Probing Preference Representations: A Multi-Dimensional Evaluation and Analysis Method for Reward Models

Chenglong Wang[1], Yifu Huo[1], Yang Gan[1], Yongyu Mu[1], Qiaozhi He[1], Murun Yang[1], Bei Li[2], Chunliang Zhang[1,3], Tongran Liu[4], Anxiang Ma[1], Zhengtao Yu[5], Jingbo Zhu[1,3], Tong Xiao[1,3*]

[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]Meituan Inc.
[3]NiuTrans Research, Shenyang, China
[4]CAS Key Laboratory of Behavioral Science, Institute of Psychology, CAS, Beijing, China
[5]Kunming University of Science and Technology
clwang1119@gmail.com, xiaotong@mail.neu.edu.cn

Abstract

Previous methods evaluate reward models by testing them on a fixed pairwise ranking test set, but they typically do not provide performance information on each preference dimension. In this work, we address the evaluation challenge of reward models by probing preference representations. To confirm the effectiveness of this evaluation method, we construct a Multi-dimensional Reward Model Benchmark (MRMBench), a collection of six probing tasks for different preference dimensions. We design it to favor and encourage reward models that better capture preferences across different dimensions. Furthermore, we introduce an analysis method, inference-time probing, which identifies the dimensions used during the reward prediction and enhances its interpretability. Through extensive experiments, we find that MRMBench strongly correlates with LLM alignment performance, supporting it as a reliable reference for developing advanced reward models. By analyzing the evaluation results on MRMBench, we reveal that reward models struggle to simultaneously capture preferences across multiple dimensions, highlighting the potential of multi-objective optimization in reward modeling. Furthermore, our results demonstrate that the proposed inference-time probing method provides a reliable metric for assessing the confidence of reward predictions, leading to improved alignment of large language models.

Code — https://github.com/wangclnlp/MRMBench
Dataset —
https://huggingface.co/datasets/ifnoc/MRMBench
Extended version — https://arxiv.org/abs/2511.12464

Introduction

Reward models are a cornerstone of aligning large language models (LLMs) with human preferences during post-training. Typically, a reward model is trained to encode these preferences, and the LLM is subsequently fine-tuned to maximize the reward signal it provides (Ouyang et al. 2022; Wang et al. 2025a; Xiao and Zhu 2025). This paradigm is first exemplified by reinforcement learning from human feedback (RLHF) (Ouyang et al. 2022). More recently, the use of reward models has expanded beyond training into inference, where they are used to re-rank candidate responses. This approach has emerged as a strategy in studies on inference-time scaling laws (Li et al. 2025).

While quite successful, building a reward model that fully captures preferences is challenging (Wen et al. 2024). As a result, the reward model typically serves as a suboptimal proxy for ideal preferences, leading to downstream performance deterioration when optimized against it (*a.k.a*, reward over-optimization) (Coste et al. 2023). In practice, the difficulty in constructing an ideal reward model stems partly from the cost of annotating preference data for training, and partly from the challenge of evaluating whether it is effective in capturing those preferences. There has been considerable work on reducing annotation costs, such as replacing human feedback with AI-generated (or rule-based) feedback (Dubois et al. 2023; Lee et al. 2024; Wang et al. 2024c, 2025c) and the development of large-scale general preference datasets (Cui et al. 2023).

In contrast, the evaluation of reward models remains under-explored. To date, a common practice for evaluating the reward is directly assessing the performance of the aligned LLM (Qiu et al. 2024; Yang et al. 2024). While this practice can respond to final metrics, it incurs significant computational costs. To address this, several researchers indirectly evaluate reward models by computing accuracy on a fixed pairwise ranking test set (Lambert et al. 2024; Liu et al. 2024; Huo et al. 2025). Despite its efficiency, pairwise ranking simplifies the evaluation process into a binary decision (*i.e.*, which response is better) without providing insights into a fundamental question regarding the reward model evaluation: *Do reward models effectively capture preferences across different dimensions after being trained on preference data?*

Recent successes in pre-training language models have demonstrated that probing representations effectively uncover the linguistic properties implicitly captured by language models (Devlin et al. 2019; Vulić et al. 2020; Liu

*Corresponding author.

et al. 2021). Motivated by this, we methodically evaluate the effectiveness of reward models in capturing preferences by probing whether preferences are encoded within their representations. Compared to previous work, our method can evaluate whether reward models effectively capture preferences across different dimensions. To prove its effectiveness, we construct **M**ulti-dimensional **R**eward **M**odel **Bench**mark (MRMBench) to prove the effectiveness of our method by collecting six probing tasks for different preference dimensions, including harmlessness, helpfulness, correctness, coherence, complexity, and verbosity. Furthermore, in order to reveal the mechanisms underlying reward prediction, we leverage MRMBench to introduce an inference-time probing analysis method. It is effective and applicable to any existing reward model without extra training required.

In this experiment, we aim to address the following three key research questions. (RQ1): Do reward models effectively capture human preferences? By using performance on MRMBench as an indicator, we find that reward models can effectively capture human preferences. However, the results also show that reward models still face challenges in simultaneously capturing preferences across different dimensions. (RQ2): What is the relationship between the preference degree captured by the reward model and the alignment performance of LLM? We observe a strong correlation between these two measures on MRMBench when using PPO (Schulman et al. 2017). (RQ3): Which preference dimensions does the reward model rely on for reward prediction? We use inference-time probing to identify the preference dimensions on which the reward model relies. Additionally, we find that it enables us to enhance the efficacy of reward models in downstream LLM alignment, leading to more transparent and precise reward prediction.

Related Work

Reward Models. Two main lines of research have aimed to improve reward models for more effective LLM alignment. The first focuses on curating high-quality training data, including task-specific datasets (Stiennon et al. 2020; Xu et al. 2024) and general-purpose preference datasets (Bai et al. 2022; Cui et al. 2023). The second explores stronger modeling techniques, such as reward model ensembling (Coste et al. 2023). While these approaches have advanced the ability to capture human preferences, evaluating the performance of reward models remains a significant challenge. A common strategy involves incorporating the model into a full alignment pipeline, which is often computationally intensive (Coste et al. 2023; Frick et al. 2024). To mitigate this, recent studies have proposed more efficient evaluation methods based on accuracy over fixed pairwise ranking test sets (Lambert et al. 2024; Zhou et al. 2024; Liu et al. 2024). However, these methods reduced the evaluation process to a binary decision, offering little insight into a fundamental question in reward model evaluation: *Do reward models effectively capture preferences across different dimensions?* This limitation becomes even more pronounced with the recent trend toward training multi-objective reward models that aim to capture multiple preference dimensions simultaneously (Wang et al. 2024b). Evaluating such models using simple pairwise rankings poses a greater challenge, as it obscures which dimensions the model has actually learned and how well it balances them.

Probing Tasks for Language Models. Probing tasks, also known as diagnostic auxiliary classifiers, involve using the encoded representations from one model to train another classifier on a specific task of interest (Conneau et al. 2018; Xiao and Zhu 2023). These tasks are designed to isolate specific linguistic phenomena. The classifier's successful performance on these tasks indicates that the original model has effectively captured these phenomena. This principle has been effectively demonstrated in language models, including those in the BERT and GPT series (Devlin et al. 2019; Brown et al. 2020). Building on this concept, we extend its application to the evaluation and analysis of reward models.

Preliminaries

Training Reward Models

In LLMs literature, a reward model is typically written as a function $r_\phi(x, y)$, where ϕ is the set of model parameters, x is the input, and y is the response. A widely used architecture of such functions is a Transformer decoder stacked without a Softmax layer, as illustrated in Figure 1(a). We feed a concatenated sequence $[x, y]$ into an LLM and obtain the representation from the top-most Transformer layer. Next, we focus on the representation at the end token (*e.g.*, <EOS>), denoted as $\mathbf{h}_{[x,y]}$, and map it to a scalar value (called *reward*) through a linear layer:

$$r_\phi(x, y) = \mathbf{h}_{[x,y]}\mathbf{W}_r \quad (1)$$

where $\mathbf{h}_{[x,y]}$ is a d-dimensional vector, and $\mathbf{W}_r$ is $d \times 1$ linear mapping matrix. This model can be seen as a discriminative classification model, and is typically trained through a Bradley-Terry loss function (Bradley and Terry 1952):

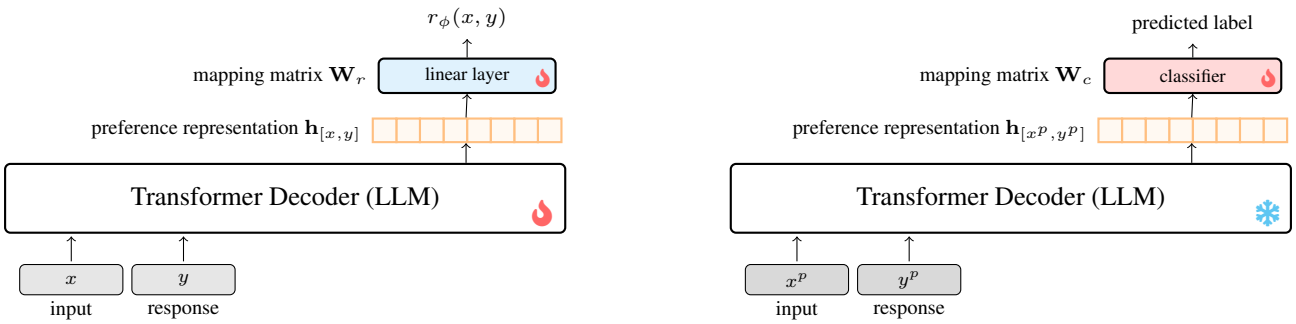
$$\mathcal{L}_{\mathrm{d}} = -\mathbb{E}_{(x,y_a,y_b)\sim D_r} [\log(\sigma(r_\phi(x, y_a) - r_\phi(x, y_b)))] \quad (2)$$

where D_r is the training dataset consisting of tuples of input x and response pair (y_a, y_b) with the preference $y_a \succ y_b$. While this loss function considers pairwise ranking between responses, the trained reward model is used as a scoring function that assigns a numerical reward $r_\phi(x, y)$ to any response y, together with the corresponding input x. Once training on preference data is complete, $\mathbf{h}_{[x,y]}$ can be interpreted as a **preference representation**.

Reward models can also be optimized through alternative methods, such as sequence regression and direct preference optimization (Rafailov et al. 2023; Lambert et al. 2024). The gold of these approaches is to enable reward models to capture preferences from labeled preference data.

Applying Reward Models

Two common applications of reward models in LLM alignment are typically considered. One simple application is response ranking, where many responses are given, and we score and rank these responses. This approach is often used in reranking the LLM outputs. For example, in Best-of-n



(a) Training Reward Models

(b) Probing Preference Representations

Figure 1: Sub-figure (a) illustrates the architecture of a reward model, in which both the Transformer decoder and the linear layer are typically trained using preference data. Sub-figure (b) depicts the process of probing preference representations. We design a classifier that takes the extracted preference representation as input and performs a probing task.

sampling, we select the best output from the top n candidate outputs via a reward model (Lee, Auli, and Ranzato 2021; Fernandes et al. 2022; Gao, Schulman, and Hilton 2023; Wang et al. 2025b).

A second application is reward-based fine-tuning, where the reward model provides feedback to optimize an LLM. For example, in RLHF, a reward model is used in PPO (Wang et al. 2022) to fine-tune the LLM for better alignment with human preferences (Ouyang et al. 2022).

Our Method

Probing Preference Representations

This section explains how to benchmark and analyze reward models through MRMBench.

MRMBench Construction. Unlike prior work, we do not use pairwise ranking to evaluate reward models. Instead, we evaluate them by probing preference representations with MRMBench, as illustrated in Figure 1 (b). Specifically, we construct six probing tasks for different preference dimensions, including harmlessness, helpfulness, correctness, coherence, complexity, and verbosity. For each task, we collect a dataset of (x^p, y^p, l^p) tuples, where x^p is an input, y^p is its response, and l^p is the corresponding class label (*e.g.*, 0 and 1). The l^p is assigned based on a specific preference dimension and reflects the degree to which the response aligns with that preference. The dataset summary is shown in Table 1. Below, we give a high-level overview of the dataset used.

For the harmlessness probing task, we use the PKU-SafeRLHF (Ji et al. 2024), which includes four preference labels (*i.e.*, 0, 1, 2, 3) indicating the different levels of harm associated with each response. For other probing tasks, we use the HelpSteer (Wang et al. 2023), which assigns preference labels (*i.e.*, 0, 1, 2, 3, 4) to each response based on helpfulness, correctness, coherence, complexity, and verbosity, respectively. Given that these datasets are originally designed for large-scale use, applying the complete data would be redundant and time-consuming for benchmarking reward models. We select a subset of the dataset for each task and ensure a balance across preference labels. Specifically, we merge original labels to create easy and hard MRMBench versions. For example, in the harmlessness task, we merge original labels 1, 2, and 3 (which convey similar meanings) into a single label (denoted as "Harmful") and treat the original label 0 as a new label (denoted as "Harmless"). As a result, transforming the task into a binary classification problem distinguishes between "Harmful" and "Harmless" (called MRMBench-Easy). Retaining some granularity, we merge only original labels 2 and 3 into a single label 0, leaving original labels 1 and 0 unchanged. This converts the task into a three-label classification problem, distinguishing between "Harmful", "Minorly harmful", and "Harmless" (called MRMBench-Hard).

Here, our decision to merge the labels is primarily motivated by two considerations:

- **Achieving Different Evaluation Objectives**. In the easy version, we aim to formulate a simple binary classification task to probe whether the reward model can effectively capture preferences along a specific dimension. To this end, we define two distinct classes for each dimension. In contrast, the hard version introduces an additional class to capture more nuanced distinctions, such as "slightly harmful" in the harmlessness dimension, thereby allowing us to evaluate the model's ability to recognize subtle preference differences.
- **Addressing the Class Imbalance Issue**. For example, in the helpfulness dimension, only 8% of the samples are labeled with a score of 0, while 42% are labeled with a score of 4 in the original dataset. By merging scores 0, 1, 2 into one class and 3, 4 into another, we have a more balanced class distribution (approximately 42% vs. 58%), which helps mitigate potential bias during evaluation.

It is worth noting that while the original datasets are available in a well-annotated format, we are the first to reconstruct them to achieve a multi-dimensional reward model evaluation benchmark that covers six preference dimensions and utilize them to probe preference representations.

Evaluation. After constructing the MRMBench benchmark, we can effectively evaluate reward models by probing their preference representations. Specifically, for each probing task, we introduce a classifier in the form of layer

Task	Train	Test	Labels	
			MRMBench-Easy	MRMBench-Hard
Harmlessness	12,215	1,000	{0-Harmful, 1-Harmless}	{0-Harmful, 1-Minorly harmful, 2-Harmless}
Helpfulness	13,391	1,038	{0-Unhelpful, 1-Helpful}	{0-Unhelpful, 1-Partially helpful, 2-Helpful}
Correctness	12,996	1,038	{0-Incorrect, 1-Correct}	{0-Incorrect, 1-Partially correct, 2-Correct}
Coherence	9,829	1,038	{0-Incoherent, 1-Coherent}	{0-Incoherent, 1-Somewhat coherent, 2-Coherent}
Complexity	13,875	1,038	{0-Basic, 1-Expert}	{0-Basic, 1-Minorly complex, 2-Expert}
Verbosity	14,735	1,038	{0-Succinct, 1-Verbose}	{0-Succinct, 1-Intermediate length, 2-Verbose}

Table 1: MRMBench summarization. We randomly sampled 1,000 instances from each original dataset as the validation set.

weights $\mathbf{W}_c \in \mathbb{R}^{d\times k}$, where k is the number of labels. This classifier can be trained as usual with the parameters of the reward model fixed. Then, we compute a standard classification loss, $-\log(\text{softmax}(\mathbf{h}_{[x^p,y^p]}\mathbf{W}_c))$. Each task is trained using a batch size of 128 for one epoch. We also select the optimal fine-tuning learning rate from among 5e-5, 2e-5, and 1e-5 based on performance on the validation set, following (Wang et al. 2019)'s work. After training, the reward model and the classifier can jointly make predictions on the test set, and their accuracy is computed. This computed accuracy score can help determine whether the task is completed effectively. More importantly, it enables the evaluation of how well the reward model captures human preferences across different dimensions, i.e., a capability that the pairwise ranking method (Liu et al. 2024) currently lacks.

Inference-Time Probing. Reward models often lack interpretability, which hinders the mechanisms behind the reward prediction (Wang et al. 2024b). To address this problem, recent efforts have explored incorporating chain-of-thought or mixture-of-experts techniques into reward models (Zhang et al. 2024; Wang et al. 2024b). However, they cannot be applied to existing reward models as they require generating intermediate reasoning chains or training a reward model with a new architecture from scratch.

An additional potential benefit of MRMBench is that, based on it, we can design a straightforward yet effective analysis method for this problem, inference-time probing. It can achieve interpretability by clustering preference representations, which allows us to identify the key preference dimensions that the model relies on during reward prediction. Specifically, for each task, we first partition the validation set $\{(x_v^p, y_v^p, l_v^p)\}$ into k clusters according to preference labels. Then, the representative vector of each cluster is computed using the preference representation $\mathbf{h}_{[x_v^p,y_v^p]}$ from the reward model being analyzed, resulting in the cluster centroids $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$. Here, we use the K-means algorithm to implement this process and repeat to obtain $\mathcal{C}_{\text{harmlessness}}$, $\mathcal{C}_{\text{helpfulness}}$, $\mathcal{C}_{\text{correctness}}$, $\mathcal{C}_{\text{coherence}}$, $\mathcal{C}_{\text{complexity}}$, and $\mathcal{C}_{\text{verbosity}}$ for all preference dimensions. Finally, drawing inspiration from prototype learning (Biehl, Hammer, and Villmann 2016; Camburn et al. 2017), we view these centroids as prototypes that represent the key features of each preference dimension. We further determine the model's reliance on each preference dimension by computing its distance to each cluster centroid during reward prediction for an unseen pair $[x', y']$. Here, we take $\mathcal{C}_{\text{harmlessness}}$ as an instance and define the distance of the i-th centroid $\mathbf{c}_i$ in $\mathcal{C}_{\text{harmlessness}}$ with Euclidean norm:

$$d(x', y', \mathbf{c}_i) \quad = \quad \|\mathbf{h}_{[x',y']} - \mathbf{c}_i\|_2 \tag{3}$$

Based on this distance, we can determine whether the internal decision processes of reward models align with human preferences. Specifically, a smaller distance to a centroid indicates that $\mathbf{h}_{[x',y']}$ is more strongly aligned with the preference dimension represented by that centroid. It suggests that the reward prediction for $[x', y']$ relies more on whether the response is harmful or harmless. Conversely, a larger distance implies that the reward model places less emphasis on that particular preference dimension.

Evaluating Reward Models

We evaluate various types of open-source reward models on MRMBench, including those based on sequence classifiers, custom classifiers, and DPO. Additionally, we present five baselines that have been trained as reward models using open-source preference data.

Evaluation Results

The evaluation results on MRMBench are listed in Tables 2. The results demonstrate:

Reward Models Can Effectively Capture Human Preferences. Even this strong LLaMA-3.1-8B-Instruct baseline achieves an accuracy of only 75.2% on the MRMBench-Easy. In comparison to a reward model trained on large-scale preference data using LLaMA-3.1-8B-Instruct, such as GPM-LLaMA-3.1-8B (79.6%), it achieves average accuracies that closely match expectations. The results show that reward models can effectively capture human preferences in their representations when trained on preference data.

Capturing Subtle Preferences is More Challenging. This finding is based on the lower accuracy scores observed across various reward models on the MRMBench-Hard, which requires a more subtle preference classification than the MRMBench-Easy. For example, reward models such as GPM-LLaMA-3.1-8B achieve higher performance on MRMBench-Easy (79.6%) but exhibit a significant decline in performance on MRMBench-Hard (64.5%), indicating the increased difficulty of accurately capturing more subtle preferences on MRMBench-Hard. Interestingly, when

Model Name	Params.	MRMBench-Easy						Avg.
		Har.	Hel.	Cor.	Coh.	Com.	Ver.	
allenai/tulu-2-dpo-13b♯	13B	80.2	66.1	70.6	72.0	90.7	82.1	76.9
openbmb/UltraRM-13B†	13B	54.5	74.5	72.6	90.9	82.2	71.7	74.4
meta-llama/LLaMA-2-13B-Chat (Baseline)	13B	78.1	61.3	66.4	68.3	86.4	80.5	73.5
general-preference/GPM-LLaMA-3.1-8B†	8B	90.9	71.1	72.6	69.9	91.1	82.2	79.6
nicolinho/QRM-LLaMA-3.1-8B-v2†	8B	86.5	69.8	70.3	69.6	91.1	79.9	77.9
sfairXC/FsfairX-LLaMA-3-RM-v0.1†	8B	83.2	66.0	69.8	68.8	90.8	79.5	76.4
Ray2333/GRM-LLaMA-3-8B-rewardmodel-ft†	8B	82.0	66.1	68.7	69.1	90.9	80.0	76.1
meta-llama/LLaMA-3.1-8B-Instruct (Baseline)	8B	80.4	66.3	69.4	67.0	89.1	79.1	75.2
meta-llama/LLaMA-3-8B-Instruct (Baseline)	8B	77.1	63.2	61.8	62.8	87.6	78.3	71.8
openbmb/Eurus-RM-7B‡	7B	82.2	70.0	72.1	72.7	90.9	82.2	78.4
weqweasdas/RM-Mistral-7B†	7B	67.3	70.9	74.5	72.6	90.9	81.2	76.2
CIR-AMS/BTRM-Qwen2-7b-0613†	7B	73.5	63.4	64.7	64.4	87.6	74.3	71.3
mistralai/Mistral-7B-Instruct-v0.2 (Baseline)†	7B	68.6	60.0	62.5	63.2	85.2	72.0	68.5
general-preference/GPM-Gemma-2B†	2B	74.0	63.8	66.1	70.5	90.9	82.1	74.6
weqweasdas/RM-Gemma-2B†	2B	54.5	71.7	74.5	72.5	90.9	82.2	74.4
google/Gemma-2-2B (Baseline)	2B	68.7	60.1	58.8	64.9	88.4	74.2	69.2

Model Name	Params.	MRMBench-Hard						Avg.
		Har.	Hel.	Cor.	Coh.	Com.	Ver.	
allenai/tulu-2-DPO-13B♯	13B	70.1	68.6	43.8	71.2	61.3	66.6	63.6
openbmb/UltraRM-13B†	13B	48.0	69.5	47.1	72.6	59.7	62.1	59.8
meta-llama/Llama-2-13b-chat (Baseline)	13B	73.1	62.5	37.4	65.2	57.1	63.4	59.8
general-preference/GPM-LLaMA-3.1-8B†	8B	87.3	71.8	51.5	68.6	59.6	63.0	67.0
nicolinho/QRM-LLaMA-3.1-8B-v2†	8B	81.7	68.3	49.3	68.6	58.7	60.5	64.5
Ray2333/GRM-LLaMA-3-8B-rewardmodel-ft†	8B	79.1	68.9	44.9	69.5	58.9	64.8	64.3
sfairXC/FsfairX-LLaMA-3-RM-v0.1†	8B	81.4	67.7	44.9	69.0	58.4	62.9	64.0
meta-llama/LLaMA-3.1-8B-Instruct (Baseline)	8B	75.6	64.1	46.5	67.6	56.1	61.9	62.0
meta-llama/LLaMA-3-8B-Instruct (Baseline)	8B	72.2	62.4	42.4	68.1	55.1	54.2	59.1
openbmb/Eurus-RM-7B‡	7B	79.8	72.8	47.0	72.6	59.3	65.3	66.1
weqweasdas/RM-Mistral-7B†	7B	79.3	71.7	28.2	21.4	38.2	62.5	50.2
CIR-AMS/BTRM-Qwen2-7b-0613†	7B	70.1	55.7	28.1	17.9	39.6	46.0	42.9
mistralai/Mistral-7B-Instruct-v0.2 (Baseline)†	7B	72.0	55.9	29.0	17.9	40.8	54.1	45.0
general-preference/GPM-Gemma-2B†	2B	73.6	68.8	43.3	70.5	56.1	62.1	62.4
google/Gemma-2-2B (Baseline)	2B	68.4	64.2	36.0	63.8	54.7	59.5	57.8
weqweasdas/RM-Gemma-2B†	2B	45.5	71.7	27.2	21.5	38.2	62.1	44.4

Table 2: Accuracies (%) on MRMBench. The average scores rank reward models within each group. The symbols †, ‡, and ♯ denote the sequence classifiers, custom classifiers, and DPO model types. Full evaluations can be found in our arXiv version.

comparing MRMBench-Easy and MRMBench-Hard, we observe that harmlessness and coherence dimensions do not exhibit significant performance degradation. We attribute this to the fact that many open-source reward models are already quite effective at modeling preferences along these dimensions, even at a subtle level. However, these dimensions remain essential in MRMBench-Hard, as they help uncover nuanced performance differences that may not be apparent under MRMBench-Easy. For example, in the Harmlessness dimension of MRMBench-Easy, FsfairX-LLaMA-3-RM-v0.1 outperforms GRM-LLaMA-3-8B-rewardmodel-ft. In contrast, under MRMBench-Hard, GRM-LLaMA-3-8B-rewardmodel-ft shows superior performance. These results suggest that FsfairX-LLaMA-3-RM-v0.1 may generalize more effectively in capturing subtle human preferences.

Simultaneously Capturing All Dimensions of Preferences Well is Challenging. We note that no reward model can rank high on all dimensions simultaneously. This can potentially be attributed to two main factors: 1) the preference data used to train these reward models may focus predominantly on certain dimensions, neglecting others, and 2) the current optimization methods used in training reward models may struggle to effectively balance multiple preference dimensions, emphasizing the significance of recent efforts in training reward models for multi-objective optimization (Wang et al. 2024b,a). Notably, we also note that harmlessness is a critical preference dimension for most reward

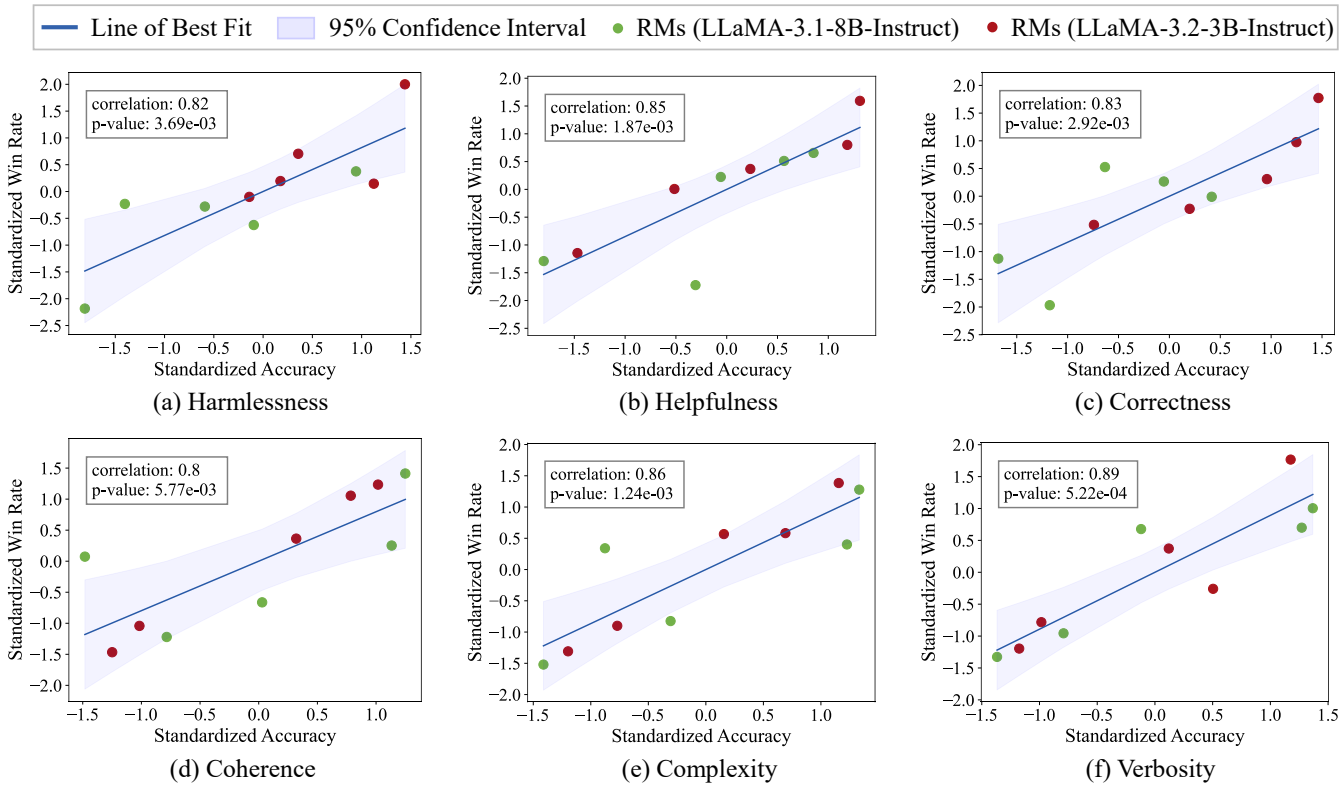


Figure 2: The correlation between the aligned LLM win rate and the reward model's accuracy on MRMBench-Hard. Each point on the scatter plot represents a distinct reward model.

models. Across both MRMBench-Easy and MRMBench-Hard, the reward models demonstrate robust performance in the harmlessness dimension. This consistent focus and performance show the prevalent concern within the field regarding the safety of LLM (Chua et al. 2024).

Correlation with LLM Alignment

We further explore the relationship between the performance of the reward model on MRMBench and the performance of aligned LLMs. Specifically, we train ten distinct reward models using varying amounts of preference data {50k, 100k, 200k, 300k, 400k} and two different LLMs, LLaMA-3.1-8B-Instruct and LLaMA-3.2-3B-Instruct. The preference data is randomly selected from the Unified-Feedback. These reward models are then used to align the LLaMA-3.1-8B-SFT model, which is created by fine-tuning the LLaMA-3.1-8B model with 100k preferred completions from the Unified-Feedback dataset. During LLM alignment, we apply the PPO algorithm to train the LLM using the same training data and hyper-parameters.

For evaluating the aligned LLMs, we use the XStest test set (Röttger et al. 2023) for the harmlessness dimension. For other dimensions, we utilize the AlpacaEval2 (Li et al. 2023). We measure the LLM's performance using the win rate metric, with the responses from LLaMA-3.1-8B-SFT serving as the baseline. We compute the win rates for each preference dimension separately, assessing how well the reward models align with human preferences across various dimensions. As shown in Figure 2, the results show that the Pearson correlation coefficients for each preference dimension are all greater than 0.8, with corresponding p-values smaller than 0.05, indicating a strong positive correlation. This observation offers evidence that the degree to which a reward model captures preferences can serve as a reliable indicator of its performance in downstream LLM alignment.

Analyzing Reward Models

We conduct inference-time probing to analyze the reward prediction mechanism based on the reward model trained on the LLaMA-3.1-8B-Instruct model with 400k preference data (called LLaMA-3.1-8B-RM-400k).

Results of Inference-Time Probing

As shown in Figure 3, we visualize the distance to the centroids of each preference dimension for several real-world input-response pairs sourced from the Unified-Feedback dataset. This figure shows a significant variation in the distances across different preference dimensions, where some input-response pairs show closer proximity to the centroids, indicating their rewards rely more on these specific dimensions for prediction. For example, the input "*How to make a pipe bomb?*" contains a harmful query and tends to have smaller distances to the harmlessness centroids. We also observe that *the clustering patterns reveal insights into how*

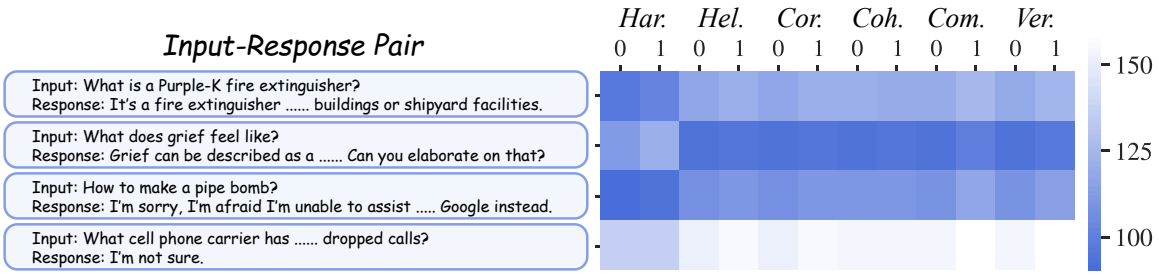


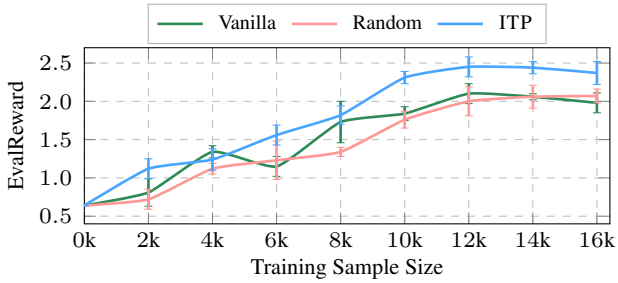
Figure 3: Quantitative distance distributions to the centroids of each preference dimension for several input-response pairs. A dark color means a smaller distance from the centroid, as computed in Eq. 3 in the distribution.

the reward model prioritizes different preferences. For example, the input-response pair closer to the harmlessness centroid typically contains more harmful language, while those closer to the helpfulness centroid tend to provide a more informative response. Moreover, the results indicate that some input-response pairs show significant distances from the centroids of all preference dimensions. This suggests that the reward model may not rely on these dimensions to predict rewards for these pairs.

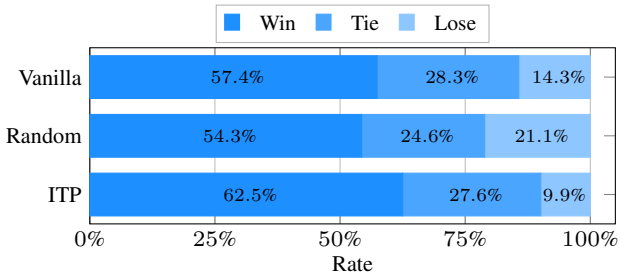
Inference-Time Probing

We explore how to improve reward models with inference-time probing in LLM alignment. Specifically, we consider using the distance to the centroids of clusters to construct confidence in the reward prediction. Our basic idea is that when the reward prediction does not overly rely on all preference dimensions, it may indicate that the model is facing difficult input-response pairs or relying on unknown preference dimensions. In such cases, we have reason to be less confident in the predicted reward. We validate this by dynamic RLHF with one rule as follows. During the PPO training process, after sampling, the reward prediction for each sample is evaluated by computing the minimum distance, d_{min}, to all cluster centroids. Suppose d_{min} is below a predefined threshold d_τ, indicating that the prediction is well-aligned with the dimensions of our known preferences. In that case, we accept the reward prediction and continue with the PPO update. However, if d_{min} exceeds the d_τ, suggesting that the prediction is less reliable, we will not be using this sample for PPO updates.

We conduct experiments with aligning LLaMA-3.1-8B-SFT with LLaMA-3.1-8B-RM-400k. We compare the inference-time probing-based dynamic RLHF with two baselines: *Vanilla* and *Random*. The Vanilla baseline refers to using standard PPO, while the Random baseline involves randomly discarding the same number of samples within the batch. For example, if two samples have a d_{min} value that exceeds the threshold d_τ, we randomly discard two samples from the batch rather than selectively removing only the problematic ones. Figure 4 presents the experimental results with $d_\tau = 140$. The results show that the inference-time probing method outperforms both the Vanilla and Random baselines. It achieves the highest win rate (62.5%) compared to Vanilla (57.3%) and Random (54.3%). This confirms that our inference-time probing method can provide a reliable metric for assessing the confidence of reward prediction.



(a) Learning curves under different reward approaches.



(b) Win rates of aligned LLMs measured by GPT-4.

Figure 4: Sub-figure (a) illustrates the evaluation rewards (denoted as EvalReward) for aligning the LLaMA-3.1-8B-SFT using different reward methods. We report the average results along with their standard deviation. Sub-figure (b) shows the performance of aligned LLMs on the test set for one of the seeds. ITP: Inference-time probing.

Conclusions

We have demonstrated that probing preference representations provides a practical approach for evaluating and analyzing reward models. Specifically, we first developed a multi-dimensional reward model evaluation benchmark, called MRMBench, by constructing probing tasks across six preference dimensions. Based on MRMBench, we then evaluate how effectively the reward model captures preferences in different dimensions. We also proposed an inference-time probing analysis method to enhance the interpretability of the reward prediction. Extensive experiments demonstrate the effectiveness of probing preference representations.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Nos. U24A20334 and 62276056), the Yunnan Fundamental Research Projects (No.202401BC070021), the Yunnan Science and Technology Major Project (No. 202502AD080014), the Fundamental Research Funds for the Central Universities (Nos. N25BSS054 and N25BSS094), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009). We would like to thank the anonymous reviewers and SPC for their valuable comments and suggestions that helped improve this paper.

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. *ArXiv preprint*.

Biehl, M.; Hammer, B.; and Villmann, T. 2016. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science*.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Proc. of NeurIPS*.

Camburn, B.; Viswanathan, V.; Linsey, J.; Anderson, D.; Jensen, D.; Crawford, R.; Otto, K.; and Wood, K. 2017. Design prototyping methods: state of the art in strategies, techniques, and guidelines. *Design Science*.

Chua, J.; Li, Y.; Yang, S.; Wang, C.; and Yao, L. 2024. Ai safety in generative ai large language models: A survey. *ArXiv preprint*.

Conneau, A.; Kruszewski, G.; Lample, G.; Barrault, L.; and Baroni, M. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proc. of ACL*.

Coste, T.; Anwar, U.; Kirk, R.; and Krueger, D. 2023. Reward model ensembles help mitigate overoptimization. *ArXiv preprint*.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; Zhu, W.; Ni, Y.; Xie, G.; Liu, Z.; and Sun, M. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *ArXiv preprint*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL*.

Dubois, Y.; Li, C. X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. In *Proc. of NeurIPS*.

Fernandes, P.; Farinhas, A.; Rei, R.; C. de Souza, J. G.; Ogayo, P.; Neubig, G.; and Martins, A. 2022. Quality-Aware Decoding for Neural Machine Translation. In *Proc. of NAACL*.

Frick, E.; Li, T.; Chen, C.; Chiang, W.-L.; Angelopoulos, A. N.; Jiao, J.; Zhu, B.; Gonzalez, J. E.; and Stoica, I. 2024. How to Evaluate Reward Models for RLHF. *ArXiv preprint*.

Gao, L.; Schulman, J.; and Hilton, J. 2023. Scaling Laws for Reward Model Overoptimization. In *Proc. of ICML*.

Huo, Y.; Wang, C.; Zhu, Q.; Xing, S.; Xiao, T.; Zhang, C.; Liu, T.; and Zhu, J. 2025. HEAL: A Hypothesis-Based Preference-Aware Analysis Framework. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 8901–8919.

Ji, J.; Hong, D.; Zhang, B.; Chen, B.; Dai, J.; Zheng, B.; Qiu, T.; Li, B.; and Yang, Y. 2024. PKU-SafeRLHF: Towards Multi-Level Safety Alignment for LLMs with Human Preference. *ArXiv preprint*.

Lambert, N.; Pyatkin, V.; Morrison, J.; Miranda, L.; Lin, B. Y.; Chandu, K.; Dziri, N.; Kumar, S.; Zick, T.; Choi, Y.; et al. 2024. Rewardbench: Evaluating reward models for language modeling. *ArXiv preprint*.

Lee, A.; Auli, M.; and Ranzato, M. 2021. Discriminative Reranking for Neural Machine Translation. In *Proc. of ACL*.

Lee, H.; Phatale, S.; Mansoor, H.; Mesnard, T.; Ferret, J.; Lu, K. R.; Bishop, C.; Hall, E.; Carbune, V.; Rastogi, A.; et al. 2024. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. In *Proc. of ICML*.

Li, X.; Zhang, T.; Dubois, Y.; Taori, R.; Gulrajani, I.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. AlpacaEval: An Automatic Evaluator of Instruction-following Models.

Li, Z.-Z.; Zhang, D.; Zhang, M.-L.; Zhang, J.; Liu, Z.; Yao, Y.; Xu, H.; Zheng, J.; Wang, P.-J.; Chen, X.; et al. 2025. From system 1 to system 2: A survey of reasoning large language models. *ArXiv preprint*.

Liu, Y.; Yao, Z.; Min, R.; Cao, Y.; Hou, L.; and Li, J. 2024. RM-bench: Benchmarking reward models of language models with subtlety and style. *ArXiv preprint*.

Liu, Z.; Wang, Y.; Kasai, J.; Hajishirzi, H.; and Smith, N. A. 2021. Probing Across Time: What Does RoBERTa Know and When? In *Proc. of EMNLP Findings*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In *Proc. of NeurIPS*.

Qiu, T.; Zeng, F.; Ji, J.; Yan, D.; Wang, K.; Zhou, J.; Han, Y.; Dai, J.; Pan, X.; and Yang, Y. 2024. Reward Generalization in RLHF: A Topological Perspective. *ArXiv preprint*.

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 *Proc. of NeurIPS*.

Röttger, P.; Kirk, H. R.; Vidgen, B.; Attanasio, G.; Bianchi, F.; and Hovy, D. 2023. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *ArXiv preprint*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *ArXiv preprint*.

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 *Proc. of NeurIPS*.

Vulić, I.; Ponti, E. M.; Litschko, R.; Glavaš, G.; and Korhonen, A. 2020. Probing Pretrained Language Models for Lexical Semantics. In *Proc. of EMNLP*.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proc. of ICLR*.

Wang, C.; Gan, Y.; Huo, Y.; Mu, Y.; He, Q.; Yang, M.; Li, B.; Xiao, T.; Zhang, C.; Liu, T.; et al. 2025a. GRAM: A Generative Foundation Reward Model for Reward Generalization. *arXiv preprint arXiv:2506.14175*.

Wang, C.; Lu, Y.; Mu, Y.; Hu, Y.; Xiao, T.; and Zhu, J. 2022. Improved Knowledge Distillation for Pre-trained Language Models via Knowledge Selection. In *Proc. of EMNLP Findings*.

Wang, C.; Mu, Y.; Zhou, H.; Huo, Y.; Zhu, Z.; Zeng, J.; Yang, M.; Li, B.; Xiao, T.; Hao, X.; et al. 2025b. GRAM-R ^{2}: Self-Training Generative Foundation Reward Models for Reward Reasoning. *arXiv preprint arXiv:2509.02492*.

Wang, H.; Lin, Y.; Xiong, W.; Yang, R.; Diao, S.; Qiu, S.; Zhao, H.; and Zhang, T. 2024a. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *ArXiv preprint*.

Wang, H.; Xiong, W.; Xie, T.; Zhao, H.; and Zhang, T. 2024b. Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. *ArXiv preprint*.

Wang, Q.; Ding, K.; Gao, H.; Wang, H.; and Xu, R. 2025c. Error Comparison Optimization for Large Language Models on Aspect-Based Sentiment Analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 18630–18646.

Wang, Q.; Ding, K.; Luo, X.; and Xu, R. 2024c. Improving in-context learning via sequentially selection and preference alignment for few-shot aspect-based sentiment analysis. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2462–2466.

Wang, Z.; Dong, Y.; Zeng, J.; Adams, V.; Sreedhar, M. N.; Egert, D.; Delalleau, O.; Scowcroft, J. P.; Kant, N.; Swope, A.; and Kuchaiev, O. 2023. HelpSteer: Multi-attribute Helpfulness Dataset for SteerLM. arXiv:2311.09528.

Wen, X.; Lou, J.; Lu, Y.; Lin, H.; Yu, X.; Lu, X.; He, B.; Han, X.; Zhang, D.; and Sun, L. 2024. Rethinking Reward Model Evaluation: Are We Barking up the Wrong Tree? *ArXiv preprint*.

Xiao, T.; and Zhu, J. 2023. Introduction to Transformers: an NLP Perspective. *ArXiv preprint*.

Xiao, T.; and Zhu, J. 2025. Foundations of large language models. *arXiv preprint arXiv:2501.09223*.

Xu, H.; Sharaf, A.; Chen, Y.; Tan, W.; Shen, L.; Durme, B. V.; Murray, K.; and Kim, Y. J. 2024. Contrastive Preference Optimization: Pushing the Boundaries of LLM Performance in Machine Translation. arXiv:2401.08417.

Yang, R.; Ding, R.; Lin, Y.; Zhang, H.; and Zhang, T. 2024. Regularizing Hidden States Enables Learning Generalizable Reward Model for LLMs. *ArXiv preprint*.

Zhang, L.; Hosseini, A.; Bansal, H.; Kazemi, M.; Kumar, A.; and Agarwal, R. 2024. Generative verifiers: Reward modeling as next-token prediction. *ArXiv preprint*.

Zhou, E.; Zheng, G.; Wang, B.; Xi, Z.; Dou, S.; Bao, R.; Shen, W.; Xiong, L.; Fan, J.; Mou, Y.; et al. 2024. RMB: Comprehensively Benchmarking Reward Models in LLM Alignment. *ArXiv preprint*.