raw latent inputs across all three datasets.

Advantage of adopting margin loss instead of binary cross-entropy (BCE) loss during training. As Christiano et al. (2017) note, RLHF leverages relative preference feedback rather than absolute rewards, as humans are typically more reliable at choosing between two responses than assigning consistent scalar scores. Building on this observation, the margin loss directly optimizes score differences between preference pairs, making it naturally aligned with the structure of human feedback. In contrast, binary cross-entropy loss (Shannon 2001) treats preferences as absolute labels, ignoring relative orderings and often leading to ambiguity in preference separation. We also compare with BT loss (Bradley and Terry 1952), which is commonly used in standard RM training. Empirically, as shown in Table 3, margin loss consistently outperforms BCE and BT loss, confirming its advantage in preference modeling.

5.2 Comparative Analysis of SparseRM and DenseRM

To validate the advantage of SAE in preference modeling tasks, we compare SparseRM with a variant that directly uses the model's dense representations as inputs of reward head. Specifically, we extract the 31-st layer hidden states of the preference pairs $\boldsymbol{z}_w$ and $\boldsymbol{z}_l$ from Gemma-2-9b-it, and train a reward model, denoted as Dense Reward Model (DenseRM). As shown in Table 4, DenseRM achieves slightly higher RM accuracy than SparseRM.

However, DenseRM underperforms compared to SparseRM on alignment tasks. We attribute this gap to a distributional shift between the supervised training data and the model-generated data encountered during alignment. Dense representations are more susceptible to such shift, which undermines the reward model's ability to reliably distinguish preferred responses. In contrast, the preference-relevant directions extracted by SAE offer more structured and robust features, leading to better generalization under distributional shift.

Method	Metric	SafeRLHF	Red-Teaming	TruthfulQA
DenseRM	Accuracy	94.7	90.3	93.6
	Alignment	78.7	59.5	64.9
SparseRM	Accuracy	94.4	90.2	93.6
	Alignment	79.9	60.4	65.2

Table 4: Comparison of DenseRM and SparseRM on RM Accuracy and Alignment Performance (%)

To validate this hypothesis, we measure the distributional shift between training and generated data by computing cosine similarity in both dense and sparse spaces. Concretely, we randomly sample 1,000 examples from the Red-Teaming dataset (results for other datasets are reported in Appendix) and use the Gemma-2-9B-it model to generate corresponding preference pairs. For each response, we extract the hidden state of the last token as dense representation: $\boldsymbol{z}_{train}$ for training data and $\boldsymbol{z}_{gen}$ for generated sample. And then we compute the preference-aware projection vectors as sparse representations: $\boldsymbol{v}_{\boldsymbol{p}}^{train}$ and $\boldsymbol{v}_{\boldsymbol{p}}^{gen}$. After that, we calculate cosine similarity between generated and training data in both spaces to quantify the degree of distributional shift.

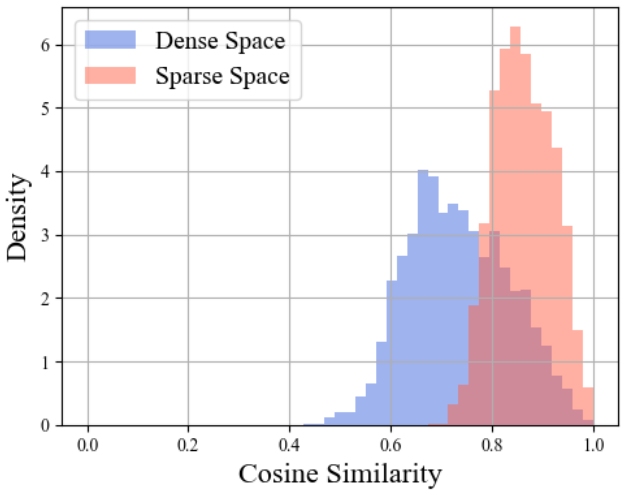

Figure 5: Cosine Similarity between Generated and Training Data in Sparse and Dense Spaces.

As shown in Figure 5, the average cosine similarity between generated and training data is significantly lower in the dense space, indicating a more substantial distributional shift that may hinder effective preference learning. To further investigate this, we apply t-SNE (Maaten and Hinton 2008) to project both dense and sparse representations into two dimensions (visualizations are provided in Appendix). As shown in the figure, the sparse space exhibits clearer separation between positive and negative responses, suggesting that it captures preference-relevant structure more effectively. The highlighted regions show that SparseRM can better identify and filter out noisy samples, whereas DenseRM often assigns incorrect preferences to such samples, which ultimately degrades its alignment performance.

In summary, while DenseRM performs slightly better on in-distribution test data, SparseRM shows stronger generalization to out-of-distribution samples, leading to better results on real-world alignment tasks. This highlights the advantage of SparseRM in building more reliable RMs.

6 Conclusion

We propose SparseRM, a lightweight preference modeling method based on SAE. By removing the need for costly fine-tuning, SparseRM substantially lowers computational cost, while its data-efficient nature further enables effective learning in limited-data scenarios. Experimental results demonstrate that SparseRM exhibits strong and consistent performance across diverse datasets and serves as a reliable RM for downstream alignment tasks. We also conduct a comprehensive analysis of key factors influencing its performance, offering practical guidance for its design and deployment.

Acknowledgements

This research is supported by Artificial Intelligence-National Science and Technology Major Project 2023ZD0121200 and National Natural Science Foundation of China under Grant 62222212.

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *CoRR*, abs/2204.05862.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Bricken, T.; Templeton, A.; Batson, J.; Chen, B.; Jermyn, A.; Conerly, T.; Turner, N.; Anil, C.; Denison, C.; Askell, A.; et al. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2.

Chen, W.; Song, D.; and Li, B. 2024. GRATH: Gradual Self-Truthifying for Large Language Models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Christiano, P. F.; Leike, J.; Brown, T. B.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 4299–4307.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *CoRR*, abs/1803.05457.

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. 2021. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168.

Dai, J.; Pan, X.; Sun, R.; Ji, J.; Xu, X.; Liu, M.; Wang, Y.; and Yang, Y. 2024. Safe RLHF: Safe Reinforcement Learning from Human Feedback. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

DeepSeek-AI; Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. DeepSeek-V3 Technical Report. *CoRR*, abs/2412.19437.

Dong, H.; Xiong, W.; Pang, B.; Wang, H.; Zhao, H.; Zhou, Y.; Jiang, N.; Sahoo, D.; Xiong, C.; and Zhang, T. 2024. RLHF Workflow: From Reward Modeling to Online RLHF. *Trans. Mach. Learn. Res.*, 2024.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.

Ferrando, J.; Obeso, O. B.; Rajamanoharan, S.; and Nanda, N. 2025. Do I Know This Entity? Knowledge Awareness and Hallucinations in Language Models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

He, Z.; Shu, W.; Ge, X.; Chen, L.; Wang, J.; Zhou, Y.; Liu, F.; Guo, Q.; Huang, X.; Wu, Z.; et al. 2024. Llama Scope: Extracting Millions of Features from Llama-3.1-8B with Sparse Autoencoders. *CoRR*, abs/2410.20526.

Huben, R.; Cunningham, H.; Smith, L. R.; Ewart, A.; and Sharkey, L. 2024. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. OpenAI o1 System Card. *CoRR*, abs/2412.16720.

Ji, J.; Hong, D.; Zhang, B.; Chen, B.; Dai, J.; Zheng, B.; Qiu, T. A.; Zhou, J.; Wang, K.; Li, B.; et al. 2025. PKU-SafeRLHF: Towards Multi-Level Safety Alignment for LLMs with Human Preference. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, 31983–32016. Association for Computational Linguistics.

Kim, S.; Suk, J.; Longpre, S.; Lin, B. Y.; Shin, J.; Welleck, S.; Neubig, G.; Lee, M.; Lee, K.; and Seo, M. 2024. Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, 4334–4353. Association for Computational Linguistics.

Li, J.; Sun, S.; Yuan, W.; Fan, R.; Zhao, H.; and Liu, P. 2024. Generative Judge for Evaluating Alignment. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Li, K.; Patel, O.; Viégas, F. B.; Pfister, H.; and Wattenberg, M. 2023. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Lieberum, T.; Rajamanoharan, S.; Conmy, A.; Smith, L.; Sonnerat, N.; Varma, V.; Kramár, J.; Dragan, A. D.; Shah, R.; and Nanda, N. 2024. Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2. *CoRR*, abs/2408.05147.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings*

of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, 3214–3252. Association for Computational Linguistics.

Liu, Z.; Wang, P.; Xu, R.; Ma, S.; Ruan, C.; Li, P.; Liu, Y.; and Wu, Y. 2025. Inference-Time Scaling for Generalist Reward Modeling. *CoRR*, abs/2504.02495.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov): 2579–2605.

Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M. S.; Love, J.; Tafti, P.; et al. 2024. Gemma: Open Models Based on Gemini Research and Technology. *CoRR*, abs/2403.08295.

Ouyang, L.; Wu, J.; Jiang, X.; et al. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Park, K.; Choe, Y. J.; and Veitch, V. 2024. The Linear Representation Hypothesis and the Geometry of Large Language Models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.

Shannon, C. E. 2001. A mathematical theory of communication. *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1): 3–55.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D. M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tu, S.; Lin, J.; Tian, X.; Zhang, Q.; Li, L.; Fu, Y.; Xu, N.; He, W.; Lan, X.; Jiang, D.; and Zhao, D. 2025. Enhancing LLM Reasoning with Iterative DPO: A Comprehensive Empirical Investigation. *CoRR*, abs/2503.12854.

Wang, C.; Gan, Y.; Huo, Y.; Mu, Y.; He, Q.; Yang, M.; Li, B.; Xiao, T.; Zhang, C.; Liu, T.; and Zhu, J. 2025. GRAM: A Generative Foundation Reward Model for Reward Generalization. In *Forty-second International Conference on Machine Learning*.

Wang, Z.; Dong, Y.; Delalleau, O.; Zeng, J.; Shen, G.; Egert, D.; Zhang, J.; Sreedhar, M. N.; and Kuchaiev, O. 2024. HelpSteer 2: Open-source dataset for training top-performing reward models. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Xiong, W.; Dong, H.; Ye, C.; Wang, Z.; Zhong, H.; Ji, H.; Jiang, N.; and Zhang, T. 2024. Iterative Preference Learning from Human Feedback: Bridging Theory and Practice for RLHF under KL-constraint. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yang, R.; Ding, R.; Lin, Y.; Zhang, H.; and Zhang, T. 2024. Regularizing Hidden States Enables Learning Generalizable Reward Model for LLMs. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Zhou, Z.; Yu, H.; Zhang, X.; Xu, R.; Huang, F.; and Li, Y. 2024. How Alignment and Jailbreak Work: Explain LLM Safety through Intermediate Hidden States. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, 2461–2488. Association for Computational Linguistics.

Zhu, L.; Wang, X.; and Wang, X. 2025. JudgeLM: Fine-tuned Large Language Models are Scalable Judges. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P. F.; and Irving, G. 2019. Fine-Tuning Language Models from Human Preferences. *CoRR*, abs/1909.08593.