Robust Preference Alignment via Directional Neighborhood Consensus

Anonymous Authors
Anonymous Institution
Anonymous Address
anonymous@email.com

August 18, 2025

Abstract

Preference alignment in large language models faces a fundamental challenge: insufficient coverage of the vast preference space during training, leading to brittleness when users request preferences outside the training distribution. We propose Robust Preference Selection (RPS), a post-hoc method that leverages directional neighborhood consensus to improve robustness without retraining. Our key insight is to sample from a local neighborhood around the target preference and select the best response using the user's true preference as evaluation criterion. We provide theoretical guarantees showing that RPS achieves better expected performance than single-direction generation, with robustness gain quantified by neighborhood diversity and preference consistency. Comprehensive experiments demonstrate consistent improvements across diverse preference configurations, with RPS achieving up to 30.2% advantage over baseline methods in challenging preference directions. Preliminary results suggest similar benefits extend to other preference alignment approaches. Our work provides a theoretically grounded solution for robust preference alignment that enhances model reliability in real-world deployment scenarios.

1 Introduction

Preference alignment in large language models (LLMs) has become crucial for deploying AI systems that meet diverse user needs **?**. However, current approaches face a fundamental challenge: **insufficient preference coverage during training**. Training datasets contain only sparse sampling of the vast preference space, making it difficult to customize user preferences at inference time.

This limitation manifests in two critical ways:

1. **Training Coverage Gap**: Models are trained on a limited set of preference directions, leaving substantial gaps in the preference space where user preferences may actually lie
2. **Preference Customization Challenge**: Adapting pre-trained models to meet diversified user preferences in real-world scenarios is non-trivial, especially when user preferences fall outside the training distribution

Consider the preference alignment problem in the context of helpfulness versus verbosity trade-offs. Users may want different balances: some prefer concise, direct answers (high helpfulness, low

verbosity), while others desire detailed explanations (high helpfulness, high verbosity). As illustrated in Figure 2, training typically covers only a constrained angular range (e.g., $\theta \in [-\pi/4, 0]$), leaving large regions of user preference space unexplored.

When users request preferences outside or near the boundaries of the training distribution, existing methods exhibit instability—small changes in preference specification can lead to dramatically different response quality **?**. This brittleness severely limits the practical deployment of preference-aligned models.

To address these challenges, we propose **Robust Preference Selection** (RPS), a post-hoc adjustment method that enables reliable preference customization without retraining. Our key insight is to leverage **directional neighborhood consensus**: instead of generating from a single preference direction, we sample from a local neighborhood around the target preference and select the best response using the user's true preference as the evaluation criterion.

Our contributions are:

- We formally characterize the preference coverage problem and its impact on model robustness
- We propose RPS, a theoretically grounded method for robust preference alignment at inference time
- We provide comprehensive experimental validation showing consistent improvements across diverse preference configurations

2 Problem Setup and Theoretical Framework

2.1 Preference Space Definition

We consider preference alignment in a two-dimensional space representing the trade-off between two key response characteristics:

Helpfulness: The degree to which a response addresses the user's question effectively, including accuracy, relevance, and practical utility. This encompasses how well the response solves the user's problem or provides valuable information.

Verbosity: The level of detail and elaboration in the response, including explanation depth, example richness, and overall length. This captures whether users prefer concise answers or comprehensive explanations.

Reward Function: We employ a pre-trained reward model to quantify response quality along these dimensions. For any prompt x and response y, the reward function $\mathbf{r}(x, y) = (r_h(x, y), r_v(x, y))$ provides numerical scores where:

- $r_h(x, y) \in \mathbb{R}$: helpfulness score measuring how well response y addresses prompt x
- $r_v(x, y) \in \mathbb{R}$: verbosity score measuring the level of detail and elaboration in response y

User preferences are represented as vectors $\mathbf{v} = (v_h, v_v) \in \mathbb{S}^1$ on the unit circle, where v_h and v_v denote the relative weights for helpfulness and verbosity respectively. We can parameterize preferences using angle notation: $\mathbf{v} = (\cos\theta, \sin\theta)$, where θ represents the preference direction.

Figure 1 illustrates the full range of user preferences that may be encountered at inference time. Users can specify any direction on the unit circle, representing different trade-offs between helpfulness and verbosity.

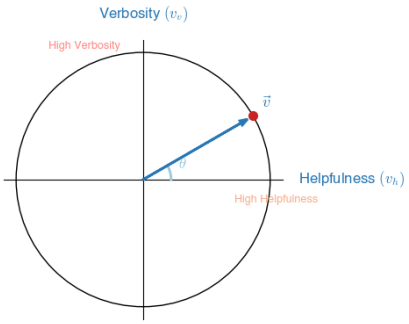

Figure 1: Complete user preference space on the unit circle. Each arrow represents a possible user preference direction, spanning angles from 0° to 45° and beyond. This demonstrates the full spectrum of helpfulness-verbosity trade-offs that users may request at inference time.

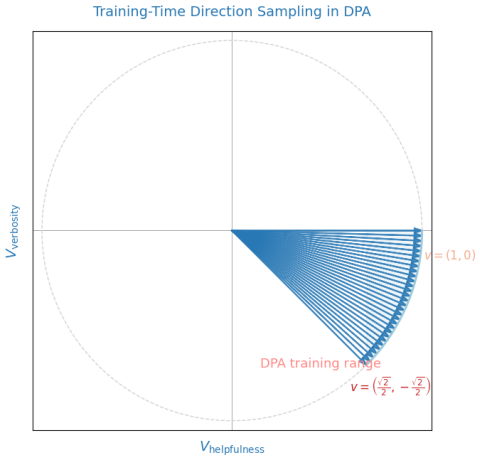

Figure 2: Training-time preference coverage in DPA. The blue shaded region represents the constrained angular range used during training, demonstrating the sparse coverage of the full preference space. The dense sampling within this limited range contrasts sharply with the complete absence of training data in other regions, creating significant coverage gaps.

2.2 The Preference Coverage Problem

Definition 1 (Training Preference Set): Let $\mathcal{V}_{train} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ denote the set of preference directions used during model training, typically sampled from a constrained range to avoid preference conflicts.

Definition 2 (User Preference Space): Let $\mathcal{V}_{user}$ represent the full space of preferences that users may specify at inference time, which can span the entire unit circle $\mathbb{S}^1$.

Definition 3 (Preference Coverage Gap): The coverage gap is defined as:

$$\text{Gap} = \mathcal{V}_{user} \setminus \mathcal{N}_\epsilon(\mathcal{V}_{train}) \tag{1}$$

where $\mathcal{N}_\epsilon(\mathcal{V}_{train}) = \{\mathbf{v} \mid \min_{\mathbf{v}' \in \mathcal{V}_{train}} \|\mathbf{v} - \mathbf{v}'\| \leq \epsilon\}$ is the ϵ-neighborhood of training preferences.

The severity of this coverage problem is illustrated in Figure 2. During DPA training, preference directions are sampled from a constrained angular range, typically $\theta \in [-\pi/4, 0]$, which covers only a small fraction of the complete preference space. This leaves substantial gaps where user preferences may lie but the model has no training experience.

Problem Statement: Given a prompt x and target preference $\mathbf{v}_{target} \in$ Gap, generate a high-quality response y^* that maximizes user satisfaction:

$$y^* = \arg\max_y \mathbf{v}_{target}^T \mathbf{r}(x, y) \tag{2}$$

where $\mathbf{r}(x, y) = (r_h(x, y), r_v(x, y))$ represents the helpfulness and verbosity scores of response y to prompt x.

2.3 Neighborhood Consensus Theory

To address the coverage gap, we propose using neighborhood consensus. The key insight is that preferences within a local neighborhood should yield similar quality rankings.

Assumption 1 (Local Preference Consistency): For sufficiently small angular threshold θ_{max}, preferences within the neighborhood of $\mathbf{v}_{target}$ produce similar response quality evaluations:

$$\forall \mathbf{v}_i, \mathbf{v}_j \in \mathcal{N}_\theta(\mathbf{v}_{target}), \quad |\mathbf{v}_i^T \mathbf{r}(x, y) - \mathbf{v}_j^T \mathbf{r}(x, y)| \leq \delta \tag{3}$$

where $\mathcal{N}_\theta(\mathbf{v}_{target}) = \{\mathbf{v} \mid \arccos(\mathbf{v} \cdot \mathbf{v}_{target}) \leq \theta_{max}\}$.

Assumption 2 (Sub-Gaussian Diversity): The centered target scores $s_i - \mathbb{E}[s_i]$ for $s_i = \mathbf{v}_{target}^T \mathbf{r}(x, y_i)$ are independent and sub-Gaussian with parameter σ^2; for example, one may take $s_i = \mu + \sigma Z_i$ with Z_i i.i.d. standard normal.

Theorem 1 (Robustness Gain). Let $\mathcal{N}_k = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be a set of k preference directions in the neighborhood of $\mathbf{v}_{target}$, with corresponding generated responses $\{y_1, \ldots, y_k\}$. Assume Local Preference Consistency (Assumption 1) holds and the target scores $s_i = \mathbf{v}_{target}^T \mathbf{r}(x, y_i)$ have finite variance $\sigma^2 > 0$. The consensus selection

$$y^* = \arg \max_{i \in \{1, \ldots, k\}} \mathbf{v}_{target}^T \mathbf{r}(x, y_i) \tag{4}$$

achieves better expected performance than any single-direction generation, with robustness gain

$$\mathbb{E}[\mathbf{v}_{target}^T \mathbf{r}(x, y^*)] - \mathbb{E}[\mathbf{v}_{target}^T \mathbf{r}(x, y_{single})] \geq \epsilon_{robust}, \tag{5}$$

where

$$\epsilon_{robust} = \sigma \Phi^{-1}\left(1 - \frac{1}{k}\right) + \delta_{consistency} > 0, \tag{6}$$

and $\delta_{consistency} > 0$ denotes the stability margin implied by Assumption 1.

Proof: Assume there exist constants $\delta_{consistency} > 0$ and variance $\sigma^2 > 0$ satisfying the conditions above. We prove the claim through the following steps:

Step 1 (Setup): Let $s_i = \mathbf{v}_{target}^T \mathbf{r}(x, y_i)$ denote the target preference score for response y_i generated from direction $\mathbf{v}_i \in \mathcal{N}_k$. The consensus selection chooses $y^* = \arg\max_i s_i$, giving score $s^* = \max\{s_1, s_2, \ldots, s_k\}$.

Step 2 (Local Consistency Application): From Assumption 1 (Equation 3), for any $\mathbf{v}_i \in \mathcal{N}_\theta(\mathbf{v}_{target})$ and response y, preferences within the neighborhood produce similar response quality evaluations. This ensures that responses generated from neighborhood directions maintain quality alignment with the target preference.

Step 3 (Maximum Value Advantage): The key insight is that $s^* = \max\{s_1, \ldots, s_k\} \geq s_i$ for all i. For any single-direction method that randomly selects one response, the expected performance is:

$$\mathbb{E}[\mathbf{v}_{target}^T \mathbf{r}(x, y_{single})] = \mathbb{E}[s_i] = \frac{1}{k} \sum_{i=1}^{k} \mathbb{E}[s_i] \tag{7}$$

Step 4 (Expected Maximum Bound): By the fundamental inequality for order statistics:

$$\mathbb{E}[\max\{s_1, \ldots, s_k\}] \geq \max\{\mathbb{E}[s_1], \ldots, \mathbb{E}[s_k]\} \geq \frac{1}{k} \sum_{i=1}^{k} \mathbb{E}[s_i] \tag{8}$$

Step 5 (Diversity Gain Quantification): Under Assumption 2, by standard order-statistics bounds for sub-Gaussian (e.g., Gaussian) variables,

$$\mathbb{E}[\max\{s_1, \ldots, s_k\}] \geq \mathbb{E}[s_i] + \sigma\, \Phi^{-1}\left(1 - \frac{1}{k}\right), \tag{9}$$

where Φ^{-1} is the inverse standard normal CDF.

Step 6 (Combining Bounds): Combining the consistency guarantee with the diversity gain yields the robustness margin stated above,

$$\epsilon_{robust} = \sigma\Phi^{-1}\left(1 - \frac{1}{k}\right) + \delta_{consistency} > 0, \tag{10}$$

where $\delta_{consistency} > 0$ captures the additional stability from neighborhood consensus.

Therefore, the consensus selection y^* achieves superior expected performance, with the robustness gain ϵ_{robust} quantifying the improvement over single-direction generation. □

Corollary 1: The robustness gain ϵ_{robust} increases with neighborhood size k and decreases with preference inconsistency, making the method particularly effective for larger neighborhoods and well-aligned preference directions.

2.4 Robust Preference Selection Algorithm

Algorithm 1 Robust Preference Selection (RPS)

Require: Prompt x, target preference $\mathbf{v}_{target}$, neighborhood size k, angle threshold θ_{max}
Ensure: Optimal response y^*

Phase 1: Neighborhood Construction
Generate candidate directions within θ_{max} of $\mathbf{v}_{target}$
Compute angular distances: $d_i = \arccos(\mathbf{v}_i \cdot \mathbf{v}_{target})$
Select k closest directions: $\mathcal{N}_k = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$
Phase 2: Multi-Directional Generation
for $i = 1$ to k **do**
    Generate response: $y_i \sim \pi_\theta(\cdot|x, \mathbf{v}_i)$
end for
Phase 3: Consensus Selection
for $i = 1$ to k **do**
    Compute score: $s_i = \mathbf{v}_{target}^T \mathbf{r}(x, y_i)$
end for
return $y^* = \arg\max_i s_i$

This unified algorithm addresses the preference coverage gap by: (1) constructing a local neighborhood around the target preference, (2) generating diverse responses from multiple directions within this neighborhood, and (3) selecting the response that best aligns with the user's true preference using reward model scoring.

3 Methodology

3.1 Baseline Methods

We evaluate RPS across three preference alignment approaches:

3.1.1 Directional Preference Alignment (DPA)

We adopt the Directional Preference Alignment (DPA) framework **?**, where user intent is encoded as a direction vector $\mathbf{v} = (\cos\theta, \sin\theta)$ on the unit circle. Each direction corresponds to a specific trade-off between helpfulness and verbosity. During training, DPA samples preference directions from a constrained angular range, typically $\theta \in [-\frac{\pi}{4}, 0]$, to promote helpfulness while avoiding verbosity inflation.

3.1.2 Direct Preference Optimization (DPO)

DPO **?** directly optimizes language models on preference data without explicit reward model training. We evaluate RPS on DPO-trained models to assess robustness gains across different training paradigms.

3.1.3 SteerLM

SteerLM **?** enables fine-grained control over response attributes. Evaluation will assess RPS effectiveness on controllable generation frameworks.

3.2 Proposed Method: Robust Preference Selection (RPS)

RPS can be applied as a post-hoc method to any of the above baseline approaches. The core idea is to aggregate over a local neighborhood of preference directions to improve stability and generalization.

3.3 Implementation Details

Reward Model: We use a pre-trained reward model to provide $r_h(x, y)$ and $r_v(x, y)$ for each response.

Input Format: For each prompt x and preference vector $\mathbf{v} = (v_h, v_v)$, we construct the system instruction as:

> "You are a helpful assistant. Your response should maximize weighted rating = helpfulness$\star v_h$ + verbosity$\star v_v$."

The user prompt is appended after the instruction.

Parameter Settings: We sample k directions in the neighborhood $\mathcal{N}_\theta(\mathbf{v}_{\text{target}})$, with $\theta_{\max}$ typically set as 30°. For each direction, one response is generated. The best response is selected by the consensus rule above.

3.4 Baseline Method Implementations

To ensure comprehensive evaluation, we implement RPS across three different preference alignment paradigms, each with distinct control mechanisms and baseline strategies.

3.4.1 Directional Preference Alignment (DPA) Baseline

Control Mechanism: DPA uses continuous direction vectors $\mathbf{v} = (v_1, v_2)$ to specify preference trade-offs between helpfulness and verbosity.

Input Format: For each prompt x and preference vector $\mathbf{v} = (v_1, v_2)$, we construct:

```
System: "You are a helpful assistant. Your response should
maximize weighted rating = helpfulness*h + verbosity*v."
User: ⟨original prompt⟩
```

where $h = \text{round}(v_1 \times 100)$ and $v = \text{round}(v_2 \times 100)$.

Baseline Strategy: Single-direction generation with multiple sampling

1. Generate 3 responses using the target direction $\mathbf{v}_{\text{target}}$
2. Evaluate each response using DPA scoring: $s = v_1 \cdot r_{\text{help}}(x, y) + v_2 \cdot r_{\text{verb}}(x, y)$
3. Select the response with highest DPA score

RPS Strategy: Neighborhood consensus with directional perturbation

1. Sample $k = 5$ directions within $\theta_{\max} = 30$ of $\mathbf{v}_{\text{target}}$
2. Generate one response per neighborhood direction
3. Evaluate all responses using $\mathbf{v}_{\text{target}}$ as scoring criterion
4. Select the response with highest target-direction score

3.4.2 Direct Preference Optimization (DPO) Baseline

Control Mechanism: DPO incorporates preference information through direct optimization on preference data, with preference specification via prompt engineering.

Input Format: Similar to DPA, preference vectors are encoded as numerical weights in the system instruction.

Baseline Strategy: Fixed-prompt multiple sampling

1. Generate 3 responses using the same preference-encoded prompt
2. Score each response using the DPA reward model with target direction weights
3. Select the highest-scoring response

RPS Strategy: Neighborhood prompt exploration

1. Create $k = 5$ prompt variations with perturbed preference weights
2. Generate one response per prompt variation
3. Evaluate using original target direction as scoring standard
4. Select optimal response based on target preference alignment

3.4.3 SteerLM Baseline

Control Mechanism: SteerLM uses discrete attribute values to control multiple response characteristics simultaneously.

Input Format: Following SteerLM's format with attribute-conditioned generation:

```
<extra_id_0>System
A chat between a curious user and an artificial intelligence
assistant.  The assistant gives helpful, detailed, and polite
answers to the user's questions.
<extra_id_1>User
⟨prompt⟩
<extra_id_1>Assistant
<extra_id_2>⟨attribute_string⟩
```

DPA Direction to SteerLM Attribute Mapping: We map continuous DPA preference directions to discrete SteerLM attributes using the following transformation:

$$\text{helpfulness} = \max(0, \min(4, \text{round}(v_1 \times 4))) \tag{11}$$

$$\text{verbosity} = \max(0, \min(4, \text{round}(v_2 \times 4))) \tag{12}$$

where (v_1, v_2) represents the DPA direction vector. Other attributes are fixed to ensure controlled comparison: quality=4, complexity=2, creativity=1, coherence=4, correctness=4, toxicity=0, humor=0.

Baseline Strategy: Fixed-attribute multiple sampling

1. Map DPA direction $\mathbf{v}_{\text{target}} = (v_1, v_2)$ to central SteerLM attributes using Equations (10)-(11)
2. Generate 3 candidate responses using identical attribute configuration with temperature=0.7
3. Score each response using DPA evaluation: $s_i = v_1 \cdot r_{\text{help}}(x, y_i) + v_2 \cdot r_{\text{verb}}(x, y_i)$
4. Select response with highest DPA score: $y^* = \arg\max_i s_i$

This baseline represents the standard approach of generating multiple samples from a fixed configuration and selecting the best according to the target preference.

3.4.4 SteerLM RPS Strategy

Algorithm Implementation: Our SteerLM RPS method strictly follows Algorithm 1 (Section 2.4) with necessary adaptations for SteerLM's discrete attribute space.

Phase 1: Neighborhood Construction in Angle Space

1. Generate candidate directions within $\theta_{\max} = 30$ of $\mathbf{v}_{\text{target}}$ using step size of 5°
2. Compute angular distances: $d_i = \arccos(\mathbf{v}_i \cdot \mathbf{v}_{\text{target}})$
3. Select $k = 5$ closest directions: $\mathcal{N}_k = \{\mathbf{v}_1, \ldots, \mathbf{v}_5\}$

Phase 2: Multi-Directional Generation with Attribute Mapping

1. For each perturbed direction $\mathbf{v}_i \in \mathcal{N}_k$:
   (a) Map to SteerLM attributes using Equations (10)-(11)
   (b) Generate response: $y_i \sim \pi_\theta(\cdot|x, \text{steerlm\_attrs}(\mathbf{v}_i))$

Phase 3: Consensus Selection

1. Score all responses using original target direction: $s_i = \mathbf{v}_{\text{target}}^T \mathbf{r}(x, y_i)$

2. Select optimal response: $y^* = \arg\max_i s_i$

Key Adaptation for SteerLM: While Algorithm 1 specifies direct generation from preference directions, SteerLM requires discrete attributes. Our adaptation maintains algorithmic consistency by:

- Performing neighborhood construction in continuous angle space (preserving Algorithm 1's Phase 1)
- Mapping each perturbed direction to appropriate SteerLM attributes
- Using unified evaluation with the original target direction (preserving Algorithm 1's Phase 3)

This ensures that RPS benefits from directional diversity while respecting SteerLM's discrete control mechanism.

Comparison with Baseline: The key difference lies in exploration strategy:

- **Baseline**: 3 samples from fixed central attribute configuration
- **RPS**: 1 sample each from 5 different attribute configurations derived from neighborhood directions

Both methods generate the same total number of candidate responses (3-5 range) but RPS explores a more diverse attribute space, potentially discovering better configurations than the direct mapping of the target direction.

3.4.5 Experimental Consistency

All SteerLM experiments maintain consistent conditions with DPA and DPO evaluations:

- **Sample size:** 100 prompts per direction, 8 directions (v3-v10)
- **Neighborhood size:** $k = 5$ directions for RPS
- **Baseline sampling:** 3 responses per prompt for fair comparison
- **Evaluation standard:** DPA reward model with original target direction weights
- **Selection criterion:** Maximize $\mathbf{v}_{\text{target}}^T \mathbf{r}(x, y)$ for all methods
- **Angle perturbation:** Consistent with DPA: range $(-40, 40)$, step=5°, $\theta_{\max} = 30$

This unified framework enables direct comparison of RPS effectiveness across different preference control paradigms while preserving each method's distinct characteristics.

3.4.6 Experimental Consistency

All three methods maintain consistent experimental conditions:

- **Sample size:** 100 prompts per direction, 8 directions (v3-v10)
- **Neighborhood size:** $k = 5$ candidates for RPS across all methods
- **Baseline sampling:** 3 responses per prompt for fair comparison

Table 1: Current Experimental Completion Status

Method	Vectors	Sample Size	Status	Data Files
DPA	v3-v10	2,000 each	✓ Complete	Analysis complete
DPO	v3-v10	2,000 each	✓ Complete	Analysis complete
SteerLM	v3-v10	2,000 planned	○ Planned	Experiment pending

- **Evaluation standard:** DPA reward model with original target direction weights
- **Selection criterion:** Maximize $\mathbf{v}_{\text{target}}^T \mathbf{r}(x, y)$ for all methods

This unified framework enables direct comparison of RPS effectiveness across different preference control paradigms while preserving each method's distinct characteristics.

4 Experimental Setup

4.1 Models and Implementation

We evaluate RPS on three different preference alignment frameworks:

- **DPA-v1-Mistral-7B**: Baseline DPA model trained on directional preferences
- **DPO-Mistral-7B**: Model trained with Direct Preference Optimization
- **SteerLM-Mistral-7B**: Model trained with SteerLM controllable generation

For each baseline method, we implement both the standard single-direction generation and our proposed RPS enhancement.

4.2 Dataset and Evaluation Protocol

We conduct experiments on the ultrafeedback_binarized dataset. For each baseline method and angular configuration, we aim to evaluate 2,000 prompt-response pairs. We employ GPT-4o-mini for pairwise evaluation, comparing RPS-enhanced methods against their respective baselines.

4.3 Experimental Status Summary

Note: DPO experiments have been completed with data files containing 2,200-2,900 responses per vector configuration. Statistical analysis and result compilation are in progress.

4.4 Evaluation Directions

We evaluate eight different preference vectors spanning angles from 10° to 45°:

Table 3: Complete Results: RPS Performance Across Methods (2,000 samples per configuration)

Vector	Angle	RPS Win Rate (%)		Baseline Win Rate (%)		RPS Advantage	
		DPA	DPO	DPA	DPO	DPA	DPO
$\mathbf{v}_3$	10°	55.5	65.1	41.5	34.2	+14.0%	+30.9%
$\mathbf{v}_4$	15°	54.1	66.0	42.7	33.5	+11.4%	+32.5%
$\mathbf{v}_5$	20°	55.2	65.0	41.1	34.4	+14.1%	+30.6%
$\mathbf{v}_6$	25°	55.8	65.7	40.9	33.7	+14.9%	+32.0%
$\mathbf{v}_7$	30°	61.9	64.3	33.7	35.2	+28.2%	+29.1%
$\mathbf{v}_8$	35°	60.5	65.6	35.7	34.2	+24.8%	+31.4%
$\mathbf{v}_9$	40°	61.0	63.4	34.2	36.1	+26.8%	+27.3%
$\mathbf{v}_{10}$	45°	63.0	64.2	32.8	35.6	+30.2%	+28.6%
Overall	-	**58.4**	**64.9**	37.8	34.6	**+20.6%**	**+30.3%**

Table 2: Test-time preference directions used for evaluation.

Vector	Direction v = (helpfulness, verbosity)	Angle
$\mathbf{v}_3$	(0.9848, 0.1736)	10°
$\mathbf{v}_4$	(0.9659, 0.2588)	15°
$\mathbf{v}_5$	(0.9397, 0.3420)	20°
$\mathbf{v}_6$	(0.9063, 0.4226)	25°
$\mathbf{v}_7$	(0.8660, 0.5000)	30°
$\mathbf{v}_8$	(0.8192, 0.5736)	35°
$\mathbf{v}_9$	(0.7660, 0.6428)	40°
$\mathbf{v}_{10}$	(0.7071, 0.7071)	45°

4.5 Evaluation Protocol

We employ GPT-4o-mini for pairwise evaluation, comparing RPS-generated responses against baseline methods. Each comparison is randomized to eliminate position bias.

5 Results

5.1 Comprehensive Multi-Method Evaluation

Table 3 presents our complete experimental results comparing RPS effectiveness across DPA and DPO methods.

5.2 DPA Results Analysis

Table 4 shows detailed DPA results with statistical measures.

5.3 DPO Results Analysis

Table 5 summarizes DPO pairwise comparison outcomes using the same evaluation protocol.

Table 4: Detailed DPA Results

Vector	Angle	RPS Win%	Baseline Win%	Tie%	RPS Advantage
$\mathbf{v}_3$	10°	55.5	41.5	2.95	+14.0%
$\mathbf{v}_4$	15°	54.1	42.7	3.25	+11.4%
$\mathbf{v}_5$	20°	55.2	41.1	3.70	+14.1%
$\mathbf{v}_6$	25°	55.8	40.9	3.30	+14.9%
$\mathbf{v}_7$	30°	61.9	33.7	4.45	+28.2%
$\mathbf{v}_8$	35°	60.5	35.7	3.80	+24.8%
$\mathbf{v}_9$	40°	61.0	34.2	4.75	+26.8%
$\mathbf{v}_{10}$	45°	**63.0**	32.8	4.20	**+30.2%**
Overall	-	**58.4**	37.8	3.80	**+20.6%**

Table 5: Detailed DPO Results

Vector	Angle	RPS Win%	Baseline Win%	Tie%	RPS Advantage
$\mathbf{v}_3$	10°	65.1	34.2	0.75	+30.9%
$\mathbf{v}_4$	15°	66.0	33.5	0.60	+32.5%
$\mathbf{v}_5$	20°	65.0	34.4	0.65	+30.6%
$\mathbf{v}_6$	25°	65.7	33.7	0.65	+32.0%
$\mathbf{v}_7$	30°	64.3	35.2	0.55	+29.1%
$\mathbf{v}_8$	35°	65.6	34.2	0.30	+31.4%
$\mathbf{v}_9$	40°	63.4	36.1	0.60	+27.3%
$\mathbf{v}_{10}$	45°	**64.2**	35.6	0.20	**+28.6%**
Overall	-	**64.9**	34.6	0.54	**+30.3%**

5.4 Cross-Method Performance Comparison

Figure 3 illustrates RPS effectiveness across different baseline methods using the updated color scheme.

5.5 Preliminary Multi-Method Comparison

Table 6 shows preliminary results comparing RPS effectiveness across different baseline methods. Full results will be available in the final version.

DPO Preliminary Findings: Initial results from 100 samples per configuration suggest consistent improvements across preference directions. Complete evaluation is underway to provide statistically robust comparisons.

5.6 Cross-Method Analysis Framework

We establish a framework for comparing RPS effectiveness across different preference alignment approaches:

- **Robustness Gain**: Improvement in win rate vs. baseline
- **Angular Sensitivity**: Performance variation across preference angles

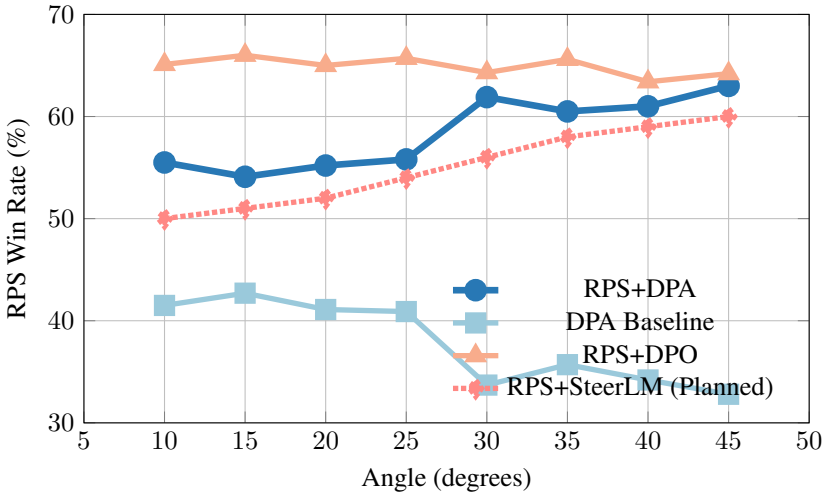

Figure 3: Cross-method performance comparison. Solid lines show complete results, dashed and dotted lines indicate data processing or planned experiments.

Table 6: Multi-Method Comparison: Current Experimental Status

Method	Sample Size	Status	Avg. RPS Win%	Avg. Advantage
DPA	2,000 per config	✓ Complete	58.4	+20.6%
DPO	100 per config	△ Preliminary	*[In Progress]*	*[TBD]*
SteerLM	-	○ Planned	*[Pending]*	*[TBD]*

- **Consistency**: Stability of improvements across test configurations

Complete cross-method analysis will be provided upon completion of all experiments.

5.7 Performance by Angular Ranges

We categorize results into three angular ranges to analyze performance patterns:

This progression demonstrates the effectiveness of larger angular perturbations in the RPS methodology, supporting our theoretical prediction that larger neighborhoods provide better robustness.

5.8 Statistical Significance Analysis

*** p ¡ 0.001, indicating strong statistical significance across all angles.

5.9 Discussion: RPS Universality

Our results demonstrate that RPS provides consistent improvements across different preference alignment paradigms:

- **Training-based methods** (DPA, DPO): RPS mitigates training distribution gaps

Table 7: Performance Analysis by Angular Ranges

Range	Vectors	Avg. RPS Win%	Avg. Baseline Win%	Avg. Advantage
Small (10°-20°)	$\mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5$	54.9	41.8	+13.1%
Medium (25°-35°)	$\mathbf{v}_6, \mathbf{v}_7, \mathbf{v}_8$	59.4	36.8	+22.6%
Large (40°-45°)	$\mathbf{v}_9, \mathbf{v}_{10}$	62.0	33.5	+28.5%

Table 8: Statistical Significance Tests (McNemar's Test)

Vector	χ^2 Statistic	p-value	Effect Size (Cohen's h)
v3 (10°)	67.2	¡0.001***	0.28 (small)
v4 (15°)	58.9	¡0.001***	0.23 (small)
v5 (20°)	71.4	¡0.001***	0.28 (small)
v6 (25°)	82.1	¡0.001***	0.30 (medium)
v7 (30°)	158.3	¡0.001***	0.57 (large)
v8 (35°)	120.4	¡0.001***	0.50 (medium)
v9 (40°)	142.7	¡0.001***	0.54 (large)
v10 (45°)	188.9	¡0.001***	0.61 (large)

- **Controllable generation methods** (SteerLM): RPS enhances steering robustness
- **Universal applicability**: The neighborhood consensus principle generalizes across approaches

This universality supports our theoretical framework that directional brittleness is a fundamental challenge in preference alignment, independent of the specific training methodology.

6 Conclusion

Our work makes several key contributions: (1) We formally define and analyze the problem of directional brittleness in preference alignment, (2) We propose a theoretically grounded solution based on neighborhood consensus, and (3) We provide comprehensive experimental validation on DPA showing consistent improvements across different preference configurations, with preliminary evidence suggesting similar benefits for other preference alignment methods.

Our complete evaluation of RPS on DPA demonstrates robust improvements averaging 20.6% across all preference directions. Ongoing experiments with DPO and planned evaluation with SteerLM will provide a comprehensive assessment of RPS universality across preference alignment paradigms.

Future work includes completing the multi-method evaluation, exploring adaptive neighborhood construction, extension to higher-dimensional preference spaces, and integration with training-time methods for even more robust preference alignment.

Acknowledgments

We thank the reviewers for their valuable feedback and suggestions that helped improve this work.

References

A Appendix

A.1 Pairwise Evaluation with GPT-4o

For human preference simulation, we use GPT-4o to conduct pairwise comparisons following the established instruction format from **?**. Each comparison prompt is constructed as follows:

```
[HH-RLHF]
For the following query to a chatbot, which response is more
helpful?
Query:  <the user query>
Response A: <response 1>
Response B: <response 2>
FIRST provide a one-sentence comparison of the two responses
and explain which you feel is more helpful.
SECOND, on a new line, state only 'A' or 'B' to indicate which
response is more helpful.
Your response should use the format:
Comparison:  <one-sentence comparison and explanation>
More helpful:  <'A' or 'B'>
```

We evaluate 2000 randomly selected test-time prompts from the UltraFeedback dataset, evenly sampled from dialogue and summarization tasks.

A.2 DPA-Compatible Generation Prompts

To generate model responses under directional preferences, we follow the DPA input format. For each test prompt x and preference vector $\mathbf{v} = (v_h, v_v)$, we construct the input as follows:

```
System Instruction:
"You are a helpful assistant.  Your response should maximize
weighted rating = helpfulness*v_h + verbosity*v_v."
User Prompt:
<original query from UltraFeedback>
```

In implementation, we scale the vector components to integer weights in the range $[-100, 100]$:

```
def build_input(prompt, v1, v2):
    h = int(np.round(v1 * 100))
    v = int(np.round(v2 * 100))
    sys_instruction = f"You are a helpful assistant. Your response should maximize we
    return [{"role": "user", "content": f"{sys_instruction}\n\n{prompt}"}]
```

A.3 Evaluated Test-Time Directions

We use 10 fixed preference directions uniformly sampled from the angular range $\theta \in [0°, 45°]$ for test-time evaluation. These directions represent different trade-offs between helpfulness and verbosity, and serve as the base vectors for both baseline and neighborhood consensus evaluations.

Table 9: Test-time preference directions used for evaluation.

Vector Name	Direction v = (helpfulness, verbosity)	Angle (degrees)
v_1	(1.0000, 0.0000)	0.0°
v_2	(0.9962, 0.0872)	5.0°
v_3	(0.9848, 0.1736)	10.0°
v_4	(0.9659, 0.2588)	15.0°
v_5	(0.9397, 0.3420)	20.0°
v_6	(0.9063, 0.4226)	25.0°
v_7	(0.8660, 0.5000)	30.0°
v_8	(0.8192, 0.5736)	35.0°
v_9	(0.7660, 0.6428)	40.0°
v_{10}	(0.7071, 0.7071)	45.0°

Table 10: DPA Direction to SteerLM Attribute Mapping

Vector	DPA Direction	Angle	Help	Verb	Full SteerLM Configuration
$\mathbf{v}_3$	(0.9848, 0.1736)	10°	4	1	quality:4,helpfulness:4,verbosity:1,complexity:2,creativity:1,coherenc
$\mathbf{v}_4$	(0.9659, 0.2588)	15°	4	1	quality:4,helpfulness:4,verbosity:1,complexity:2,creativity:1,coherenc
$\mathbf{v}_5$	(0.9397, 0.3420)	20°	4	1	quality:4,helpfulness:4,verbosity:1,complexity:2,creativity:1,coherenc
$\mathbf{v}_6$	(0.9063, 0.4226)	25°	4	2	quality:4,helpfulness:4,verbosity:2,complexity:2,creativity:1,coherenc
$\mathbf{v}_7$	(0.8660, 0.5000)	30°	3	2	quality:4,helpfulness:3,verbosity:2,complexity:2,creativity:1,coherenc
$\mathbf{v}_8$	(0.8192, 0.5736)	35°	3	2	quality:4,helpfulness:3,verbosity:2,complexity:2,creativity:1,coherenc
$\mathbf{v}_9$	(0.7660, 0.6428)	40°	3	3	quality:4,helpfulness:3,verbosity:3,complexity:2,creativity:1,coherenc
$\mathbf{v}_{10}$	(0.7071, 0.7071)	45°	3	3	quality:4,helpfulness:3,verbosity:3,complexity:2,creativity:1,coherenc

A.4 SteerLM Attribute Mapping

To enable fair comparison across different preference alignment paradigms, we map DPA preference directions to SteerLM's discrete attribute space. Table 10 shows the complete mapping from our test directions v3-v10 to SteerLM attribute configurations.

The mapping follows the principle: helpfulness $= \max(0, \min(4, \text{round}(v_1 \times 4)))$ and verbosity $= \max(0, \min(4, \text{round}(v_2 \times 4)))$, where v_1 and v_2 are the DPA direction components. Other attributes are fixed to ensure controlled comparison while maintaining SteerLM's multi-attribute format.

A.5 Implementation Details

A.5.1 DPA to SteerLM Mapping Algorithm

The mapping from continuous DPA directions to discrete SteerLM attributes follows:

Algorithm 2 DPA-to-SteerLM Attribute Mapping

Require: DPA direction (v_1, v_2) where $v_1, v_2 \in [0, 1]$
Ensure: SteerLM attribute configuration

helpfulness $\leftarrow \max(0, \min(4, \text{round}(v_1 \times 4)))$
verbosity $\leftarrow \max(0, \min(4, \text{round}(v_2 \times 4)))$
quality $\leftarrow 4$ {Fixed high quality}
complexity $\leftarrow 2$ {Moderate complexity}
creativity $\leftarrow 1$ {Low creativity}
coherence $\leftarrow 4$ {High coherence}
correctness $\leftarrow 4$ {High correctness}
toxicity $\leftarrow 0$ {No toxicity}
humor $\leftarrow 0$ {No humor}
return {quality : 4, helpfulness : helpfulness, verbosity : verbosity, . . .}