GENERAL PREFERENCE MODELING WITH PREFERENCE REPRESENTATIONS FOR ALIGNING LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modeling human preferences is crucial for aligning foundation models with human values. Traditional reward modeling methods, such as the Bradley-Terry (BT) reward model, fall short in expressiveness, particularly in addressing intransitive preferences. Although supervised pair preference models and pair reward models can express general preferences, their implementation is highly ad-hoc and cannot guarantee a consistent preference probability of compared pairs. Additionally, they impose high computational costs due to their quadratic query complexity when comparing multiple responses. In this paper, we introduce *preference representation learning*, an approach that embeds responses into a latent space to capture intricate preference structures efficiently, achieving linear query complexity. Additionally, we propose preference score-based General Preference Optimization (GPO), which generalizes reward-based reinforcement learning from human feedback. Experimental results show that our General Preference representation model (GPM) outperforms the BT reward model on the RewardBench benchmark with a margin of up to 9.1% and effectively models cyclic preferences where any BT reward model behaves like a random guess. Furthermore, evaluations on downstream tasks such as AlpacaEval2.0, following the language model post-training with GPO and our general preference model, reveal substantial performance improvements with margins up to 8.3%. These findings indicate that our method may enhance the alignment of foundation models with nuanced human values.

1 INTRODUCTION

Modeling human preferences is a cornerstone in developing foundation models that interact seamlessly with users. In natural language modeling and reinforcement learning, aligning models with human intent and values has led to significant advancements, including improved text generation and enhanced decision-making policies (Ouyang et al., 2022; Christiano et al., 2017). Traditional approaches often rely on reward modeling, wherein a reward function is learned to guide the optimization of policies. While effective in certain contexts, these methods face expressiveness and computational efficiency challenges, particularly when addressing complex or intransitive human preferences (Tversky, 1969; Munos et al., 2023).

Preference learning algorithms typically employ pairwise comparisons to capture human judgments (Ibarz et al., 2018; Ziegler et al., 2019). The Bradley-Terry (BT) model (Bradley & Terry, 1952) is popular for modeling such pairwise preferences due to its simplicity and computational efficiency: given K responses, a BT reward model cost $\mathcal{O}(K)$ inference-time compute to output the reward dictating the preferences. The efficiency of the BT model comes from the implicit assumption that each option can be conveniently represented by a scalar reward, which inevitably limits the model's capacity to capture the richness of human judgments that may be context-dependent or exhibit intransitivity (Gardner, 1970).

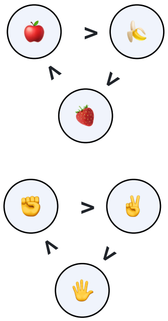
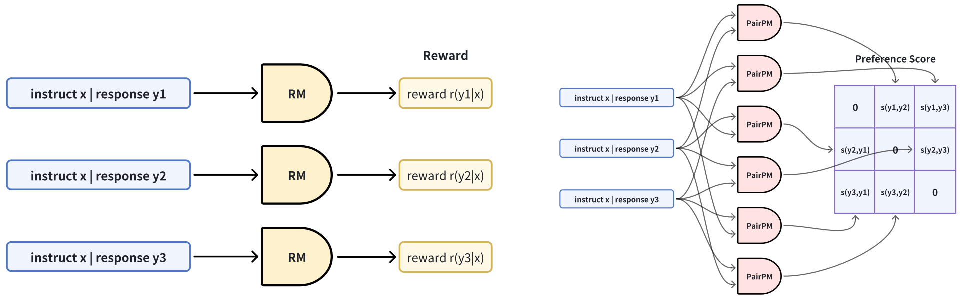
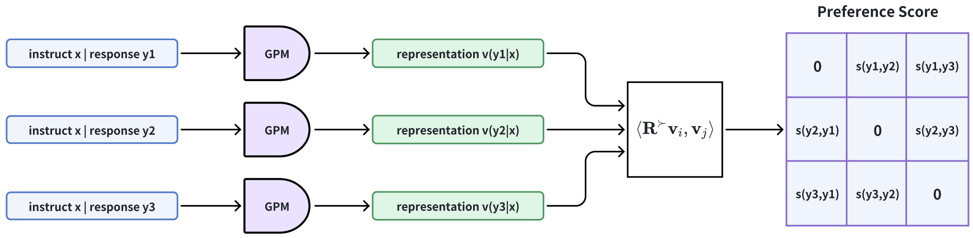


Figure 1: Intransitiveness in real-world preferences.

On the other hand, supervised (sequential-classification) pair preference models (PairPM) (Jiang et al., 2023; Dong et al., 2024) that predict the preference given a concatenation of the two responses can express complex and intran-



(a) Bradley-Terry (BT) reward model (b) PairRM / PairPM



(c) General Preference representation model (GPM)

Figure 2: Illustration of (a) Bradley Terry (BT) reward model, (b) supervised pair preference model (PairRM, PairPM) (Jiang et al., 2023; Dong et al., 2024), and (c) our General Preference representation model (GPM).

sitive (cyclic) structures. But to fully capture the preference relations among K responses, it requires evaluating $\mathcal{O}(K^2)$ pairwise preferences between all K candidate responses (Munos et al., 2023; Wu et al., 2024b). This quadratic scaling hinders them for applications with larger response sets especially in test-time scaling for reasoning tasks using verifiers and ranking models (Snell et al., 2024; Wu et al., 2024a).

Aside from computational inefficiency, supervised preference models also exhibit asymmetric preference behaviors related to positions. Also, the model's design choice can be highly ad-hoc, varying among different templates and different linear heads.

Based on the above observations, it is thus natural to raise the following question:

Is there a principled way to model general preference?

In this paper, we answer this question affirmatively by proposing *preference representation learning*, which bridges the gap between expressiveness and efficiency in general preference modeling. Our method embeds responses into a multi-dimensional latent space that captures the complex preference structure beyond transitive relations while allowing for efficient querying of preferences. Notably, our approach achieves a computational complexity of $\mathcal{O}(K)$, matching the efficiency of the BT model but with enhanced expressiveness.

The main contributions of our work are summarized as follows:

- We introduce preference representation learning for general preference modeling, enabling both efficient and expressive representation of human preferences. Our approach generalizes the Bradley-Terry (BT) reward model by embedding responses into a latent space, capturing complex structures, including intransitive preferences. Notably, our General Preference representation model (GPM) achieves a query complexity of $\mathcal{O}(K)$ for evaluating preferences among K responses, a significant improvement over the $\mathcal{O}(K^2)$ complexity of traditional supervised preference models that rely on pairwise inputs (see Section 4).
- We demonstrate GPM's effectiveness across various tasks, including CyclicPreference (ours) and the renowned RewardBench (Lambert et al., 2024). Specifically, GPM models intransitive (e.g., cyclic) preferences with 100% accuracy, whereas the BT reward model performs like random

guessing (see Section 6.1). Additionally, GPM outperforms the BT model on RewardBench with performance margins of up to 8.3% (see Section 6.2).

- For language model alignment, we propose General Preference Optimization (GPO), which leverages the preference scores provided by GPM. The general preference score can also be integrated as a preference signal into a wide range of RLHF and preference optimization methods, such as (iterative) DPO (Rafailov et al., 2024), SPPO (Wu et al., 2024b), and PPO-based methods (Ouyang et al., 2022). Experimental results on AlpacaEval-2.0 reveal that our approach may improve reward-based language model alignment methods (see Section 6.3).

2 RELATED WORK

Reward-Based Reinforcement Learning from Human Feedback (RLHF). The earlier approaches to modeling human preference for language model alignment usually learn a *reward model* from a preference dataset. The human preference is assumed to follow the Bradley-Terry (BT) model (Bradley & Terry, 1952) or the Thurstone model (Thurstone, 2017). LLM policies then are fine-tuned to maximize these scalar reward signals for better alignment (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022). Later, the direct preference optimization (DPO) methods are proposed by Rafailov et al. (2024) to only implicitly learn a reward model represented by an LLM. The human preference is still assumed to follow the Bradley-Terry model. However, the reliance on scalar rewards imposes a total ordering on preferences, which may not reflect the intransitive or stochastic nature of human judgments (Tversky, 1969; Agranov & Ortoleva, 2017).

Preference-Based Reinforcement Learning from Human Feedback. Recently, there emerged a line of works that directly estimates the preference probability without imposing a reward-based preference model or any transitivity assumptions (Lou et al., 2022; Wu et al., 2023; Wang et al., 2023) either for preference-based RL or in the context of RLHF. Efforts have been made to optimize policies directly from pair-wise preference comparisons, thereby mitigating the limitations of scalar reward functions (Munos et al., 2023; Swamy et al., 2024; Rosset et al., 2024; Wu et al., 2024b).

3 BACKGROUND

In this section, we present preliminaries on reward modeling, preference modeling, and reinforcement learning from human feedback (RLHF) for language model alignment. We consider an autoregressive language model that generates responses to the given prompts. Let $\mathbf{x} = [x_1, x_2, \ldots]$ denote a prompt, a sequence of tokens. The language model π generates a response $\mathbf{y} = [y_1, y_2, \ldots, y_N]$ based on the conditional probability distribution: $\pi(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{N} \pi\left(y_i \mid \mathbf{x}, \mathbf{y}_{<i}\right)$, where $\mathbf{y}_{<i}$ represents the sequence of tokens generated before position i. In this paper, we assume a general-preference oracle. Given two responses $\mathbf{y}$ and $\mathbf{y}'$ to the same prompt $\mathbf{x}$, the oracle provides the feedback indicating which response is preferred.

$$\mathbb{P}\left(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}\right) := \mathbb{E}\left[o\left(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}\right)\right].$$

3.1 REWARD-BASED REINFORCEMENT LEARNING FROM HUMAN FEEDBACK

The most prevalent approach to aligning language models with human preferences is to consider a scalar reward function $r(\mathbf{y}; \mathbf{x})$ that assigns a numerical score to each response. The preference between two responses is then determined solely by the reward scores for the two responses. For example, the Bradley-Terry (BT) model (Bradley & Terry, 1952) is a widely used method for modeling pairwise preferences in this context. However, the BT model can not capture intransitive (e.g. cyclic) preferences effectively (Bertrand et al., 2023). Under the BT model, the probability that response $\mathbf{y}$ is preferred over $\mathbf{y}'$ is given by:

$$\mathbb{P}(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}) = \sigma\big(r(\mathbf{y}; \mathbf{x}) - r(\mathbf{y}'; \mathbf{x})\big),$$

where $\sigma(z) = 1/(1 + e^{-z})$ is the logistic (sigmoid) function.

In practice, the reward function $r(\mathbf{y}; \mathbf{x})$ is learned by maximizing the likelihood of the observed preference data. Once the reward function is established, policy optimization techniques, such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), can be applied to adjust the language model to generate responses that maximize expected rewards. The optimization problem can be formulated as:

$$\max_{\theta} \mathbb{E}_{\mathbf{x}\sim\mathcal{X},\, \mathbf{y}\sim\pi_\theta(\cdot\mid\mathbf{x})}\left[r(\mathbf{y}; \mathbf{x})\right] - \beta\mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[\mathrm{KL}\left(\pi_\theta(\cdot \mid \mathbf{x}) \,\|\, \pi_{\text{ref}}(\cdot \mid \mathbf{x})\right)\right], \tag{1}$$

where θ are the parameters of the policy π_θ, π_{ref} is a reference policy (often the pre-trained or supervised-fine-tuned language model), β is a scaling parameter that controls the strength of regularization, and KL denotes the Kullback-Leibler divergence.

3.2 GENERAL PREFERENCE MODELING

We consider the scenario where given a prompt $\mathbf{x}$, a set of responses $\{\mathbf{y}_i\}$ is generated, and human preferences over these responses are represented as pairwise probabilities $\mathbb{P}(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) \in (0, 1)$, indicating the likelihood that response $\mathbf{y}_i$ is preferred over $\mathbf{y}_j$ given the prompt $\mathbf{x}$.

To model these preferences, we define a (pairwise) preference score function:

$$s(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) := \log \frac{\mathbb{P}(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x})}{1 - \mathbb{P}(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x})}, \tag{2}$$

which represents the log-odds of $\mathbf{y}_i$ being preferred over $\mathbf{y}_j$. This score function allows us to express the preference probability as:

$$\mathbb{P}(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) = \sigma\left(s(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x})\right), \tag{3}$$

where $\sigma(z) = 1/(1 + e^{-z})$ is the logistic function. One can see that the BT model is a special case: $s(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) = r(\mathbf{y}_i; \mathbf{x}) - r(\mathbf{y}_j; \mathbf{x})$.

3.2.1 SUPERVISED PAIR PREFERENCE MODELS

Existing approaches often involve concatenating the prompt and responses with a template and training an LLM-based sequential classifier in a supervised learning manner. For example, Jiang et al. (2023) simply concatenate the three segments $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ sequentially and form a single input sequence with special tokens as separators:

```
‘<s> <source> x </s> <candidate1> y1 </s> <candidate2> y2 </s>’
```

Then a sequential classification head on the last token is trained to predict the preference. Another example is Munos et al. (2023), which uses the following template for text summarization:

```
‘You are an expert summary rater. Given a piece of text and two of
its possible summaries, output 1 or 2 to indicate which summary
is better.
Text - ⟨text⟩, Summary 1 - ⟨summary1⟩, Summary 2 - ⟨summary2⟩.
Preferred Summary -’
```

Then use the last logit for an arbitrarily chosen token as $s(\mathbf{y}_1 \succ \mathbf{y}_2 | \mathbf{x})$ for training.

However, due to the language model's position encoding (Press et al., 2021; Su et al., 2024) and the causal attention (Radford et al., 2018; 2019) mechanism not being symmetric, the candidate's order in the concatenation will affect the final prediction results. It is mitigated by randomly shuffling the two responses in the training dataset but the output is still highly asymmetric. Another limitation is that how to represent the preference score can be highly ad-hoc. The two examples above already use different templates and different linear heads (sequential classification v.s. language modeling).

3.3 PREFERENCE-BASED REINFORCEMENT LEARNING FROM HUMAN FEEDBACK

To address the potential intransitive human preference, the preference-based LLM alignment algorithms (Munos et al., 2023; Azar et al., 2023; Wu et al., 2024b; Rosset et al., 2024) have been proposed to directly work on the preference pairs instead of assuming a reward function.

Given a preference oracle $\mathbb{P}\left(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}\right)$. The objective is to find a policy π that performs well against another competing policy π' in terms of these preference probabilities. For example, Azar et al. (2023) consider competing with another fixed policy μ ($\mathcal{X}$ denotes the distribution over prompts):

$$\max_{\pi} \mathbb{E}_{\mathbf{x}\sim\mathcal{X}} \left[\mathbb{E}_{\mathbf{y}\sim\pi(\cdot|\mathbf{x}),\, \mathbf{y}'\sim\mu(\cdot|\mathbf{x})} \left[\mathbb{P}\left(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}\right)\right] - \beta \mathrm{KL}(\pi \| \pi_{\text{ref}})\right], \tag{4}$$

Other works (Munos et al., 2023; Wu et al., 2024b; Rosset et al., 2024) consider solving the two-player constant-sum game:

$$\max_{\pi} \min_{\pi'} \mathbb{E}_{\mathbf{x}\sim\mathcal{X}} \left[\mathbb{E}_{\mathbf{y}\sim\pi(\cdot|\mathbf{x}),\, \mathbf{y}'\sim\pi'(\cdot|\mathbf{x})} \left[\mathbb{P}\left(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}\right)\right]\right]. \tag{5}$$

To simplify notation, we define the winning probability of a policy π over another policy π' as:

$$\mathbb{P}\left(\pi \succ \pi' \mid \mathbf{x}\right) = \mathbb{E}_{\mathbf{y}\sim\pi(\cdot|\mathbf{x}),\, \mathbf{y}'\sim\pi'(\cdot|\mathbf{x})} \left[\mathbb{P}\left(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}\right)\right]. \tag{6}$$

The optimization problem then becomes:

$$\max_{\pi} \min_{\pi'} \mathbb{E}_{\mathbf{x}\sim\mathcal{X}} \left[\mathbb{P}\left(\pi \succ \pi' \mid \mathbf{x}\right)\right]. \tag{7}$$

4 GENERAL PREFERENCE MODELING WITH PREFERENCE REPRESENTATIONS

In this section, we propose a general preference representation learning framework that can model human preferences efficiently and expressively. Each response is embedded as a vector in a latent space, and the preferences are modeled through interactions between these representations (embeddings) using a skew-symmetric operator. We first define preference representations, which serve as the foundation for modeling the relationships between responses.

Definition 4.1 (Preference Representations)**.** Given a prompt $\mathbf{x}$, we assign to each response $\mathbf{y}$ a preference representation vector $\mathbf{v}_{\mathbf{y}|\mathbf{x}} \in \mathbb{R}^{2k}$. These representations are designed to capture the features relevant to human preferences beyond what can be represented by scalar rewards.

Next, to model the directional nature of preferences, we introduce the skew-symmetric preference operator, which ensures that the model respects the skew-symmetry (anti-symmetry) in preference modeling.

Definition 4.2 (Skew-symmetric Preference Operator)**.** To capture the directional nature of preferences, we define a skew-symmetric (anti-symmetric) preference operator $\mathbf{R}^{\succ} \in \mathbb{R}^{2k\times 2k}$. Specifically, $\mathbf{R}^{\succ}$ is a block-diagonal matrix consisting of k skew-symmetric blocks of the form (for more discussion, please see Appendix A):

$$\mathbf{R}_l = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad l = 1, \ldots, k. \tag{8}$$

An example of $\mathbf{R}^{\succ}$ for $k = 2$ is:

$$\mathbf{R}^{\succ} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Finally, we define the preference score, which quantifies the degree to which one response is preferred over another. This score is calculated based on the interaction between the preference representations, mediated by the skew-symmetric operator.

Definition 4.3 (Preference Score)**.** The preference score between two responses $\mathbf{y}_i$ and $\mathbf{y}_j$ using preference representations is defined as:

$$s(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) = \left\langle \mathbf{R}^{\succ}\mathbf{v}_{\mathbf{y}_i|\mathbf{x}}, \mathbf{v}_{\mathbf{y}_j|\mathbf{x}} \right\rangle, \tag{9}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbb{R}^{2k}$. This score captures the anti-symmetric relationship between responses induced by human preferences.

We model the preference probability using the logistic function as defined in Equation (3). Our general preference representation model (GPM) exhibits two desirable properties:

1. **Skew-symmetry.** The preference score function is skew-symmetric, satisfying:

   $$s(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) = -s(\mathbf{y}_j \succ \mathbf{y}_i \mid \mathbf{x}).$$

   This reflects the fact that the preference relation is naturally skew-symmetric: if $\mathbf{y}_i$ is preferred over $\mathbf{y}_j$ with probability $p_{i,j}$, then $\mathbf{y}_j$ is preferred over $\mathbf{y}_i$ with probability $1 - p_{i,j}$. Specifically,

   $$s(\mathbf{y} \succ \mathbf{y} \mid \mathbf{x}) = \left\langle \mathbf{R}^{\succ}\mathbf{v}_{\mathbf{y}|\mathbf{x}}, \mathbf{v}_{\mathbf{y}|\mathbf{x}} \right\rangle = 0.$$

   This means that a response is neither superior nor inferior to itself.
2. **Magnitude preserving.** The skew-symmetric preference operator does not change the representation vector's magnitude, which makes this operation stable for training and inference.

   $$\left\langle \mathbf{R}^{\succ}\mathbf{v}_{\mathbf{y}|\mathbf{x}}, \mathbf{R}^{\succ}\mathbf{v}_{\mathbf{y}|\mathbf{x}} \right\rangle = \left\langle \mathbf{v}_{\mathbf{y}|\mathbf{x}}, \mathbf{v}_{\mathbf{y}|\mathbf{x}} \right\rangle.$$

Relation to Bradley-Terry Model. If we set $k = 1$, $\mathbf{v_y} = [r(\mathbf{y} \mid \mathbf{x}), c]^\top$, where c is a constant and $c \neq 0$ (e.g., $c = 1$), and $\mathbf{R}^{\succ} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, then the preference score reduces to:

$$s(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) = c\big(r(\mathbf{y}_i \mid \mathbf{x}) - r(\mathbf{y}_j \mid \mathbf{x})\big),$$

and the preference probability becomes:

$$\mathbb{P}(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) = \sigma\big[c\big(r(\mathbf{y}_i \mid \mathbf{x}) - r(\mathbf{y}_j \mid \mathbf{x})\big)\big],$$

which is exactly the Bradley-Terry (BT) model as a disk game (Balduzzi et al., 2019).

4.1 EXPRESSIVENESS OF THE MODEL

Our general preference representation model is fully expressive for any real skew-symmetric preference matrix (see Appendix A.1 for complex representations interpretation). Specifically, we establish the following theorem (similar results have been proved in Balduzzi et al. (2018)):

Theorem 4.4 (Expressiveness of Preference Representation Model)**.** *Let* $\mathbf{P} \in \mathbb{R}^{k\times k}$ *be a real skew-symmetric matrix (i.e.,* $\mathbf{P} = -\mathbf{P}^\top$*). Then there exist vectors* $\{\mathbf{v}_i\}_{i=1}^{k} \subset \mathbb{R}^{2k}$ *and a block-diagonal skew-symmetric matrix* $\mathbf{R}^{\succ} \in \mathbb{R}^{2k\times 2k}$*, with* $\mathbf{R}^{\succ}$ *consisting of* k *blocks of the form:*

$$\mathbf{R}_l = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad l = 1, \ldots, k,$$

such that:

$$P_{ij} = \mathbf{v}_i^\top \mathbf{R}^{\succ} \mathbf{v}_j, \quad \forall\, i, j.$$

Moreover, the vectors $\{\mathbf{v}_i\}$ *can be constructed explicitly from* $\mathbf{P}$*.*

Theorem 4.4 suggests that our preference representation framework can theoretically model arbitrary complex and potentially intransitive (e.g., cyclic) preference structures (see Appendix A.3 for proofs).

4.2 IMPLEMENTING GENERAL PREFERENCE REPRESENTATION MODEL

When the preference score matrix $\mathbf{P}$ has an even dimension, i.e., $\mathbf{P} \in \mathbb{R}^{2k\times 2k}$, we have a more interesting interpretation based on spectral decomposition.

Theorem 4.5 (Expressiveness of Preference Representation Model)**.** *Let* $\mathbf{P} \in \mathbb{R}^{2k\times 2k}$ *be a real skew-symmetric matrix (i.e.,* $\mathbf{P} = -\mathbf{P}^\top$*). Then there exist representations (embeddings)* $\{\mathbf{v}_i\}_{i=1}^{2k} \subset \mathbb{R}^{2k}$ *and a block-diagonal skew-symmetric matrix* $\mathbf{R}^{\succ} \in \mathbb{R}^{2k\times 2k}$*, such that:*

$$P_{ij} = \mathbf{v}_i^\top \mathbf{R}^{\succ} \mathbf{v}_j, \quad \forall\, i, j.$$

Moreover, the representations $\{\mathbf{v}_i\}$ *can be constructed from the orthogonal matrix* $\mathbf{U}$ *in the decomposition of* $\mathbf{P}$*, scaled by the square roots of the positive eigenvalues of* $\mathbf{P}$*.*

To effectively capture general preferences while maintaining computational efficiency, we implement our preference representation model by augmenting an existing language model with two additional components: an eigenvalue scale gate and an eigenvector embedding head.

Eigenvalue Scale Gate. The eigenvalue scale gate $\mathcal{G}_\lambda$ computes context-dependent scaling factors $\{\lambda_l(\mathbf{x})\}$, where $\lambda_l(\mathbf{x}) \geq 0$, based solely on the prompt $\mathbf{x}$:

$$\{\lambda_l(\mathbf{x})\} = \mathcal{G}_\lambda(\mathbf{x}).$$

This component models how different preference dimensions are weighted in the context of the given prompt, effectively adjusting the importance of various aspects such as helpfulness, instruction-following, and creativity.

Eigenvector Embedding Head. The eigenvector embedding head $\mathcal{E}_\mathbf{v}$ generates embeddings $\mathbf{v}_{\mathbf{y}|\mathbf{x}}$ for each response $\mathbf{y}$ in the context of the prompt $\mathbf{x}$:

$$\mathbf{v}_{\mathbf{y}|\mathbf{x}} = \mathcal{E}_\mathbf{v}(\mathbf{x}, \mathbf{y}).$$

These embeddings capture the nuanced characteristics of the responses relevant to human preferences.

Preference Score. The preference score between two responses is computed as:

$$s(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) = \mathbf{v}_{\mathbf{y}_i|\mathbf{x}}^{\top} \mathbf{D}(\mathbf{x}) \mathbf{R}^{\succ} \mathbf{D}(\mathbf{x}) \mathbf{v}_{\mathbf{y}_j|\mathbf{x}}.$$

where $\mathbf{D}(\mathbf{x})$ is a block-diagonal matrix with blocks $\sqrt{\lambda_l(\mathbf{x})}\mathbf{I}_2$, and $\mathbf{R}^{\succ}$ is the skew-symmetric operator defined previously. We normalize the embeddings $\mathbf{v}_{\mathbf{y}}$ to have unit length to ensure stability in training.

Automatic Subspace Discovery. The use of multiple dimensions in the embeddings allows the model to discover different subspaces corresponding to various preference dimensions automatically. Each pair of dimensions can capture distinct aspects of preferences, such as helpfulness, correctness, or stylistic elements. The context-dependent eigenvalues $\lambda_l(\mathbf{x})$ modulate the contributions of these subspaces based on the prompt, enabling the model to adapt to varying user preferences dynamically.

We have conducted ablation studies on the architecture of the general preference representation model—specifically, evaluating the inclusion of the eigenvalue scale gate and L2 normalization in the eigenvector embedding head. These results are detailed in Table 7 of Appendix B.1.

5 EFFICIENT PREFERENCE OPTIMIZATION WITH GENERAL PREFERENCE

The previous general preference models require $\mathcal{O}(K^2)$ inference-time compute to evaluate all pairwise preferences among K responses (Munos et al., 2023; Swamy et al., 2024). In contrast, computing the preference representation for K responses requires only $O(K)$ forward passes: we first calculate the representation $\mathbf{v}_i$ for each $\mathbf{y}_i$, and then use them to calculate the preference probability between any two responses using formula $s(\mathbf{y}_i \succ \mathbf{y}_j) = \langle \mathbf{R}^{\succ}\mathbf{v}_i, \mathbf{v}_j \rangle$. In this way, our model is as efficient as a (Bradley-Terry) reward model while being way more expressive.

Policy Optimization with Preference Score. Once we have a general preference model that outputs the preference score $s(\mathbf{y}_i \succ \mathbf{y}_j|\mathbf{x})$ at hand, we aim to find a policy π that performs well against an opponent policy μ in terms of expected preference scores. The optimization problem is formulated as:

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}} \left[\mathbb{E}_{\mathbf{y} \sim \pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x}),\, \mathbf{y}' \sim \mu(\cdot|\mathbf{x})} \left[s(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}) \right] \right] - \beta \mathbb{E}_{\mathbf{x}} \left[\mathrm{KL} \left(\pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{x}) \| \pi_{\text{ref}}(\cdot \mid \mathbf{x}) \right) \right], \tag{10}$$

where π_{ref} is a reference policy (e.g., the initial language model), μ is the opponent policy (usually the same as π_{ref}), and $\beta > 0$ is a regularization parameter controlling the divergence from the reference policy. We would like to point out that this formulation is different from the many previous works (Wu et al., 2024b; Swamy et al., 2024; Rosset et al., 2024; Munos et al., 2023; Azar et al., 2023) as they consider maximizing the win rate $\mathbb{P}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$, while our formulation is to maximize $s(\mathbf{y} \succ \mathbf{y}'|\mathbf{x}) = \log \frac{\mathbb{P}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})}{\mathbb{P}(\mathbf{y} \prec \mathbf{y}'|\mathbf{x})}$. Note that $\mathbb{P}(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$ only varies between 0 and 1, while $s(\mathbf{y} \succ \mathbf{y}'|\mathbf{x})$, similar to the reward $r(\mathbf{y}; \mathbf{x})$ in RLHF or DPO, can take arbitrary values. The flexibility in its value range might benefit fine-tuning.

General Preference Optimization (GPO). We consider the SPPO loss used by Wu et al. (2024b) for iterative preference optimization, except that we use preference score instead of preference probability in the loss form. SPPO used K responses for each prompt $\mathbf{x}$ and calculated the empirical win rate of each response $\mathbf{y}_k$. Instead, we calculate $\widehat{s}\,(\mathbf{y}_i \succ \mu \mid \mathbf{x})$ to estimate the empirical win rate over the distribution μ as below:

$$\widehat{s}\,(\mathbf{y}_i \succ \mu \mid \mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} s\left(\mathbf{y}_i \succ \mathbf{y}_k \mid \mathbf{x}\right), \forall i \in [K], \tag{11}$$

At each iteration t, GPO has the following learning objective:

$$\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \mathbf{y} \sim \pi_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})} \left[\left(\log \left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x})}{\pi_{\boldsymbol{\theta}_t}(\mathbf{y} \mid \mathbf{x})} \right) - \frac{1}{\beta} \left(\widehat{s}\,(\mathbf{y} \succ \pi_{\boldsymbol{\theta}_t} \mid \mathbf{x}) - \log Z_{\pi_{\boldsymbol{\theta}_t}}(\mathbf{x}) \right) \right)^2 \right], \tag{12}$$

where the normalizing factor $Z_{\pi_{\boldsymbol{\theta}_t}}(\mathbf{x}) := \sum_{\mathbf{y}} \pi_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) \exp\left(\widehat{s}\,(\mathbf{y} \succ \pi_{\boldsymbol{\theta}_t} \mid \mathbf{x})\right)$.

In practice, we directly replace $\log Z_{\pi_{\boldsymbol{\theta}_t}}(\mathbf{x})$ with 0[1]. Intuitively, if a response $\mathbf{y}$ receives a high average score, GPO will increase its log probability. We report the empirical performance of GPO in Section 6.3 (we present convergence analysis of GPO in Appendix C).

Remark 5.1. Notice that the GPO learning objective can be seen as an offline policy gradient algorithm (see Appendix C) for the optimization problem defined in Equation (12), similar results have been discussed in Munos et al. (2023); Wu et al. (2024b).

Remark 5.2. Note that the general preference score given by our GPM in Equation (10) can also be integrated as preference (reward) signal for any off-the-shelf RLHF and preference optimization methods, including (iterative) DPO (Rafailov et al., 2024), IPO (Azar et al., 2023), NLHF (Munos et al., 2023), SPPO (Wu et al., 2024b) and REBEL (Gao et al., 2024), as well as PPO-based methods (Ouyang et al., 2022) by directly optimizing Equation (10).

6 EXPERIMENTS

We conducted several experiments to evaluate the effectiveness of the proposed General Preference representation model (GPM) in comparison to traditional reward-based models, particularly focusing on its ability to model cyclic preferences and improve language model alignment. Our experiments are designed to address the following questions:

- **Q1:** Can the GPM effectively capture and model cyclic and intransitive preferences, where traditional models like the Bradley-Terry (BT) reward model struggle?
- **Q2:** How does the GPM perform on standard preference modeling benchmarks (RewardBench) compared to the BT model?
- **Q3:** How does using the GPM for downstream policy optimization impact language model performance on real-world tasks such as AlpacaEval compared to reward-based approaches?

6.1 CYCLIC PREFERENCE MODELING

To address **Q1**, we evaluate the ability of the GPM to capture intransitive, cyclic preferences that traditional transitive models (like the BT model) struggle to represent.

Cyclic Preference Dataset. We constructed a dataset by inducing cyclic preferences from the Ultrafeedback dataset Cui et al. (2024). The dataset includes responses evaluated across four key metrics: *instruction following*, *honesty*, *truthfulness*, and *helpfulness*. We created preference cycles such as: `instruction following` $\succ$ `honesty` $\succ$ `truthfulness` $\succ$ `helpfulness` $\succ$ `instruction following`, ensuring the presence of intransitive cycles. We further generated four sub-datasets by omitting one metric from each cycle, resulting in datasets of varying complexity with 216 to 363 instances.

Training and Evaluation. We trained the GPM using the **Gemma-2B-it** language model as the base and evaluated the models based on their ability to predict the human-provided preferences in these datasets. For the Bradley-Terry (BT) model, the loss function is $\mathcal{L} = -\log \sigma(r_w - r_l)$ (Ouyang et al., 2022). Since cyclic preferences are inherently intransitive, we measure accuracy as the percentage of correctly predicted human preferences, where higher scores indicate better handling of non-transitive preferences. As shown in Table 1, the GP representation model achieves near-perfect accuracy across all datasets, significantly outperforming the BT model (we report the test accuracy on the training dataset but with different comparison pairs used in the training dataset). These results validate the GP representation model's ability to capture complex, cyclic preferences, confirming the theoretical advantages of using a preference representation-based approach over traditional reward models that assume transitivity (more on implementation details are presented in Appendix B.2).

6.2 EXPERIMENTS ON REWARDBENCH

To address **Q2**, we compare the GP representation model and the BT reward model on the RewardBench benchmark (Lambert et al., 2024), which covers diverse preference modeling tasks, including Chat, Chat-Hard, Safety, and Reasoning.

---

[1] In late stages of the iterative training, $\pi_{\boldsymbol{\theta}_t}$ is close to equilibrium so the preference model can not distinguish between policy $\pi_{\boldsymbol{\theta}}$ and the opponent policy $\pi_{\boldsymbol{\theta}_t}$(meaning $\widehat{s}(\mathbf{y} \succ \pi_{\boldsymbol{\theta}_t} \mid \mathbf{x}) \approx 0$). Therefore, we have $\log Z_{\pi_{\boldsymbol{\theta}_t}}(\mathbf{x}) \approx 0$.

Table 1: Comparison of Bradley-Terry (BT) reward model and General Preference representation models (GPM) on cyclic preference datasets.

Model	Dataset	Acc. (%)
Random Guess		50.0
BT RM	w.o. `instruction following`	62.4
GPM	w.o. `instruction following`	**100.0 (+37.6)**
BT RM	w.o. `honesty`	61.6
GPM	w.o. `honesty`	**100.0 (+38.4)**
BT RM	w.o. `truthfulness`	50.0
GPM	w.o. `truthfulness`	**100.0 (+50.0)**
BT RM	w.o. `helpfulness`	62.9
GPM	w.o. `helpfulness`	**100.0 (+37.1)**

Datasets and Experimental Setup. We train both the BT and GPMs using the decontaminated version of Skywork Reward Data Collection (Liu & Zeng, 2024), which contains around 80 thousand pairwise preference examples from tasks in various domains. We evaluate both models on RewardBench, using two different base models: **Gemma-2B-it** (Team et al., 2024) (2B parameters) and **Llama-3.1-8B-Instruct** (Dubey et al., 2024) (8B parameters), which are well-suited for instruction-following tasks (please refer to Appendix B.2 for the implementation details).

Results and Analysis. Table 2 presents the results. The GPM consistently outperforms the BT model for both base models on RewardBench, with notable improvements in tasks involving complex reasoning (e.g., Chat-Hard and Reasoning). These results highlight the superior expressiveness of the GPM in preference modeling.

Table 2: Comparison between the Bradley-Terry (BT) models and the General Preference representation models (GPM) with varying embedding head dimensions on RewardBench. The highest scores are in bold and the second highest are underlined.

Model	Embed Dim.	Chat	Chat-Hard	Safety	Reasoning	Average
Base Model: Gemma-2B-it						
BT RM	1	71.51	64.69	75.00	61.90	68.27
GPM	2	**78.49**	65.35	78.92	72.64	73.85
	4	76.54	64.91	78.51	79.80	74.94
	6	76.82	64.04	73.24	77.02	72.78
	8	**78.49**	**66.23**	**84.32**	**80.47**	**77.38 (+9.11)**
Base Model: Llama-3.1-8B-Instruct						
BT RM	1	88.55	85.75	91.49	**96.47**	90.56
GPM	2	**93.30**	86.40	91.22	94.01	91.23
	4	**93.30**	86.18	91.22	95.69	**91.60 (+1.04)**
	6	91.90	86.40	90.95	94.06	90.83
	8	92.18	**87.06**	**91.76**	94.49	91.37
Base Model: Gemma-2-9B-it						
BT RM	1	91.62	85.96	**92.70**	95.55	91.46
GPM	2	92.46	85.96	92.30	94.56	91.32
	4	**93.58**	**87.72**	92.30	95.71	**92.33 (+0.87)**
	6	92.46	86.18	92.43	95.67	91.69
	8	91.62	85.96	92.43	**95.89**	91.48
Other state-of-the-art models						
GPT-4	-	95.3	74.3	87.6	86.9	86.0
GPT-4o	-	96.1	76.1	88.1	86.6	86.7
Gemini-1.5	-	92.3	80.6	87.9	92.0	88.2
RLHFlow/pair-pm-8B	1	92.3	80.6	89.7	94.7	87.1
ArmoRM-8B	5	98.3	65.8	90.5	97.3	90.4
Nemotron-4-340B	5	95.8	87.1	91.5	93.6	92.0

Ablation Studies. We further conducted ablation studies to assess the impact of varying the representation (embedding) dimension in the GPM. Table 2 shows that increasing the embedding dimension generally improves performance.

6.3 DOWNSTREAM PERFORMANCE ON ALIGNING LANGUAGE MODELS WITH HUMAN PREFERENCES

To address **Q3**, we investigate the effectiveness of the GPM in language model for alignment using Self-Play Policy Optimization (SPPO) (Wu et al., 2024b) and our proposed General Preference Optimization (GPO), integrating preference scores provided by our GP representation model (GPM). We evaluated the models on AlpacaEval 2.0 (Dubois et al., 2024), MT-Bench (Zheng et al., 2023), GSM8K, MMLU, etc., several widely used benchmarks for evaluating LLM alignment.

Results and Analysis. The evaluation results on the benchmarks are as follows. For AlpacaEval 2.0, we compared the generated responses of the aligned models with those of GPT-4-turbo. To avoid the preference bias when using GPT-4-turbo as the evaluator, we also used DeepSeek-V2 (DeepSeek-AI, 2024) and GPT-4o-mini as the evaluators besides GPT-4-turbo itself. Notice that the Length Controlled (LC) Win Rate results are using a generalized linear model fitted using default evaluator GPT-4-turbo, so it does not apply to other evaluators. The results of the three evaluators are presented in Tables 3,4 and 5. From Table 3, we observe that both SPPO and GPO demonstrate improved win rates with successive iterations, highlighting the iterative nature of these optimization methods, and GPO consistently outperforms SPPO. In addition, the bolded entries indicate that GPM-integrated SPPO/GPO consistently outperforms the BT RM-based SPPO/GPO under the same settings, underscoring the superior expressiveness and flexibility of the GPM in modeling human preferences (for additional experimental results on MT-Bench, GSM8K, MMLU, etc., please see Appendix B.1).

Table 3: AlpacaEval 2.0 evaluation results. Base model: LLama3-8B-it, Evaluator: GPT-4-turbo. The results are grouped by the size and type of the RM or PM, and the number of iterations. Bold entries indicate that the GPM outperforms the corresponding BT RM under the same training settings.

Size	Type	Iter	SPPO LC. WR	SPPO WR	SPPO Avg. Len	GPO LC. WR	GPO WR	GPO Avg. Len
		base	23.07	23.34	1959	23.07	23.34	1959
2B	**BT RM**	1	31.95	31.59	1939	34.01	33.08	1929
		2	36.00	36.77	2032	38.90	39.90	2049
		3	40.01	42.12	2136	42.21	44.20	2151
	GPM	1	30.87	**32.48 (+0.89)**	2066	**35.27**	**37.95 (+4.87)**	2102
		2	34.54	**40.76 (+3.99)**	2301	36.77	**42.96 (+3.06)**	2343
		3	36.06	**45.61 (+3.49)**	2498	37.74	**48.25 (+4.05)**	2582
8B	**BT RM**	1	32.20	27.83	1740	36.32	30.37	1702
		2	39.75	36.95	1868	41.79	40.11	1933
		3	42.55	40.92	1948	40.37	38.56	1969
	GPM	1	**33.48**	**30.85 (+3.02)**	1861	36.00	**33.19 (+2.82)**	1850
		2	37.93	**38.38 (+1.43)**	2029	40.81	**42.80 (+2.69)**	2115
		3	39.45	**41.64 (+0.72)**	2385	38.98	**41.54 (+2.98)**	3249

7 CONCLUSION

In this work, we introduce preference representation learning, a framework for modeling human preferences that can capture complex, intransitive structures like cyclic preferences. Our General Preference representation model (GPM) achieves linear complexity while maintaining the ability to model intricate preference relationships. It consistently outperforms traditional models like Bradley-Terry reward models across various benchmarks, including cyclic preference datasets and real-world tasks from RewardBench. Additionally, incorporating preference scores from GPM into policy optimization methods, such as SPPO and the newly introduced General Preference Optimization (GPO), led to significant performance improvements in downstream tasks that require alignment with intricate human preferences, as demonstrated in benchmarks like AlpacaEval 2.0 and MT-Bench.

Ethics Statement. This research introduces a new approach to modeling human preferences for aligning language models with nuanced human values. We utilized publicly available datasets such as the Ultrafeedback dataset, Skywork Reward Data Collection, AlpacaEval 2.0, and MT-Bench. These datasets comprise anonymized human-generated text and are used under their respective licenses. No personally identifiable information is included, and we did not collect any new data involving human subjects.

We recognize that enhancing language models' ability to align with human preferences can have both beneficial and unintended consequences. While we aim to improve the positive interactions between AI systems and users, there is a potential risk that such models could be misused to generate misleading or biased content. To mitigate this, we advocate for the responsible deployment of our methods and encourage further research into safeguarding against misuse.

Reproducibility Statement. We have taken several measures to ensure the reproducibility of our results. The architecture and implementation details of the General Preference representation model (GPM) and General Preference Optimization (GPO) are thoroughly described in Sections 4 and 5 of the main text and Appendix A. Hyperparameters, training procedures, and experimental setups are detailed in Section 6 and Appendix B.2.

All datasets used in our experiments are publicly accessible, with proper citations provided. We employed open-source language models, specifically Gemma-2B-it and Llama-3.1-8B-Instruct, to facilitate replication. Our source codes are included in the supplementary files submitted with this paper. This package contains all scripts and instructions necessary to reproduce the experiments and results presented in the paper.

REFERENCES

Marina Agranov and Pietro Ortoleva. Stochastic choice and preferences for randomization. *Journal of Political Economy*, 125(1):40–68, 2017.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. Re-evaluating evaluation. *Advances in Neural Information Processing Systems*, 31, 2018.

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In *International Conference on Machine Learning*, pp. 434–443. PMLR, 2019.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard (2023-2024). `https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard`, 2023.

Quentin Bertrand, Wojciech Marian Czarnecki, and Gauthier Gidel. On the limitations of the elo, real-world games are transitive, not additive. In *International Conference on Artificial Intelligence and Statistics*, pp. 2905–2921. PMLR, 2023.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607. PMLR, 2020.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled ai feedback, 2024. URL `https://arxiv.org/abs/2310.01377`.

Wojciech M Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. Real world games look like spinning tops. *Advances in Neural Information Processing Systems*, 33:17443–17454, 2020.

DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and et al. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

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

Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In *Conference on Learning Theory*, pp. 563–587. PMLR, 2015.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Zhaolin Gao, Jonathan D Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, J Andrew Bagnell, Jason D Lee, and Wen Sun. Rebel: Reinforcement learning via regressing relative rewards. *arXiv preprint arXiv:2404.16767*, 2024.

Martin Gardner. Mathematical games. *Scientific american*, 222(6):132–140, 1970.

Jian Hu, Xibin Wu, Weixun Wang, Xianyu, Dehao Zhang, and Yu Cao. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling. `https://huggingface.co/spaces/allenai/reward-bench`, 2024.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 5 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Chris Yuhao Liu and Liang Zeng. Skywork reward model series. `https://huggingface.co/Skywork`, September 2024. URL `https://huggingface.co/Skywork`.

Hao Lou, Tao Jin, Yue Wu, Pan Xu, Quanquan Gu, and Farzad Farnoud. Active ranking without strong stochastic transitivity. *Advances in neural information processing systems*, 35:297–309, 2022.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and Shayne Longpre. Octopack: Instruction tuning code large language models. *arXiv preprint arXiv:2308.07124*, 2023.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Guillermo Owen. *Game theory*. Emerald Group Publishing, 2013.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *openai.com*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models, 2024. URL `https://arxiv.org/abs/2308.01263`.

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

Amartya Sen. Social choice theory. *Handbook of mathematical economics*, 3:1073–1181, 1986.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. *arXiv preprint arXiv:2401.04056*, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, and et al. Gemma 2: Improving open language models at a practical size, 2024. URL `https://arxiv.org/abs/2408.00118`.

Louis L Thurstone. A law of comparative judgment. In *Scaling*, pp. 81–92. Routledge, 2017.

Amos Tversky. Intransitivity of preferences. *Psychological review*, 76(1):31, 1969.

Yuanhao Wang, Qinghua Liu, and Chi Jin. Is rlhf more difficult than standard rl? *arXiv preprint arXiv:2306.14111*, 2023.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-not-answer: Evaluating safeguards in LLMs. In Yvette Graham and Matthew Purver (eds.), *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 896–911, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.findings-eacl.61`.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024a.

Yue Wu, Tao Jin, Hao Lou, Farzad Farnoud, and Quanquan Gu. Borda regret minimization for generalized linear dueling bandits. *arXiv preprint arXiv:2303.08816*, 2023.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024b.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. In *International Conference on Learning Representations (ICLR)*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Appendix

A More on General Preference Representation Learning **16**
A.1 Complex Representations Interpretation 16
A.2 Training Objective 16
A.3 Appendix for Proofs 17

B More on Experiments **20**
B.1 Additional Experimental Results 20
B.2 Implementation Details 21

C More on General Preference Optimization **23**

D More Related Work **26**

A MORE ON GENERAL PREFERENCE REPRESENTATION LEARNING

In this section, we present additional discussion on general preference modeling with preference representations.

Proposition A.1. *For any two vectors* $\mathbf{v}_i \in \mathbb{R}^{2k}$ *and* $\mathbf{v}_j \in \mathbb{R}^{2k}$*, if* $\mathbf{R} \in \mathbb{R}^{2k \times 2k}$ *satisfies the following two properties:*

1. Skew-symmetry: $\langle \mathbf{R}\mathbf{v}_i, \mathbf{v}_j \rangle = -\langle \mathbf{R}\mathbf{v}_j, \mathbf{v}_i \rangle$.

2. Magnitude preserving: $\langle \mathbf{R}\mathbf{v}_i, \mathbf{R}\mathbf{v}_i \rangle = \langle \mathbf{v}_i, \mathbf{v}_i \rangle$.

Then $\mathbf{R}$ *must be in the form* $\mathbf{R} = \mathbf{U}\mathbf{J}\mathbf{U}^\top$*, where* $\mathbf{U} \in \mathbb{R}^{2k \times 2k}$ *is an orthonormal matrix (e.g. identity matrix* $\mathbf{I}_{2k}$*) and* $\mathbf{J}$ *is a block-diagonal matrix consisting of* k *skew-symmetric blocks of the form:*

$$\mathbf{J}_l = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad l = 1, \dots, k.$$

A.1 COMPLEX REPRESENTATIONS INTERPRETATION

Our model can also be interpreted using complex representations. By representing the representations as complex vectors $\mathbf{v}_\mathbf{y} \in \mathbb{C}^k$, we can express the preference score as:

$$s(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) = \operatorname{Im}\left(\langle \mathbf{v}_{\mathbf{y}_i}, \mathbf{v}_{\mathbf{y}_j} \rangle\right),$$

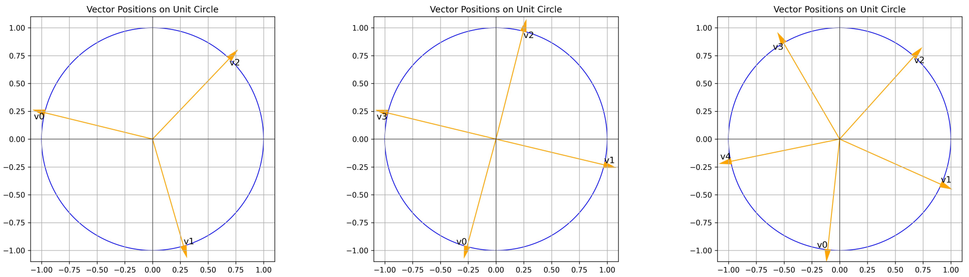
where $\operatorname{Im}(\cdot)$ denotes the imaginary part, and $\langle \cdot, \cdot \rangle$ is the Hermitian inner product. This formulation captures cyclic and intransitive preferences through the angular relationships between complex presentations.

Theorem A.2 (Expressiveness of Complex Preference Representations)**.** *Let* $\mathbf{P} \in \mathbb{R}^{k \times k}$ *be a real skew-symmetric matrix (i.e.,* $\mathbf{P} = -\mathbf{P}^\top$*). Then, there exist complex vectors* $\{\mathbf{v}_i\}_{i=1}^k \subset \mathbb{C}^k$ *such that:*

$$P_{ij} = \operatorname{Im}\left(\langle \mathbf{v}_i, \mathbf{v}_j \rangle\right), \quad \forall\, i, j.$$

Example. For $k = 1$, let $\mathbf{v}_\mathbf{y} = e^{i\theta_\mathbf{y}}$, then:

$$s(\mathbf{y}_i \succ \mathbf{y}_j \mid \mathbf{x}) = \sin(\theta_{\mathbf{y}_i} - \theta_{\mathbf{y}_j}).$$



(a) Cyclic 3 (b) Cyclic 4 (c) Cyclic 5

Figure 3: Visualization of learned preference embedding vectors for cyclic preferences with sizes 3, 4, and 5, e.g., $A \succ B \succ C \succ A$.

A.2 TRAINING OBJECTIVE

The preference embedding can thus be obtained by minimizing the cross-entropy loss over observed preference data. Given a dataset $(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l) \sim \mathcal{D}$ of preference comparisons, we denote $\mathbb{P}(\mathbf{y}_w \succ \mathbf{y}_l|\mathbf{x})$ as the probability of the winner $\mathbf{y}_w$ being chosen over the loser $\mathbf{y}_l$ (1 if hard preference is given). The cross-entropy loss function is:

$$\mathcal{L}_{\text{CE}} = -\sum_{(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l) \in \mathcal{D}} \Bigg[\mathbb{P}_\mathcal{D}(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x}) \log \sigma\left(\frac{1}{\beta} s(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x})\right) + (1 - \mathbb{P}_\mathcal{D}(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x})) \log \sigma\left(-\frac{1}{\beta} s(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x})\right) \Bigg].$$

Alternatively, if there is an oracle providing continuous scores, we can use a regression loss:

$$\mathcal{L}_{\text{MSE}} = \sum_{(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\in\mathcal{D}} \left(\frac{1}{\beta} s(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x}) - s_{\mathcal{D}}(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x})\right)^2,$$

where $s_{\mathcal{D}}(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x})$ is the dataset-provided score satisfying $\sigma\left(s_{\mathcal{D}}(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x})\right) = \mathbb{P}_{\mathcal{D}}(\mathbf{y}_w \succ \mathbf{y}_l \mid \mathbf{x})$.

A.3 Appendix for Proofs

Proof of the Proposition A.1.

Proof. Let $\mathbf{R} \in \mathbb{R}^{2k\times 2k}$ be a real matrix satisfying the following properties:

1. **Skew-symmetry with respect to the inner product:**

$$\langle \mathbf{R}\mathbf{v}, \mathbf{w}\rangle = -\langle \mathbf{R}\mathbf{w}, \mathbf{v}\rangle, \quad \forall\, \mathbf{v}, \mathbf{w} \in \mathbb{R}^{2k}.$$

2. **Magnitude preserving:**

$$\langle \mathbf{R}\mathbf{v}, \mathbf{R}\mathbf{v}\rangle = \langle \mathbf{v}, \mathbf{v}\rangle, \quad \forall\, \mathbf{v} \in \mathbb{R}^{2k}.$$

Recall that the standard inner product in $\mathbb{R}^{2k}$ is given by $\langle \mathbf{v}, \mathbf{w}\rangle = \mathbf{v}^\top \mathbf{w}$, which is symmetric: $\langle \mathbf{v}, \mathbf{w}\rangle = \langle \mathbf{w}, \mathbf{v}\rangle$.

From the skew-symmetry condition, we have:

$$\langle \mathbf{R}\mathbf{v}, \mathbf{w}\rangle + \langle \mathbf{R}\mathbf{w}, \mathbf{v}\rangle = 0, \quad \forall\, \mathbf{v}, \mathbf{w} \in \mathbb{R}^{2k}.$$

Since $\langle \mathbf{R}\mathbf{w}, \mathbf{v}\rangle = (\mathbf{R}\mathbf{w})^\top \mathbf{v} = \mathbf{w}^\top \mathbf{R}^\top \mathbf{v}$, the above condition becomes:

$$\mathbf{v}^\top \mathbf{R}^\top \mathbf{w} + \mathbf{w}^\top \mathbf{R}^\top \mathbf{v} = 0, \quad \forall\, \mathbf{v}, \mathbf{w} \in \mathbb{R}^{2k}.$$

This implies that $\mathbf{R}^\top$ is skew-symmetric:

$$\mathbf{R}^\top = -\mathbf{R}.$$

From the magnitude-preserving property, we have:

$$\langle \mathbf{R}\mathbf{v}, \mathbf{R}\mathbf{v}\rangle = (\mathbf{R}\mathbf{v})^\top \mathbf{R}\mathbf{v} = \mathbf{v}^\top \mathbf{R}^\top \mathbf{R}\mathbf{v} = \mathbf{v}^\top \mathbf{v}, \quad \forall\, \mathbf{v} \in \mathbb{R}^{2k}.$$

Therefore,

$$\mathbf{R}^\top \mathbf{R} = \mathbf{I}_{2k}.$$

Using $\mathbf{R}^\top = -\mathbf{R}$, we obtain:

$$(-\mathbf{R})\mathbf{R} = \mathbf{I}_{2k} \quad \Rightarrow \quad \mathbf{R}^2 = -\mathbf{I}_{2k}.$$

This shows that $\mathbf{R}$ satisfies the equation $\mathbf{R}^2 = -\mathbf{I}_{2k}$.

The characteristic polynomial of $\mathbf{R}$ is then:

$$\det(\mathbf{R} - \lambda \mathbf{I}_{2k}) = 0.$$

Since $\mathbf{R}^2 = -\mathbf{I}_{2k}$, it follows that the eigenvalues λ satisfy:

$$\lambda^2 = -1 \quad \Rightarrow \quad \lambda = \pm i.$$

Thus, $\mathbf{R}$ has eigenvalues $\pm i$, each with algebraic multiplicity k.

Because $\mathbf{R}$ is real and skew-symmetric, it can be brought into block-diagonal form via an orthogonal transformation. Specifically, there exists an orthogonal matrix $\mathbf{U} \in \mathbb{R}^{2k\times 2k}$ such that:

$$\mathbf{R} = \mathbf{U}\mathbf{J}\mathbf{U}^\top,$$

where

$$\mathbf{J} = \operatorname{blockdiag}(\mathbf{J}_1, \mathbf{J}_2, \ldots, \mathbf{J}_k),$$

and each block $\mathbf{J}_l$ is a 2×2 skew-symmetric matrix of the form:

$$\mathbf{J}_l = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad l = 1, \dots, k.$$

This decomposition leverages the standard canonical form for real skew-symmetric matrices, which states that any such matrix can be orthogonally diagonalized into blocks of this type.

Therefore, $\mathbf{R}$ can be expressed as:

$$\mathbf{R} = \mathbf{U}\mathbf{J}\mathbf{U}^\top,$$

where $\mathbf{U} \in \mathbb{R}^{2k \times 2k}$ is an orthogonal matrix, and $\mathbf{J}$ is the block-diagonal matrix consisting of k blocks $\mathbf{J}_l$.

This completes the proof. □

Proof of the Theorem 4.4.

Proof. We aim to represent the entries of the skew-symmetric matrix $\mathbf{P} \in \mathbb{R}^{k \times k}$ using vectors in $\mathbb{R}^{2k}$ and a block-diagonal skew-symmetric matrix $\mathbf{R}^{\succ} \in \mathbb{R}^{2k \times 2k}$.

For each $i = 1, \dots, k$, define the vector $\mathbf{v}_i \in \mathbb{R}^{2k}$ as:

$$\mathbf{v}_i = \begin{bmatrix} \mathbf{a}_i \\ \mathbf{b}_i \end{bmatrix},$$

where $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^k$ are real vectors to be specified.

Set $\mathbf{a}_i = \mathbf{e}_i$, the i-th standard basis vector in $\mathbb{R}^k$, and define $\mathbf{b}_i$ as:

$$\mathbf{b}_i = \frac{1}{2}\mathbf{p}_i,$$

where $\mathbf{p}_i$ is the i-th row of $\mathbf{P}$. Thus, the j-th component of $\mathbf{b}_i$ is $(\mathbf{b}_i)_j = \frac{1}{2}P_{ij}$.

Define the block-diagonal matrix $\mathbf{R}^{\succ} \in \mathbb{R}^{2k \times 2k}$ as:

$$\mathbf{R}^{\succ} = \text{blockdiag}(\mathbf{R}_1, \dots, \mathbf{R}_k),$$

where each block $\mathbf{R}_l$ is the 2×2 skew-symmetric matrix:

$$\mathbf{R}_l = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad l = 1, \dots, k.$$

Now, compute the inner product $\mathbf{v}_i^\top \mathbf{R}^{\succ} \mathbf{v}_j$:

$$\mathbf{v}_i^\top \mathbf{R}^{\succ} \mathbf{v}_j = \begin{bmatrix} \mathbf{a}_i^\top & \mathbf{b}_i^\top \end{bmatrix} \begin{bmatrix} \mathbf{0}_{k \times k} & -\mathbf{I}_k \\ \mathbf{I}_k & \mathbf{0}_{k \times k} \end{bmatrix} \begin{bmatrix} \mathbf{a}_j \\ \mathbf{b}_j \end{bmatrix} = -\mathbf{a}_i^\top \mathbf{b}_j + \mathbf{b}_i^\top \mathbf{a}_j.$$

Since $\mathbf{a}_i = \mathbf{e}_i$, we have:

$$\mathbf{a}_i^\top \mathbf{b}_j = \mathbf{e}_i^\top \mathbf{b}_j = (\mathbf{b}_j)_i = \frac{1}{2}P_{ji} = -\frac{1}{2}P_{ij}, \tag{13}$$

$$\mathbf{b}_i^\top \mathbf{a}_j = \mathbf{b}_i^\top \mathbf{e}_j = (\mathbf{b}_i)_j = \frac{1}{2}P_{ij}. \tag{14}$$

Therefore,

$$\mathbf{v}_i^\top \mathbf{R}^{\succ} \mathbf{v}_j = -\left(-\frac{1}{2}P_{ij}\right) + \frac{1}{2}P_{ij} = P_{ij}.$$

Thus, for all i, j,

$$P_{ij} = \mathbf{v}_i^\top \mathbf{R}^{\succ} \mathbf{v}_j.$$

This construction shows that any real skew-symmetric matrix $\mathbf{P}$ can be represented in terms of vectors $\{\mathbf{v}_i\} \subset \mathbb{R}^{2k}$ and the block-diagonal skew-symmetric matrix $\mathbf{R}^{\succ}$.

This completes the proof. □

Proof of the Theorem A.2.

Proof. We aim to represent any real skew-symmetric matrix $\mathbf{P} \in \mathbb{R}^{k \times k}$ using the imaginary parts of inner products of complex vectors.

For each $i = 1, \dots, k$, define the complex vector $\mathbf{v}_i = \mathbf{a}_i + i\,\mathbf{b}_i$, where $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^k$. Let $\mathbf{a}_i = \mathbf{e}_i$, the i-th standard basis vector in $\mathbb{R}^k$, and set

$$\mathbf{b}_i = \frac{1}{2} \sum_{j=1}^{k} P_{ij} \mathbf{e}_j.$$

This implies that the j-th component of $\mathbf{b}_i$ is $(\mathbf{b}_i)_j = \frac{1}{2} P_{ij}$.

The Hermitian inner product of $\mathbf{v}_i$ and $\mathbf{v}_j$ is

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = (\mathbf{a}_i^\top - i\,\mathbf{b}_i^\top)(\mathbf{a}_j + i\,\mathbf{b}_j) = \mathbf{a}_i^\top \mathbf{a}_j + \mathbf{b}_i^\top \mathbf{b}_j + i\,(\mathbf{b}_i^\top \mathbf{a}_j - \mathbf{a}_i^\top \mathbf{b}_j).$$

Therefore,

$$\operatorname{Im}\left(\langle \mathbf{v}_i, \mathbf{v}_j \rangle\right) = \mathbf{b}_i^\top \mathbf{a}_j - \mathbf{a}_i^\top \mathbf{b}_j.$$

Compute $\mathbf{b}_i^\top \mathbf{a}_j$ and $\mathbf{a}_i^\top \mathbf{b}_j$:

$$\begin{aligned} \mathbf{b}_i^\top \mathbf{a}_j &= (\mathbf{b}_i)_j = \frac{1}{2} P_{ij}, \\ \mathbf{a}_i^\top \mathbf{b}_j &= (\mathbf{b}_j)_i = \frac{1}{2} P_{ji} = -\frac{1}{2} P_{ij}, \end{aligned}$$

since $P_{ji} = -P_{ij}$ due to skew-symmetry.

Thus,

$$\operatorname{Im}\left(\langle \mathbf{v}_i, \mathbf{v}_j \rangle\right) = \frac{1}{2} P_{ij} - \left(-\frac{1}{2} P_{ij}\right) = P_{ij}.$$

Therefore, we have constructed complex vectors $\mathbf{v}_i$ such that

$$P_{ij} = \operatorname{Im}\left(\langle \mathbf{v}_i, \mathbf{v}_j \rangle\right), \quad \forall\, i, j.$$

This completes the proof. □

Proof of the Theorem 4.5.

Proof. Since $\mathbf{P}$ is real and skew-symmetric with even dimension $2k$, it can be brought into block-diagonal form via an orthogonal transformation. Specifically, there exists an orthogonal matrix $\mathbf{U} \in \mathbb{R}^{2k \times 2k}$ such that:

$$\mathbf{P} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top,$$

where $\mathbf{\Lambda}$ is a block-diagonal matrix composed of k blocks $\lambda_l \mathbf{J}$, with $\lambda_l \geq 0$ and

$$\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

This decomposition leverages the fact that the eigenvalues of $\mathbf{P}$ are purely imaginary and occur in conjugate pairs $\pm i\lambda_l$.

Define the block-diagonal matrix $\mathbf{R}^{\succ} = \operatorname{blockdiag}(\mathbf{J}, \dots, \mathbf{J}) \in \mathbb{R}^{2k \times 2k}$, and let

$\mathbf{D} = \operatorname{blockdiag}(\sqrt{\lambda_1}\mathbf{I}_2, \dots, \sqrt{\lambda_k}\mathbf{I}_2) \in \mathbb{R}^{2k \times 2k}$, where $\mathbf{I}_2$ is the 2×2 identity matrix.

Observe that $\mathbf{\Lambda} = \mathbf{D} \mathbf{R}^{\succ} \mathbf{D}$.

Set $\mathbf{V} = \mathbf{U}\mathbf{D}$. Then,

$$\mathbf{P} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top = \mathbf{U} \mathbf{D} \mathbf{R}^{\succ} \mathbf{D} \mathbf{U}^\top = \mathbf{V} \mathbf{R}^{\succ} \mathbf{V}^\top.$$

Therefore,

$$P_{ij} = \mathbf{v}_i^\top \mathbf{R}^{\succ} \mathbf{v}_j, \quad \forall\, i, j,$$

where $\mathbf{v}_i$ is the i-th row of $\mathbf{V}$.

This construction shows that any real skew-symmetric matrix $\mathbf{P}$ can be represented in terms of embeddings $\{\mathbf{v}_i\}$ and the asymmetric operator $\mathbf{R}^{\succ}$, confirming the full expressiveness of our preference representation model. □

B MORE ON EXPERIMENTS

B.1 ADDITIONAL EXPERIMENTAL RESULTS

More Results on Evaluating Language Model Alignment. We further conduct a rigorous evaluation of our downstream task-specific models using various benchmarks. AlpacaEval 2.0 evaluation results are listed in Table 4 and Table 5, using GPT-4o-mini and Deepseek-V2 as evaluators respectively. For MT-Bench, we used the default mode to let GPT-4 grade and give a score to the model's answer, and the MT-Bench scores of aligned models are presented in Table 6.

For LM-Harness, we chose Arc-Challenge, TruthfulQA, WinoGrande, GSM8k, HellaSwag, and MMLU as the evaluation tasks, and used the default rule-based evaluator of lm-evaluation-harness for accuracy calculation. These tasks are the same as those evaluated by Open LLM Leaderboard v1 (Beeching et al., 2023), which no longer provides service. To facilitate direct comparison with current state-of-the-art models, we adhere to the evaluation protocol established by the Open LLM Leaderboard v1. Our models are evaluated locally using this standardized framework. The resultant performance metrics are presented in Tables 8 and Table 9.

Table 4: AlpacaEval 2.0 evaluation results. Base model: LLama3-8B-it, Evaluator: GPT-4o-mini. The results are grouped by the size and type of the RM or PM, and the number of iterations. Bold entries indicate that the GPM outperforms the corresponding BT RM under the same training settings.

Size	**Type**	**Iter**	**SPPO**		**GPO**	
			Win Rate	**Avg. Len**	**Win Rate**	**Avg. Len**
		base	32.26	1959	32.26	1959
2B	**BT RM**	1	46.09	1939	49.94	1929
		2	58.41	2032	64.88	2049
		3	67.14	2136	71.68	2151
	GPM	1	**49.15 (+3.06)**	2066	**57.12 (+7.18)**	2102
		2	**63.53 (+5.12)**	2301	**67.78 (+2.90)**	2343
		3	**70.91 (+3.77)**	2498	**74.78 (+3.10)**	2582
8B	**BT RM**	1	36.95	1740	40.26	1702
		2	50.36	1868	56.30	1933
		3	58.38	1948	59.17	1969
	GPM	1	**41.42 (+4.47)**	1861	**46.64 (+6.38)**	1850
		2	**56.07 (+5.71)**	2029	**60.37 (+4.07)**	2115
		3	**63.42 (+5.04)**	2385	**67.48 (+8.31)**	3249

Table 5: AlpacaEval 2.0 evaluation results. Base model: LLama3-8B-it, Evaluator: DeepSeek-V2. The results are grouped by the size and type of the RM or PM, and the number of iterations. Bold entries indicate that the GPM outperforms the corresponding BT RM under the same training settings.

Size	**Type**	**Iter**	**SPPO**		**GPO**	
			Win Rate	**Avg. Len**	**Win Rate**	**Avg. Len**
		base	36.64	1959	36.64	1959
2B	**BT RM**	1	44.15	1939	45.94	1929
		2	53.42	2032	55.46	2049
		3	59.46	2136	60.83	2151
	GPM	1	**46.96 (+2.81)**	2066	**51.04 (+5.10)**	2102
		2	**54.66 (+1.24)**	2301	**59.19 (+3.73)**	2343
		3	**62.62 (+3.16)**	2498	**63.25 (+2.42)**	2582
8B	**BT RM**	1	39.19	1740	40.83	1702
		2	48.89	1868	53.05	1933
		3	52.06	1948	52.22	1969
	GPM	1	**43.07 (+3.88)**	1861	**45.16 (+4.33)**	1850
		2	**51.81 (+2.92)**	2029	**56.54 (+3.49)**	2115
		3	**56.83 (+4.77)**	2385	**60.59 (+8.37)**	3249

Table 6: MT-Bench evaluation results. Base model: LLama3-8B-it, Evaluator: GPT-4. Bold entries indicate that the GPM outperforms the corresponding BT RM under the same training settings.

Size	**Type**	**Iter**		**SPPO**			**GPO**	
			1st	**2nd**	**Avg.**	**1st**	**2nd**	**Avg.**
		base	8.31	7.77	8.03	8.31	7.77	8.03
2B	**BT RM**	1	8.42	7.57	8.00	8.33	7.85	8.09
		2	8.20	7.73	7.96	8.30	7.66	7.98
		3	8.44	7.66	8.05	8.41	8.09	8.25
	GPM	1	8.23	7.65	7.94	**8.70**	**7.95**	**8.33**
		2	**8.53**	**8.24**	**8.38**	**8.69**	**8.01**	**8.35**
		3	8.39	**7.84**	**8.12**	**8.48**	7.76	8.12
8B	**BT RM**	1	8.44	8.10	8.27	8.41	7.85	8.13
		2	8.75	7.85	8.30	8.73	7.83	8.28
		3	8.34	7.99	8.17	8.68	7.83	8.26
	GPM	1	8.43	7.94	8.18	8.29	**7.90**	8.10
		2	8.51	**8.05**	8.28	8.26	**7.99**	8.13
		3	8.47	7.76	8.12	7.57	7.51	7.54

Ablations on Scale Gate and Embedding head. We investigate the effects of scale gates and embedding head dimensions, with and without L2 normalization, on model performance. As shown in Table 7, for Gemma-2B-it models, incorporating a scale gate generally enhances GPM performance across various embedding dimensions. L2 normalization on the embedding head output consistently improves models with scale gates. Interestingly, Gemma-2B-it-based models without L2 normalization or scale gates outperform those with L2 normalization but no scale gates. A plausible explanation for this phenomenon is that removing L2 normalization introduces additional degrees of freedom, particularly beneficial for models with smaller parameter spaces and high-dimensional embedding layers. This increased flexibility may allow the model to better utilize its limited parametric capacity, potentially leading to enhanced expressiveness and task-specific adaptability.

For larger models, such as those based on Llama3.1-8B-Instruct, the impact of scale gates becomes less pronounced. This diminished effect may be attributed to the inherently stronger representational capacity of the 8B parameter model, which can likely capture complex patterns more effectively without additional architectural modifications.

These observations suggest a nuanced relationship between model size, normalization techniques, and architectural enhancements like scale gates, highlighting the importance of considering these factors in model design and optimization.

B.2 IMPLEMENTATION DETAILS

Details on Training Setup. Our experiments on RewardBench and Cyclic Preference Dataset were implemented using the HuggingFace Transformers library (Wolf et al., 2020) and the OpenRLHF framework (Hu et al., 2024). For reward model training on Skywork Reward Data Collection, we employed the following settings (in Table 10):

- **Gemma-2B-it:** Trained with a learning rate of 1×10^{-5}.
- **Llama-3.1-8B-Instruct:** Trained with a learning rate of 2×10^{-6}.
- **Training Configuration:** Both models were trained for two epochs with a global batch size of 32. We used a cosine learning rate scheduler with a warm-up ratio of 0.03. Input sequences were truncated to a maximum length of 2048 tokens.
- **Hyperparameters:** For our General Preference (GP) model, we set $\beta = 0.1$, determined via hyperparameter tuning on a validation set.
- **Hardware:** All experiments were conducted on machines equipped with NVIDIA A800 80GB GPUs, utilizing 8 GPUs per experiment.

Table 7: Impact of the embedding head and the scale gate on the GPM's performance on RewardBench. Dim. represents the dimension of the embedding head. The highest average scores for each base model are in bold and the second highest are underlined.

Embedding Type	Dim.	Chat	Chat-Hard	Safety	Reasoning	Average
Base Model: Gemma-2B-it						
w. scale gate w. l2	2	78.49	65.35	78.92	72.64	73.85
w. scale gate w.o. l2	2	76.82	67.76	79.19	75.12	74.72
w. o. scale gate w. l2	2	77.65	66.45	76.89	77.30	74.57
w. o. scale gate w.o. l2	2	79.61	65.13	80.27	78.98	**76.00**
w. scale gate w. l2	4	76.54	64.91	78.51	79.80	74.94
w. scale gate w.o. l2	4	78.49	66.89	77.70	78.14	75.30
w. o. scale gate w. l2	4	72.91	65.57	73.51	74.10	71.52
w. o. scale gate w.o. l2	4	76.54	69.30	79.46	77.19	**75.62**
w. scale gate w. l2	6	76.82	64.04	73.24	77.02	72.78
w. scale gate w.o. l2	6	75.98	68.64	75.54	76.36	74.13
w. o. scale gate w. l2	6	75.14	61.62	81.35	69.45	71.89
w. o. scale gate w.o. l2	6	80.73	66.45	77.30	81.24	**76.43**
w. scale gate w. l2	8	78.49	66.23	84.32	80.47	**77.38**
w. scale gate w.o. l2	8	74.58	68.20	80.00	78.11	75.22
w. o. scale gate w. l2	8	75.14	65.79	81.08	77.18	74.80
w. o. scale gate w.o. l2	8	75.14	65.57	79.19	80.77	75.17
Base Model: Llama-3.1-8B-Instruct						
w. scale gate w. l2	2	93.30	86.40	91.22	94.01	91.23
w. scale gate w.o. l2	2	93.85	86.84	90.68	91.60	90.74
w. o. scale gate w. l2	2	92.18	86.18	91.89	94.05	91.08
w. o. scale gate w.o. l2	2	93.30	87.94	91.22	93.55	**91.50**
w. scale gate w. l2	4	93.30	86.18	91.22	95.69	**91.60**
w. scale gate w.o. l2	4	94.13	86.18	89.86	90.55	90.18
w. o. scale gate w. l2	4	92.46	87.28	91.76	93.19	91.17
w. o. scale gate w.o. l2	4	93.58	86.40	90.95	95.33	91.56
w. scale gate w. l2	6	91.90	86.40	90.95	94.06	**90.83**
w. scale gate w.o. l2	6	93.02	85.75	91.08	91.31	90.29
w. o. scale gate w. l2	6	92.18	85.53	90.81	94.20	90.68
w. o. scale gate w.o. l2	6	93.30	87.94	90.95	90.90	90.77
w. scale gate w. l2	8	92.18	87.06	91.76	94.49	**91.37**
w. scale gate w.o. l2	8	93.02	87.06	90.81	92.20	90.77
w. o. scale gate w. l2	8	91.90	86.62	91.22	92.63	90.59
w. o. scale gate w.o. l2	8	93.02	87.72	90.68	90.16	90.39

For cyclic preference experiments, the training settings are as follows, except for the parameters specified below; all other experimental parameters remain consistent with experiments on RewardBench (in Table 11):

- **Gemma-2B-it:** Trained with a learning rate of 1×10^{-6}.
- **Training Configuration:** Models were trained for 50 epochs with a global batch size of 1.
- **Hardware:** Experiments were conducted on machines equipped with NVIDIA A800 80GB GPUs, utilizing a single GPU per experiment.

Details on Evaluation Dataset RewardBench. RewardBench is divided into four core sections:

- **Chat:** Evaluates the ability to differentiate between thorough and correct responses in open-ended conversations, using data from AlpacaEval (Li et al., 2023) and MT Bench (Zheng et al., 2023).
- **Chat-Hard:** Tests the handling of trick questions and subtle instruction differences, using adversarial examples from MT Bench and LLMBar (Zeng et al., 2024).

Table 8: Open LLM Leaderboard v1 evaluation results of LLama3-8B-it model fine-tuned using SPPO with BT reward model and our GPM.

Size	Type	Iter	SPPO						
			Arc	**TruthfulQA**	**WinoGrande**	**GSM8k**	**HellaSwag**	**MMLU**	**Average**
		base	62.03	51.65	75.53	75.28	78.77	65.67	68.16
2B	**BT RM**	1	62.63	53.16	75.06	75.82	78.83	65.99	68.58
		2	63.05	53.23	74.43	77.63	78.85	66.06	68.88
		3	62.37	52.95	74.19	77.33	78.66	65.97	68.58
	GPM	1	63.14	53.09	74.98	75.44	78.99	65.74	68.56
		2	62.88	52.67	74.82	75.21	78.89	65.62	68.35
		3	63.23	53.06	74.90	75.51	78.88	65.59	68.53
8B	**BT RM**	1	64.59	56.30	75.30	76.80	79.42	65.72	69.69
		2	65.02	56.04	75.45	76.88	79.67	65.88	69.82
		3	65.44	56.13	74.98	76.35	79.50	66.15	69.76
	GPM	1	64.85	55.65	75.06	78.09	79.55	65.83	69.84
		2	64.51	55.66	74.98	76.42	79.41	65.77	69.46
		3	64.93	55.59	75.22	76.5	79.3	65.54	69.51

Table 9: Open LLM Leaderboard v1 evaluation results of LLama3-8B-it model fine-tuned using GPO with BT reward model and our GPM.

Size	Type	Iter	GPO						
			Arc	**TruthfulQA**	**WinoGrande**	**GSM8k**	**HellaSwag**	**MMLU**	**Average**
		base	62.03	51.65	75.53	75.28	78.77	65.67	68.16
2B	**BT RM**	1	63.31	54.01	74.19	77.41	78.65	65.83	68.90
		2	62.71	54.18	73.88	75.44	78.50	65.87	68.43
		3	62.03	54.54	73.16	76.57	78.58	65.87	68.46
	GPM	1	63.74	53.28	74.82	76.65	78.70	65.87	68.84
		2	62.80	52.98	74.66	76.19	78.74	65.69	68.51
		3	62.71	52.78	74.74	75.59	78.61	65.67	68.35
8B	**BT RM**	1	64.51	57.36	75.06	76.27	79.46	65.56	69.70
		2	64.85	56.25	74.90	76.35	79.35	65.71	69.57
		3	64.76	56.22	74.03	76.80	78.78	65.89	69.41
	GPM	1	64.51	56.01	74.82	78.47	79.17	65.64	69.77
		2	64.16	54.57	73.95	76.88	78.67	65.82	69.01
		3	63.40	54.46	73.56	77.63	78.19	65.51	68.79

- **Safety:** Assesses the capacity to refuse harmful content appropriately, using data from XSTest (Röttger et al., 2024), Do-Not-Answer (Wang et al., 2024), and a custom AI2 dataset.
- **Reasoning:** Measures code generation and reasoning abilities, with prompts from HumanEvalPack (Muennighoff et al., 2023) and PRM800k (Lightman et al., 2023).

C MORE ON GENERAL PREFERENCE OPTIMIZATION

The von Neumann winner represents a fundamental concept in social choice theory (Sen, 1986) that has found significant applications in preference-based reinforcement learning (Owen, 2013; Dudík et al., 2015). It corresponds to the Nash equilibrium of a two-player symmetric game (Equation 7), representing a mixed strategy—a probability distribution over possible responses—that performs optimally against any opponent in the worst-case scenario.

For notational clarity, we define the preference score of a policy π over another policy π' as:

$$s\left(\pi \succ \pi' \mid \mathbf{x}\right) = \mathbb{E}_{\mathbf{y}\sim\pi(\cdot|\mathbf{x}),\ \mathbf{y}'\sim\pi'(\cdot|\mathbf{x})}\left[s\left(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}\right)\right]. \tag{15}$$

A distribution π^* is formally defined as a von Neumann winner when it satisfies:

$$\min_{\pi'\in\Delta}\ \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[s\left(\pi^* \succ \pi' \mid \mathbf{x}\right)\right] \geq 0. \tag{16}$$

Table 10: Implementation details for experiments on RewardBench.

General Settings	
Base models	Gemma-2b-it and Llama3.1-8B-Instruct
Batch size	32
Quantization for training	bf16
Learning Rate	1×10^{-5} for Gemma and 2×10^{-6} for Llama3.1
Learning Rate Scheduler	cosine
Warmup Ratio	0.03
Max training epochs	2
Gradient accumulation step	1
Max input length	2048
Zero stage	3
Flash attention enabled	True
General Preference Model	
β for loss function	0.1
Bradly Terry Model	
β for loss function	1

Table 11: Implementation details for experiments on Cyclic Preference Dataset.

General Settings	
Base models	Gemma-2b-it
Batch size	1
Quantization for training	bf16
Learning Rate	1×10^{-6}
Learning Rate Scheduler	cosine
Warmup Ratio	0.03
Max training epochs	50
Gradient accumulation step	1
Max input length	2048
Zero stage	3
Flash attention enabled	True
General Preference Model	
β for loss function	0.1
Bradly Terry Model	
β for loss function	1

This condition ensures that π^* is, on average, at least as preferred as any other policy π'. The symmetric nature of the two-player game (Equation 7) guarantees the existence of such a winner.

General Preference Optimization (GPO) employs an iterative framework inspired by the multiplicative weights update (MWU) algorithm (Freund & Schapire, 1999). The update rule is formulated as:

$$\pi_{t+1}(\mathbf{y} \mid \mathbf{x}) \propto \pi_t(\mathbf{y} \mid \mathbf{x}) \exp\left(\eta \cdot s\left(\mathbf{y} \succ \pi_t \mid \mathbf{x}\right)\right), \quad t = 1, 2, \ldots, \tag{17}$$

where η denotes the learning rate and $s\left(\mathbf{y} \succ \pi_t \mid \mathbf{x}\right)$ represents the preference score of response $\mathbf{y}$ over the current policy π_t given prompt $\mathbf{x}$. The following theorem establishes the convergence properties of GPO (analogous to Theorem 4.1 in Wu et al. (2024b)):

Theorem C.1. *Consider the optimization problem defined by the GPO loss (Equation 12) and assume it is realizable. Let* $\{\pi_{\boldsymbol{\theta}_t}\}_{t=1}^{T}$ *denote the sequence of policies generated by GPO, and define* $\bar{\pi}_T = \frac{1}{T}\sum_{t=1}^{T} \pi_{\boldsymbol{\theta}_t}$ *as the average policy. Given that the preference score s is bounded within* $[-\rho, \rho]$*,* *by setting* $\beta = \Theta\left(\sqrt{T}\right)$*, we have:*

$$\max_\pi s\left(\pi \succ \bar{\pi}_T\right) - \min_\pi s\left(\pi \prec \bar{\pi}_T\right) = O\left(\frac{1}{\sqrt{T}}\right).$$

Proof. First, since the preference score s is bounded in $[-\rho, \rho]$, we can normalize it to $[0, 1]$ by the transformation:

$$\widetilde{s}(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x}) = \frac{s(\mathbf{y} \succ \mathbf{y}' \mid \mathbf{x})}{2\rho} + \frac{1}{2}$$

By Theorem 1 in Freund & Schapire (1999), for any sequence of mixed policies $\mu_1, \mu_2, \ldots, \mu_T$, the sequence of policies $\pi_1, \pi_2, \ldots, \pi_T$ produced by GPO satisfies:

$$\sum_{t=1}^{T} \widetilde{s}(\pi_t \prec \mu_t) \leq \min_\pi \left[\frac{\eta}{1-e^{-\eta}} \sum_{t=1}^{T} \widetilde{s}(\pi \prec \mu_t) + \frac{\mathrm{KL}(\pi \| \pi_0)}{1-e^{-\eta}}\right]$$

Setting $\mu_t = \pi_t$, note that $\widetilde{s}(\pi_t \prec \pi_t) = \frac{1}{2}$ due to the normalization and symmetry. Thus:

$$\frac{T}{2} \leq \min_\pi \left[\frac{\eta T}{1-e^{-\eta}} \widetilde{s}(\pi \prec \bar{\pi}_T) + \frac{\mathrm{KL}(\pi \| \pi_0)}{1-e^{-\eta}}\right]$$

where $\bar{\pi}_T = \frac{1}{T}\sum_{t=1}^{T} \pi_t$ is the mixture policy.

Rearranging terms:

$$\frac{1-e^{-\eta}}{2\eta} \leq \min_\pi \left[\widetilde{s}(\pi \prec \bar{\pi}_T) + \frac{\mathrm{KL}(\pi \| \pi_0)}{\eta T}\right]$$

Since π_0 is an autoregressive model with finite vocabulary support, $|\log \pi_0(\cdot)|$ is bounded from above. Thus:

$$\mathrm{KL}(\pi \| \pi_0) \leq \|\log \pi_0(\cdot)\|_\infty$$

Setting $\eta = \frac{\|\log \pi_0(\cdot)\|_\infty}{\sqrt{T}}$ and using Taylor expansion $\frac{1-e^{-\eta}}{2\eta} = \frac{1}{2} - \frac{\eta}{4} + O(\eta^2)$:

$$\frac{1}{2} - \frac{\|\log \pi_0(\cdot)\|_\infty}{4\sqrt{T}} + O(T^{-1}) \leq \min_\pi \left[\widetilde{s}(\pi \prec \bar{\pi}_T)\right] + \sqrt{\frac{\|\log \pi_0(\cdot)\|_\infty}{T}}$$

Converting back to the original preference score scale:

$$\min_\pi \left[s(\pi \prec \bar{\pi}_T)\right] \geq -\frac{\rho}{2} - O\left(\frac{\rho}{\sqrt{T}}\right)$$

By symmetry:

$$\max_\pi \left[s(\pi \succ \bar{\pi}_T)\right] \leq \frac{\rho}{2} + O\left(\frac{\rho}{\sqrt{T}}\right)$$

Therefore, the duality gap is:

$$\begin{aligned}
&\max_\pi s(\pi \succ \bar{\pi}_T) - \min_\pi s(\pi \prec \bar{\pi}_T) \\
&= \max_\pi s(\pi \succ \bar{\pi}_T) - \min_\pi s(\pi \prec \bar{\pi}_T) \\
&= O\left(\frac{1}{\sqrt{T}}\right)
\end{aligned}$$

□

Connection to Policy Gradient. Applying policy gradient theorem on Equation (10) gives:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \mathbf{y} \sim \pi_{\boldsymbol{\theta}}} \left[\widehat{s}(\mathbf{y} \succ \pi_{\boldsymbol{\theta}_t}) - \beta \log \frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right]$$

$$= \mathbb{E}_{\mathbf{x}\sim\mathcal{X},\mathbf{y}\sim\pi_{\boldsymbol{\theta}}}\left[\left(\widehat{s}(\mathbf{y} \succ \pi_{\boldsymbol{\theta}_t}) - \beta \log \frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right)\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})\right]$$

$$= \mathbb{E}_{\mathbf{x}\sim\mathcal{X},\mathbf{y}\sim\pi_{\boldsymbol{\theta}}}\left[-\nabla_{\boldsymbol{\theta}}\left(\widehat{s}(\mathbf{y} \succ \pi_{\boldsymbol{\theta}_t}) - \beta \log \frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\pi_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right)^2\right].$$

So Equation (12) can also be seen as an offline policy gradient method for the optimization problem (10).

D MORE RELATED WORK

Intransitivity in Game Theory. The symmetric zero-sum game and its intransitivity have also been frequently studied in the context of game theory. Balduzzi et al. (2018) was motivated by evaluation among different agents, showing that any symmetric zero-sum game can be decomposed into a "transitive" game and a "cyclic" game, and proposed Nash averaging for better agent/task evaluation. Balduzzi et al. (2019) generalized the results from matrix games to functional-form games and propose new algorithms to construct diverse populations of effective agents. Czarnecki et al. (2020) investigated the geometrical properties of real-world games (e.g., Tic-Tac-Toe, Go, StarCraft II) and proposed that real-world games have a "spinning top" geometry, with a strong transitive dimension and gradually diminishing non-transitive cyclic dimensions. Very recently, Bertrand et al. (2023) examined the limitations of the Elo rating system and proposed an alternative "disc decomposition" method that can better handle both transitive and cyclic game dynamics.

Representation Learning and Embedding. Representation learning and embedding techniques have successfully captured relational structures across various domains (Mikolov et al., 2013; Chen et al., 2020; Radford et al., 2021), yet their application in preference modeling and RLHF remains limited. Our work introduces preference representation learning, an approach that enhances expressiveness while maintaining computational efficiency, bridging the gap left by traditional approaches.