SAMPLE-EFFICIENT PERSONALIZATION OF LARGE LANGUAGE MODEL ALIGNMENT WITH MULTIDIMENSIONAL USER PREFERENCE DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Preference alignment has become a central tool to align *large language models* (LLMs) with human values and desired behavior, and its online variant can further advance their capabilities beyond what static data alone offer by repeatedly generating and labeling new responses with human feedback. But *costly*, *time-consuming* annotations limit its scalability, making sample efficiency essential, and existing methods typically seek to improve it by expanding coverage of the response space or selecting pairs with the highest uncertainty for comparison. Although they are effective, these methods quantify uncertainty from the perspective of the LLM, which may differ from the uncertainty perceived by the user: a pair can appear highly uncertain to the LLM even when past comparisons from the user already imply a clear ordering. In this paper, we propose *multidimensional preference-guided online direct preference optimization* (MPG-DPO), which leverages low-cost multidimensional evaluations from automated verifiers on the user historical comparisons to construct a personalized preference cone guiding response selection for online preference alignment. Specifically, it first discards responses that are confidently dominated under the estimated user preference, then selects two responses that are competitive under the retained preference cone yet exhibit the strongest unresolved disagreement. Theoretical analysis shows that, with high probability, the fraction of queries spent on cone-dominated responses vanishes as learning progresses. Experiments across multiple LLM families show that MPG-DPO outperforms strong contenders under matched query budgets.

1 INTRODUCTION

Preference alignment has achieved remarkable success in aligning *large language models* (LLMs) with human preferences and desired behaviors (Christiano et al., 2017; Ouyang et al., 2022). To continually push the boundaries of model capabilities, preference alignment is typically extended to an *online* process, where models repeatedly generate new responses for human evaluation (Guo et al., 2024; Calandriello et al., 2024; Xie et al., 2025). However, the high cost and long turnaround of the annotations required pose a major barrier to scaling this continuous feedback loop. Therefore, improving sample efficiency in online preference alignment is of critical importance.

Existing approaches can be broadly divided into two categories. The first focuses on online exploration to improve policy coverage, increasing the sampling probability of responses that are poorly covered by previous data (Xie et al., 2025; Cen et al., 2025; Chen et al., 2025). The second line of work relies on active preference alignment, which first generates a pool of candidate pairs and then selects the most informative pair for comparison in each interaction (Das et al., 2025; Lin et al., 2026; Belakaria et al., 2025). These active methods typically rank candidates using information criteria measured from the model perspective, such as reward uncertainty or gradient-based metrics.

While effective at optimizing response generation or pair selection, these methods mainly rely on *model-perspective* information, often querying feedback even when past comparisons made by the user already settle the ordering within a pair. For example, in code generation, previous choices may show that the user is willing to trade some clarity for a sufficiently large gain in accuracy. A new pair may still exhibit high reward uncertainty even when one response offers too little accuracy gain

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

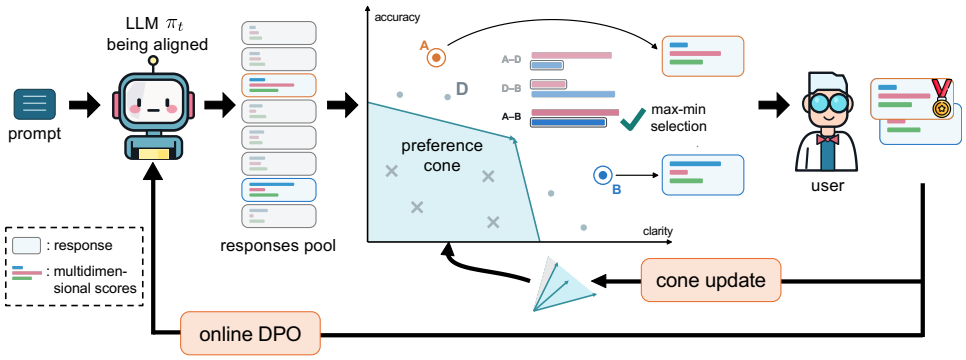


Figure 1: Summary of the proposed MPG-DPO **approach.** Given an input prompt, candidate responses are sampled and filtered using the preference cone, after which a pair is selected to maximize the weaker of two opposing preference margins. The selected responses are evaluated and used to update the cone and policy.

to justify its loss of clarity, making the comparison carry little *user-perspective* information despite appearing uncertain to the model. In a related direction, constitutional AI reduces repeated human supervision via high-level guiding principles (Bai et al., 2022; Huang et al., 2024), but these principles are *hand-written* and shared across users. To capture personalized trade-offs, specific dimensions like accuracy and clarity can fortunately be evaluated automatically, allowing them to naturally serve as low-cost multidimensional scores. This raises a question: *How can low-cost multidimensional scores be leveraged to capture personalized preferences and improve sample efficiency?*

Using such multidimensional information is non-trivial because the scores are *heterogeneous* and can hardly be directly aggregated. The insightful AMPLe (Oh et al., 2025) uses multi-objective rewards through Bayesian scalarization, but its uncertainty-driven queries may compare responses already dominated by other alternatives. We instead treat each historical comparison as revealing a *preferred direction* of the user in the sub-objective space, and these directions form a personalized preference *Pareto cone* (Censor, 1977; Rame et al., 2023; Ma et al., 2026). The key challenge is to estimate this cone from historical data and use it to identify response pairs whose preference remains unresolved.

In this paper, we use low-cost multidimensional evaluations to improve sample efficiency in online preference alignment. Each response is scored on attributes such as correctness and generation cost by low-cost or task-specific verifiers that are substantially cheaper than human evaluators. We construct a personalized preference cone from historical user comparisons and leverage its geometry to distinguish comparisons that are already resolved from those that represent genuinely unresolved trade-offs. We then select two competitive responses that *exhibit the strongest unresolved disagreement among the retained preference directions*. This preserves the exploration of potentially valuable responses while directing costly feedback toward comparisons that provide new information.

We propose *multidimensional preference-guided online direct preference optimization* (MPG-DPO), which uses the user preference cone to guide response selection at each round of online DPO. Specifically, at each round, the current policy generates a pool of on-policy responses, and a ridge-regression estimator predicts their multidimensional score vectors together with confidence bounds. We eliminate responses that are confidently dominated by another response in the sampled pool and select a pair for which different preference directions favor different responses. After obtaining the preference label, we update the preference cone and the policy. Theoretical analysis shows that, with high probability, only a slowly growing number of selections are cone-dominated, so wasted queries vanish as learning progresses. Experiments show the effectiveness of the proposed method.

2 Preliminaries

In this section, we introduce the preliminaries and notations of our proposed algorithm.

2.1 Online Preference Alignment

In this part, we introduce the notations for online preference alignment. Let Π be a class of LLM policies, where each $\pi \in \Pi$ maps a prompt $\mathbf{x} \in \mathcal{X}$ to a distribution $\pi(\cdot \mid \mathbf{x})$ over the response space

$\mathcal{Y}$. At each round t, a prompt $\mathbf{x}_t \sim \rho$ is drawn from a distribution ρ over $\mathcal{X}$. Given $\mathbf{x}_t$, we sample two responses $y_t \sim \pi_t^{(1)}(\cdot \mid \mathbf{x}_t)$ and $y_t' \sim \pi_t^{(2)}(\cdot \mid \mathbf{x}_t)$, where $\pi_t^{(1)}, \pi_t^{(2)} \in \Pi$ are the sampling policies used at round t. A preference oracle then returns a label $o_t \in \{-1, +1\}$, where $o_t = +1$ indicates that y_t is preferred to y_t', denoted by $y_t \succ y_t'$, and $o_t = -1$ indicates the reverse preference.

For any policy $\pi \in \Pi$, we define its KL-regularized value as

$$J_\beta(\pi) = \mathbb{E}_{\mathbf{x}\sim\rho,\, y\sim\pi(\cdot|\mathbf{x})}\left[r^*(\mathbf{x}, y) - \beta \log \frac{\pi(y \mid \mathbf{x})}{\pi_{\text{ref}}(y \mid \mathbf{x})}\right],$$

where r^* is the reward function, π_{ref} is a fixed reference policy and $\beta > 0$ is the KL-regularization coefficient. Let $\pi_\beta^* \in \arg\max_{\pi\in\Pi} J_\beta(\pi)$ denote an optimal KL-regularized policy.

The objective J_β depends on the unknown reward function, whereas the available feedback consists of preference labels. *Direct preference optimization* (DPO) instead learns directly from preference pairs without fitting a separate reward model (Rafailov et al., 2023). We consider the online DPO setting (Calandriello et al., 2024; Guo et al., 2024), in which preference pairs are collected during the iterative training process. Given the comparison $(\mathbf{x}_t, y_t, y_t')$ and its observed preference label o_t at round t, we perform the stochastic-gradient update

$$\pi_{t+1} = \pi_t - \eta_t \, \nabla_\pi \left[-\log \sigma\left(\beta \, o_t \left[\log \frac{\pi(y_t \mid \mathbf{x}_t)}{\pi_{\text{ref}}(y_t \mid \mathbf{x}_t)} - \log \frac{\pi(y_t' \mid \mathbf{x}_t)}{\pi_{\text{ref}}(y_t' \mid \mathbf{x}_t)}\right]\right)\right]\Bigg|_{\pi=\pi_t}, \qquad (1)$$

where $\eta_t > 0$ is the learning rate.

2.2 Preference Cone

In this part, we introduce the notations of user preference cone.

Definition 1 (Preference Matrix and Preference Cone) *Let K denote the number of preference directions and d the number of sub-objectives. Given a preference matrix $\mathbf{W} \in \mathbb{R}^{K\times d}$, the associated preference cone is defined as*

$$C = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{W}\mathbf{x} \geq \mathbf{0}\}.$$

Definition 2 (Pareto C-Optimality) *For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we say that $\mathbf{x}$ strictly C-dominates $\mathbf{y}$, denoted by $\mathbf{x} \succ_C \mathbf{y}$, if $\mathbf{x} - \mathbf{y} \in C$ and $\mathbf{W}(\mathbf{x} - \mathbf{y}) \neq \mathbf{0}$. An action is said to be* Pareto C-optimal *if no other action strictly C-dominates it. We define the set of Pareto C-optimal actions as*

$$P_C^* = \{i \in \{1, \ldots, N\} \mid \nexists\, j \in \{1, \ldots, N\} \setminus \{i\} : \; j \succ_C i\}.$$

Here, N is the number of actions. For notational convenience, for any two actions $i, j \in \{1, \ldots, N\}$, we write $i \succ_C j$ as shorthand for $\mathbf{x}_i \succ_C \mathbf{x}_j$.

We assume that, at every round, each non-optimal response has an optimal alternative with a uniformly positive advantage across all preference directions available before the query, while allowing ties in other comparisons. This condition excludes non-optimal responses with only boundary dominators.

Assumption 1 (Uniform Positive Preference Gap) *Let $\mathbf{w}_{t-1,k}^\top$ denote the k-th row of $\mathbf{W}_{t-1}$ available before query t, and let $\mathbf{x}_{t,i}$ denote the latent score vector of action i in the current candidate pool. All optimal sets in this assumption are evaluated on this pool using C_{t-1}. For each $i \notin P_{C_{t-1}}^*$, let $\mathcal{D}_t(i) = \{j \in P_{C_{t-1}}^* : j \succ_{C_{t-1}} i\}$ be nonempty and define its* preference gap *as*

$$\Delta_{t,i} = \max_{j\in\mathcal{D}_t(i)} \min_{k\in\{1,\ldots,K_{t-1}\}} \frac{\mathbf{w}_{t-1,k}^\top(\mathbf{x}_{t,j} - \mathbf{x}_{t,i})}{U_{t-1,k} - L_{t-1,k}},$$

where $L_{t-1,k} = \sum_{q=1}^d \min(0, \mathbf{w}_{t-1,k}[q])$, $U_{t-1,k} = \sum_{q=1}^d \max(0, \mathbf{w}_{t-1,k}[q])$, and $[q]$ indexes coordinates. We assume $\Delta_{t,i} \geq \Delta > 0$ uniformly over all rounds t and actions $i \notin P_{C_{t-1}}^*$.

3 Our Approach

In this section, we present *multidimensional preference-guided online direct preference optimization* (MPG-DPO). We first formulate the learning problem and then present our algorithm, which constructs a user preference cone from historical data to guide response pair selection for sample efficiency. Finally, we analyze the theoretical property of the proposed algorithm.

3.1 Problem Formulation

In this part, we formulate the learning problem.

Multidimensional score. We assume that each prompt–response pair $(\mathbf{x}, y)$ has an underlying score vector $\mu(\mathbf{x}, y) = \theta^{*\top}\phi(\mathbf{x}, y) \in [0,1]^d$. The observed multidimensional score is

$$\mathbf{z}_t(\mathbf{x}, y) = \theta^{*\top}\phi(\mathbf{x}, y) + \varepsilon_t(\mathbf{x}, y), \qquad \|\phi(\mathbf{x}, y)\|_2 \le 1, \qquad \|\theta^*\|_F \le S,$$

where $\phi(\mathbf{x}, y) \in \mathbb{R}^p$ is the prompt–response feature vector, $\theta^* \in \mathbb{R}^{p\times d}$ is unknown and fixed across rounds, and $\varepsilon_t(\mathbf{x}, y) \in \mathbb{R}^d$ is a conditionally zero-mean, σ_z-sub-Gaussian noise vector, where $\sigma_z \ge 0$. We omit the prompt argument in ϕ, μ, $\mathbf{z}_t$, and ε_t when it is clear from the context.

Learning protocol and performance measure. In the online preference alignment setting, the learner observes multidimensional scores for the queried responses and receives pairwise preference feedback at each round. At each round t, the learner proceeds as follows:

1. Sample N candidate responses from $\pi_t(\cdot \mid \mathbf{x}_t)$ and select a pair of indices $B_t = \{i_t, j_t\}$.
2. Observe the multidimensional score vectors of the selected responses and their preference label.
3. Update the estimated preference cone and the LLM policy.

The selection procedure is evaluated by whether a designated response from the selected pair belongs to the optimal set induced by the preference cone. Specifically, for any index $a_t \in B_t$, we define the cumulative indicator regret as

$$\mathrm{Regret}^{01}(T) = \sum_{t=1}^{T} \mathbb{I}\{a_t \notin P^*_{C_{t-1}}\}.$$

3.2 Preference Cone Construction

In this part, we construct the preference cone from historical comparisons. Each comparison reveals a user preference through the multidimensional score difference from the less preferred response to the preferred one. This difference defines a *preference direction*, and non-negative linear combinations of these directions form the personalized preference cone. Since the feedback may be noisy, we screen new directions against existing preferences and retain only those that pass the consistency check.

Each comparison (y_t, y'_t) with label o_t yields a preference direction by orienting the multidimensional score difference toward the preference label. Since scores and preference labels may be noisy, we use the previous preference matrix to discard a direction only when every row supports the opposite preference by a margin exceeding a threshold $\tau \ge 0$. The retained directions generate the cone

$$C_t = \mathrm{cone}\left\{ o_s \frac{\mathbf{d}_s}{\|\mathbf{d}_s\|_2} : 1 \le s \le t,\ \min_{1\le k\le K_{s-1}} \mathbf{w}_{s-1,k}^\top \left(-o_s \frac{\mathbf{d}_s}{\|\mathbf{d}_s\|_2} \right) \le \tau \right\}, \tag{2}$$

where $\mathbf{d}_s = \mathbf{z}_s(y_s) - \mathbf{z}_s(y'_s)$ is the observed multidimensional score difference, and cone denotes the set of all nonnegative linear combinations of the retained preference directions.

Given C_t, we obtain the preference matrix $\mathbf{W}_t$ using the double-description method (Motzkin et al., 1953; Fukuda & Prodon, 1995)[1]. Specifically, we compute unit-norm generators of the vectors having nonnegative inner products with every point in C_t and use their transposes as the rows of $\mathbf{W}_t$:

$$\left\{ \sum_{k=1}^{K_t} \lambda_k \mathbf{w}_{t,k} : \lambda_k \ge 0 \right\} = \left\{ \mathbf{w} \in \mathbb{R}^d : \mathbf{w}^\top \mathbf{x} \ge 0,\ \forall \mathbf{x} \in C_t \right\}. \tag{3}$$

The resulting matrix gives $C_t = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{W}_t\mathbf{x} \ge \mathbf{0}\}$. For computational efficiency, our implementation limits the number of retained rules, while the theoretical analysis uses the full cone.

3.3 Response Pair Selection

In this part, we use the current preference cone to guide response pair selection. The core idea is to *select a pair of responses such that neither is C-dominated by any other candidate, while their comparison reveals a trade-off unresolved by the current preference rules*. Each new query thus contributes a candidate preference direction in the multidimensional score space.

[1] A well-established implementation of the method is available through the Python package `pycddlib`.

Algorithm 1 Multidimensional Preference-Guided Online Direct Preference Optimization

1: Receive $\mathbf{x}_t$ and sample $y_t^i \sim \pi_t(\cdot \mid \mathbf{x}_t)$ for $i \in [N]$.
2: Fit ridge regression to past noisy scores and compute confidence bounds using $\mathbf{W}_{t-1}$ via (4).
3: Form $\mathcal{N}_t$ using (5) and select (i_t, j_t) using (6).
4: Set $y_t = y_t^{i_t}$ and $y_t' = y_t^{j_t}$, observe $\mathbf{z}_t(y_t), \mathbf{z}_t(y_t')$ and query the preference label o_t.
5: Compute $\mathbf{d}_t = \mathbf{z}_t(y_t) - \mathbf{z}_t(y_t')$ and construct C_t using (2).
6: Obtain $\mathbf{W}_t$ using (3) and update the score predictor with the observed scores.
7: Update π_{t+1} using $(\mathbf{x}_t, y_t, y_t', o_t)$ via (1).

Estimating multidimensional score confidence bounds. We estimate multidimensional scores by fitting a ridge regression model to previously observed noisy response scores:

$$V_t = \lambda I_p + \sum_{s<t} \sum_{i \in B_s} \phi(y_s^i)\phi(y_s^i)^\top, \qquad \widehat{\theta}_t = V_t^{-1} \sum_{s<t} \sum_{i \in B_s} \phi(y_s^i)\mathbf{z}_s(y_s^i)^\top.$$

Here, $\lambda > 0$ is the regularization parameter.

For any two responses y, y' to $\mathbf{x}_t$, the predicted normalized improvement along a unit-norm row $\mathbf{w}_{t-1,k}^\top$ of $\mathbf{W}_{t-1}$ and the associated confidence width are

$$\widehat{m}_{t,k}(y \mid y') = \frac{\mathbf{w}_{t-1,k}^\top \widehat{\theta}_t^\top [\phi(y) - \phi(y')]}{U_{t-1,k} - L_{t-1,k}}, \qquad c_{t,k}(y, y') = \frac{\alpha_t(\delta)\|\phi(y) - \phi(y')\|_{V_t^{-1}}}{U_{t-1,k} - L_{t-1,k}},$$

where $\alpha_t(\delta) = \sqrt{\lambda} S + \sigma_z \sqrt{d \left(2 \log \frac{d}{\delta} + \log \frac{\det V_t}{\det(\lambda I_p)}\right)}$.

Let $m_{t,k}(y \mid y')$ denote the corresponding latent margin, obtained by replacing $\widehat{\theta}_t$ with θ^*. With probability at least $1 - \delta$, $|m_{t,k}(y \mid y') - \widehat{m}_{t,k}(y \mid y')| \le c_{t,k}(y, y')$ simultaneously over rounds (See Lemma 1 in Appendix C). The resulting confidence bounds are

$$\mathrm{UCB}_{t,k}(y \mid y') = \widehat{m}_{t,k}(y \mid y') + c_{t,k}(y, y'), \qquad \mathrm{LCB}_{t,k}(y \mid y') = \widehat{m}_{t,k}(y \mid y') - c_{t,k}(y, y'). \tag{4}$$

On this event, a positive lower bound certifies an improvement from y' to y along direction $\mathbf{w}_{t-1,k}$.

Removing C-dominated responses. In the following, we show how to select the two responses. Firstly, a response is removed if another response in the pool confidently C_{t-1}-dominates it, so that we can prevent the human from comparing a response that is already known to be inferior to another available response. The retained indices are

$$\mathcal{N}_t = \left\{ i \in [N] : \nexists\, j \in [N] \setminus \{i\} \text{ such that } \min_{k \in [K_{t-1}]} \mathrm{LCB}_{t,k}(y_t^j \mid y_t^i) > 0 \right\}. \tag{5}$$

For a fixed pair (y_t^j, y_t^i), Eq. (5) evaluates the least favorable rule for y_t^j improving upon y_t^i. A positive value means that all rows of $\mathbf{W}_{t-1}$ support this ordering with positive lower confidence bounds, allowing y_t^i to be discarded without a human comparison.

Response pair selection. After filtering out confidently dominated responses under the preference cone, we select a pair from $\mathcal{N}_t$ that maximizes the weaker of the two opposing preference margins:

$$(i_t, j_t) \in \operatorname*{arg\,max}_{i,j \in \mathcal{N}_t,\ i \neq j} \min \left\{ \max_{k \in [K_{t-1}]} \mathrm{LCB}_{t,k}(y_t^i \mid y_t^j), - \min_{k \in [K_{t-1}]} \mathrm{UCB}_{t,k}(y_t^i \mid y_t^j) \right\}. \tag{6}$$

The two terms are positive when some rule confidently favors y_t^i and y_t^j, respectively. A positive minimum indicates conflicting preferences across rules and measures the weaker side of the disagreement. Maximizing it favors pairs with strong evidence on both sides of the trade-off. The algorithm is summarized in Alg. 1. In implementation, we resample when fewer than two candidates survive, skip cone updates for zero score differences, and retain the previous cone if its update yields a trivial dual.

3.4 Theoretical Analysis

In this part, we analyze the theoretical property of the proposed algorithm. We note that our theoretical analysis provides a supporting explanation of the filtering mechanism by bounding the frequency of

cone-dominated selections under simplifying assumptions. The empirical evaluation further examines the benefits of adaptive cone construction and trade-off selection for downstream performance.

We introduce the following assumption that requires the queried pair to provide sufficient information about a comparison that establishes the non-optimality of each selected response.

Assumption 2 (Coverage of a dominating response) *Let $q_t = \sum_{i \in B_t} \|\phi(y_t^i)\|^2_{V_t^{-1}}$ denote the aggregate feature uncertainty of the queried pair. There exists a constant $\kappa > 0$, independent of t, such that for every selected $i \in B_t \setminus P^*_{C_{t-1}}$, some $j \in \mathcal{D}_t(i)$ attaining the maximum defining $\Delta_{t,i}$ satisfies*

$$\|\phi(y_t^j) - \phi(y_t^i)\|^2_{V_t^{-1}} \leq \kappa q_t. \tag{7}$$

We bound regret by controlling the score-estimation uncertainty of responses dominated under C_{t-1}.

Theorem 1 *Under Assumptions 1 and 2, let $\lambda \geq 1$ and $\delta \in (0,1)$. With probability at least $1-\delta$,* MPG-DPO *satisfies, simultaneously for all generated query prefixes of length $T \geq 1$ and any choices $a_t \in B_t$,*

$$\mathrm{Regret}^{01}(T) \leq \frac{12\bar{\alpha}_T^2(\delta)\kappa p}{\Delta^2} \log\left(1 + \frac{2T}{\lambda p}\right), \tag{8}$$

where $\bar{\alpha}_T(\delta) = \sqrt{\lambda}S + \sigma_z\sqrt{d\left(2\log\frac{d}{\delta} + p\log\left(1+\frac{2T}{\lambda p}\right)\right)}$.

Remark 1 *Unlike the fixed-cone setting of Ma et al. (2026), our preference cone is estimated online from noisy pairwise comparisons and updated after every query, and is used to actively filter dominated responses and select the next query pair for online DPO. Selection and evaluation use the same cone C_{t-1}, so a dominated response can survive filtering only if its score comparison remains uncertain. The simultaneous confidence bound and coverage assumption control these selections without a separate term for cone updates.*

Theorem 1 shows that, if queries continue indefinitely, the average regret vanishes as $T \to \infty$ with probability at least $1-\delta$. The full proof is in Appendix C, which also shows that the round-level regret, counting a round once whenever at least one selected response is non-optimal under C_{t-1}, satisfies the same bound; summing the indicator over both responses instead yields twice this bound.

Proposition 1 (Avoidance of cone-dominated queries) *Under the conditions of Theorem 1, with fixed problem parameters and δ, for every generated prefix of T expensive queries, at least $T - O(\log^2 T)$ compare two Pareto-optimal responses in their respective candidate pools under the cones available before the queries, with probability at least $1-\delta$. If queries continue indefinitely, the fraction allocated to such non-dominated pairs approaches one.*

4 Experiments

In this section, we first introduce the experimental setup and then present the experimental results. Our experiments aim to answer the following three questions: **Q1**: does the proposed method enhance the policy across different LLM families; **Q2**: does the proposed preference-guided sampling make better use of the multidimensional information than other methods with additional access to it; and **Q3**: how sensitive is the proposed method to its key parameters.

4.1 Experimental Setup

Prompts. We use the prompt-only split of `ultrafeedback-prompt` for online training, following the generation and training pipeline of the official XPO implementation in `trl`[2]. All methods share this pipeline. For MPG-DPO, we initialize the preference matrix using comparisons collected from the preference oracle on `ultrafeedback_binarized`.

Models. We evaluate three model families: `Qwen2.5-7B-Instruct`, `Mistral-7B-Instruct-v0.3`, and `OLMo-2-1124-7B-Instruct`. Each reference policy is initialized from the corresponding policy

[2] `https://huggingface.co/docs/trl/xpo_trainer`

Table 1: Main results across three 7B policies under the same preference-query budget. Higher is better for every metric. Each entry reports mean ± standard deviation over three seeds.

LLM	Method	IMDB Average Reward	TL;DR Win Rate (%)	AlpacaEval2 LC Win Rate (%)	Arena-Hard Win Rate (%)
Qwen2.5-7B	Online DPO	81.6± 0.55	63.38± 0.24	59.03± 0.18	54.68± 0.17
	XPO	81.8± 0.66	63.82± 0.27	59.14± 0.20	54.74± 0.24
	APL	82.1± 0.61	64.02± 0.26	59.13± 0.22	54.94± 0.21
	APL+MD	82.3± 0.63	64.34± 0.26	59.41± 0.27	55.08± 0.23
	MPG-DPO	**83.2**± 0.64	**65.31**± 0.29	**60.71**± 0.22	**56.04**± 0.23
Mistral-7B	Online DPO	77.4± 0.54	61.76± 0.22	56.76± 0.18	53.44± 0.18
	XPO	77.8± 0.57	61.87± 0.24	56.81± 0.20	53.52± 0.24
	APL	77.9± 0.53	61.89± 0.25	56.88± 0.17	53.47± 0.20
	APL+MD	78.6± 0.57	62.74± 0.24	57.19± 0.20	53.78± 0.24
	MPG-DPO	**79.9**± 0.59	**63.63**± 0.26	**58.28**± 0.22	**54.52**± 0.27
OLMo-2-7B	Online DPO	77.2± 0.49	61.64± 0.22	56.53± 0.19	53.14± 0.21
	XPO	78.1± 0.54	62.67± 0.28	56.49± 0.21	53.28± 0.22
	APL	77.2± 0.52	61.58± 0.25	56.44± 0.18	53.20± 0.23
	APL+MD	78.0± 0.58	62.44± 0.27	56.88± 0.21	53.41± 0.26
	MPG-DPO	**78.8**± 0.58	**63.31**± 0.26	**57.73**± 0.20	**54.18**± 0.22

checkpoint and remains frozen throughout training. Unless otherwise specified, MPG-DPO samples $N = 8$ candidate responses per prompt from the current policy. All methods use the same prompt stream, maximum response length, decoding parameters, optimizer, and preference-query budget. We train with LoRA adapters (Hu et al., 2022) using rank $r = 32$ and scaling parameter $\alpha = 64$, and report results over three random seeds. ϕ is meanwhile implemented as an independent feature extractor, which we fix to use an 86M parameter frozen `microsoft/deberta-v3-base`[3] model on the prompt-response pair.

Datasets and evaluation metrics. We report four metrics: (i) average reward from a held-out reward model on the IMDB sentiment generation task (Maas et al., 2011), (ii) summarization win rate against human-written summaries on TL;DR (Völske et al., 2017), (iii) length-controlled win rate on AlpacaEval2 (Dubois et al., 2024), and (iv) win rate on Arena-Hard (Li et al., 2025).

Preference oracle. We use `Kimi-K3` (Team et al., 2026) to simulate human preference feedback. Each query uses a fixed judging template containing the input prompt and two candidate responses, whose order is randomized independently for each query. The oracle identifies the preferred response, yielding a binary preference label. Each oracle call counts as one preference query. Preference labeling and sub-objective scoring are performed by separate models.

Sub-objective oracle. We use five fixed evaluation dimensions adapted from `FLASK` (Ye et al., 2024): correctness, completeness, clarity, conciseness, and reasoning quality. The corresponding scoring rubrics are generated by prompting the preference oracle. A frozen `Prometheus2` evaluator (`prometheus-7b-v2.0`) (Kim et al., 2024) applies these rubrics to assign an integer score from 1 to 5 for each dimension. The scores are normalized to $[0, 1]$ to obtain $\mathbf{z}_t(y)$.

Contenders. We compare all methods under the same preference-query budget. We compared our approach against state-of-the-art contenders with official code implementations. Online DPO samples an on-policy response pair and applies the standard DPO update. XPO incorporates the exploratory objective of Xie et al. (2025), while APL performs active preference selection following Oh et al. (2026). APL+MD modifies the APL selection algorithm by using a weighted average of the reward margin and the sum of squared ridge-prediction confidence widths between pairs of responses.

4.2 Main Results

Table 1 compares MPG-DPO with state-of-the-art baselines across three LLM families under the same preference-query budget. The consistent improvements across model families support the effectiveness of preference-guided pair selection.

We also assess the contribution of each component in MPG-DPO through the following ablations:

[3] `https://huggingface.co/microsoft/deberta-v3-base`

Table 2: Component ablations on `Qwen2.5-7B-Instruct` under a fixed preference-query budget. Each entry reports mean ± standard deviation over three independent training seeds.

Method Variant	IMDB Average Reward	TL;DR Win Rate (%)	ALPACAEVAL2 LC Win Rate (%)	ARENA-HARD Win Rate (%)
Without cone filtering	81.9± 0.62	64.04± 0.25	58.88± 0.23	54.74± 0.22
Without trade-off selection	82.1± 0.54	64.12± 0.24	59.02± 0.19	54.82± 0.26
Fixed Pareto cone	82.2± 0.59	64.43± 0.26	59.23± 0.31	55.02± 0.28
Full method	**83.2**± 0.64	**65.31**± 0.29	**60.71**± 0.22	**56.04**± 0.23

Table 3: Comparison of multidimensional feedback utilization across two policy models under matched scoring and preference-query budgets, with $N = 8$.

LLM	Method	IMDB Average Reward	TL;DR Win Rate (%)	ALPACAEVAL2 LC Win Rate (%)	ARENA-HARD Win Rate (%)
`Qwen2.5-7B`	Scalar-Margin	81.7± 0.53	63.48± 0.27	58.67± 0.24	54.74± 0.21
	APL+MD-Score	81.9± 0.57	63.87± 0.23	59.02± 0.26	54.82± 0.26
	APL+MD-Scalar	82.1± 0.59	64.11± 0.23	59.21± 0.23	54.98± 0.20
	MPG-DPO	**83.2**± 0.64	**65.31**± 0.29	**60.71**± 0.22	**56.04**± 0.23
`Mistral-7B`	Scalar-Margin	77.6± 0.57	61.83± 0.22	56.79± 0.17	53.14± 0.22
	APL+MD-Score	77.7± 0.54	61.97± 0.27	57.01± 0.23	53.31± 0.22
	APL+MD-Scalar	78.3± 0.54	62.26± 0.27	57.11± 0.20	53.51± 0.25
	MPG-DPO	**79.9**± 0.59	**63.63**± 0.26	**58.28**± 0.22	**54.52**± 0.27

1. Without cone filtering: We omit candidate removal and apply the trade-off selection criterion to all pairs in the full candidate pool. This variant assesses the benefit of the filtering mechanism.
2. Without trade-off selection: We retain cone filtering but select a pair uniformly at random from $\mathcal{N}_t$. This variant assesses the benefit of trade-off selection beyond filtering alone.
3. Fixed Pareto cone: We replace the learned cone with the positive orthant by setting $\mathbf{W}_t = I_5$ throughout training, while retaining the filtering and trade-off selection procedures. The score predictor and policy continue to update. This variant compares learned preference trade-offs with a fixed coordinate-wise preference relation.

The results in Table 2 show that the full method outperforms each ablated variant, supporting the contribution of each component of the proposed algorithm. With Table 1, these findings answer **Q1**.

4.3 COMPARISON WITH ALTERNATIVE USES OF MULTIDIMENSIONAL INFORMATION

We compare MPG-DPO with alternative pair-selection strategies that use the same multidimensional feedback. Feedback budgets and training protocol are fixed, with only the selection criterion varied.

We construct three information-enhanced contenders:

1. Scalar-Margin selects the pair with the largest predicted scalar reward difference from the pool.
2. APL+MD-Score selects the pair with the largest sum of squared ridge-prediction confidence widths across its two responses, prioritizing uncertainty in the estimated multidimensional scores.
3. APL+MD-Scalar fits a Bradley–Terry model (Bradley & Terry, 1952) to historical five-dimensional score differences and preference labels. Treating the predicted scores as fixed, it selects the pair with the largest approximate information gain about the scalarization weights.

These strategies provide comparisons with preference-cone-guided selection based on scalar reward differences, score-estimation uncertainty, and uncertainty in preference weights.

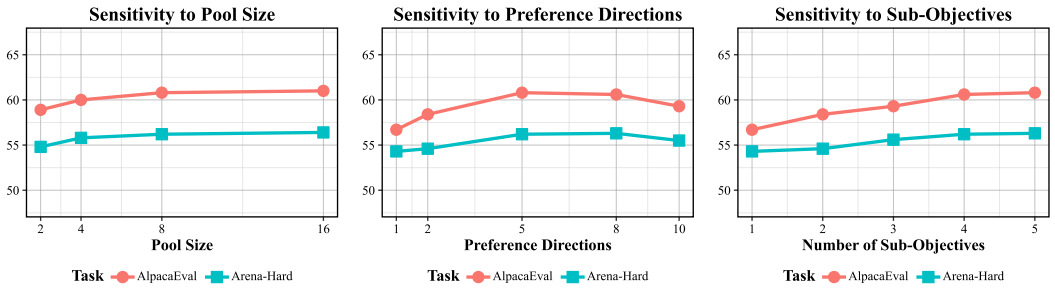
All methods share the same prompt stream, predictor architecture, warm-up data, scoring and preference-query budgets, and DPO hyperparameters. Only queried responses receive oracle scores. We evaluate `Qwen2.5-7B-Instruct` and `Mistral-7B-Instruct-v0.3` over three independent training seeds. The results in Table 3 support the effectiveness of preference-cone-guided selection relative to these alternative uses of multidimensional feedback, answering **Q2**.

4.4 ABLATION STUDIES

We examine the sensitivity of MPG-DPO to the evaluation models, the candidate-pool size, the number of retained preference rules, and the number of sub-objectives.

Table 4: Ablation study on the effectiveness of personalization.

Method	Kimi-K2.6		DeepSeek-V4.1-Flash		Qwen3.8-27B	
	AlpacaEval2	Arena-Hard	AlpacaEval2	Arena-Hard	AlpacaEval2	Arena-Hard
Online DPO	59.02± 0.23	54.34± 0.34	56.45± 0.27	53.73± 0.39	57.73± 0.28	52.98± 0.27
XPO	59.57± 0.38	54.67± 0.41	56.94± 0.34	54.10± 0.38	57.94± 0.26	53.23± 0.34
APL	59.08± 0.28	54.45± 0.25	56.75± 0.23	53.85± 0.33	57.93± 0.32	53.44± 0.31
MPG-DPO	**60.45**± 0.33	**55.98**± 0.36	**57.04**± 0.40	**54.31**± 0.38	**59.26**± 0.29	**54.33**± 0.33



(a) Pool size N. **(b)** Number of preference rules K. **(c)** Number of sub-objectives d.

Figure 2: Sensitivity studies with reported win rates of tuned `Qwen2.5-7B` models versus the base checkpoint.

Effectiveness of personalization. We evaluate adaptation to different oracle-defined preferences using `Qwen2.5-7B` as the backbone. In each setting, one of `Kimi-K2.6`, `DeepSeek-V4.1-Flash`, or `Qwen3.8-27B` serves as the oracle, providing preference feedback during training and computing win rates on AlpacaEval2 and Arena-Hard during evaluation. As shown in Table 4, MPG-DPO achieves higher mean win rates than the evaluated baselines in all three settings, supporting its effectiveness in adapting to different oracle-defined preferences.

Candidate-pool size. We vary $N \in \{2, 4, 8, 16\}$ while holding the preference-query budget and DPO hyperparameters fixed. Larger pools provide more candidates for pair selection but increase response-generation cost. Figure 2a reports win rates on AlpacaEval2 and Arena-Hard. Performance initially improves with pool size, with diminishing gains at larger values of N.

Number of retained preference. We vary the maximum number of rows retained in $\mathbf{W}_t$ to examine how the number of preferences affects pair selection. As shown in Figure 2b, performance improves as the number of retained rules increases from small values, then declines slightly at larger settings. Intermediate settings achieve the highest win rates within the evaluated range.

Number of sub-objectives. We vary the number of sub-objectives by randomly selecting subsets of the FLASK evaluation dimensions. Figure 2c shows that including more dimensions improves downstream performance within the evaluated range. Since the scoring rubrics are generated by the preference oracle, this trend may partly reflect the relevance of these dimensions to the target preferences. The interaction between the number of sub-objectives d and the number of retained rules K warrants further investigation.

These results answer **Q3**. Additional ablation results are provided in Appendix B.

5 Conclusion

In this paper, we proposed MPG-DPO, an online preference alignment method that incorporates historical user preferences into response selection alongside model uncertainty for sample efficiency. Using low-cost multidimensional evaluations, the method constructs a personalized preference cone to filter confidently dominated responses and select pairs exhibiting unresolved trade-offs. This selection mechanism accounts for preference information already implied by past user comparisons. Under the stated assumptions, our analysis suggests that the number of selected cone-dominated responses grows slowly, so such queries make up a vanishing fraction as learning progresses. Experiments across multiple LLM families show improvements over strong baselines under matched preference-query budgets, supporting the effectiveness of preference-cone-guided selection.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems (NeurIPS)*, 24:2312–2320, 2011.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Syrine Belakaria, Joshua Kazdan, Charles Marx, Chris Cundy, Willie Neiswanger, Sanmi Koyejo, Barbara E. Engelhardt, and Stefano Ermon. Sharpe ratio-guided active learning for preference optimization in RLHF. In *Second Conference on Language Modeling*, 2025.

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.

Daniele Calandriello, Zhaohan Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, Rishabh Joshi, Zeyu Zheng, and Bilal Piot. Human alignment of large language models through online preference optimisation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 211–238, 2024.

Shi Cong Cen, Jin Cheng Mei, Katayoon Goshvadi, Han Jun Dai, Tong Yang, Sherry Yang, Dale Schuurmans, Yue Jie Chi, and Bo Dai. Value-incentivized preference optimization: A unified approach to online and offline rlhf. In *The 13th International Conference on Learning Representations (ICLR)*, 2025.

Yair Censor. Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization*, 4 (1):41–59, 1977.

Mingyu Chen, Yiding Chen, Wen Sun, and Xuezhou Zhang. Avoiding exp (r) scaling in rlhf through preference-based exploration. *Advances in Neural Information Processing Systems (NeurIPS)*, 38: 164474–164510, 2025.

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 (NeurIPS)*, 30:4299–4307, 2017.

Nirjhar Das, Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury. Active preference optimization for sample efficient RLHF. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 96–112, 2025.

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.

Komei Fukuda and Alain Prodon. Double description method revisited. In *Franco-Japanese and Franco-Chinese conference on combinatorics and computer science*, pp. 91–111, 1995.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The 10th International Conference on Learning Representations (ICLR)*, 2022.

Saffron Huang, Divya Siddarth, Liane Lovitt, Thomas I. Liao, Esin Durmus, Alex Tamkin, and Deep Ganguli. Collective constitutional AI: Aligning a language model with public input. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1395–1417, 2024.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4334–4353, 2024.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: scaling reinforcement learning from human feedback with AI feedback. Proceedings of the 41st International Conference on Machine Learning (ICML), pp. 26874–26901, 2024.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: arena-hard and benchbuilder pipeline. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 1340–1363, 2025.

Xiaoqiang Lin, Arun Verma, Zhongxiang Dai, Daniela Rus, See-Kiong Ng, and Bryan Kian Hsiang Low. ActiveDPO: Active direct preference optimization for sample-efficient alignment. In *The 14th International Conference on Learning Representations (ICLR)*, 2026.

Lanjihong Ma, Zhen-Yu Zhang, Masashi Sugiyama, and Zhi-Hua Zhou. Selective ensemble based on preference-directed multi-objective bandits. *arXiv preprint arXiv:2606.21929*, 2026.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics (ACL)*, pp. 142–150, 2011.

TS Motzkin, GL Thompson, and RM Thrall. The double description method. *Contributions to the Theory of Games*, (24):51, 1953.

Giyeong Oh, Junghyun Lee, Jaehyun Park, Youngjae Yu, Wonho Bae, and Junhyug Noh. Random is hard to beat: Active selection in online dpo with modern llms. *arXiv preprint arXiv:2604.02766*, 2026.

Minhyeon Oh, Seungjoon Lee, and Jungseul Ok. Comparison-based active preference learning for multi-dimensional personalization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 33145–33166, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:27730–27744, 2022.

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 (NeurIPS)*, 36:53728–53741, 2023.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:71095–71134, 2023.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the workshop on new frontiers in summarization*, pp. 59–63, 2017.

Teng Yang Xie, Dylan J Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Hassan Awadallah, and Alexander Rakhlin. Exploratory preference optimization: Harnessing implicit q*-approximation for sample-efficient rlhf. In *The 13th International Conference on Learning Representations (ICLR)*, 2025.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. Flask: Fine-grained language model evaluation based on alignment skill sets. In *The 12th International Conference on Learning Representations (ICLR)*, volume 2024, pp. 55361–55414, 2024.

AI USE STATEMENT

AI tools were used solely for language editing and polishing. The core concepts, mathematical derivations, experimental designs, and interpretations presented in this paper were produced entirely by the human authors.

APPENDIX

A RELATED WORK

This section reviews relevant prior work in detail.

Exploration in online preference alignment. Online preference alignment repeatedly generates responses from the current policy, obtains preference feedback, and updates the policy on the newly collected data (Calandriello et al., 2024). A fruitful line of research aims to enhance policy-level coverage and generate high-value responses using a scalar implicit reward. XPO adds a principled exploration bonus to online DPO so that the policy can move beyond the support of the initial model and discover responses with higher latent reward (Xie et al., 2025). VPO similarly introduces optimism through value regularization, while recent theoretical work studies when preference-based exploration avoids poor dependence on the reward range (Cen et al., 2025; Chen et al., 2025). These methods primarily modify the DPO loss and can be seamlessly integrated into our algorithm.

Active preference selection. A second line of work assumes a collection of candidate comparisons and actively decides which examples should receive costly preference labels (Das et al., 2025; Lin et al., 2026; Oh et al., 2026). APO formulates RLHF as a contextual preference bandit and allocates queries to statistically important prompt–response comparisons (Das et al., 2025). ActiveDPO derives a selection score from uncertainty in the LLM's implicit reward difference and uses gradient projection and batch selection to make this criterion practical for nonlinear language models (Lin et al., 2026). Other work designs risk-adjusted active-learning criteria for preference optimization (Belakaria et al., 2025). These approaches rank candidates using scalar reward uncertainty while not exploiting the sub-objective trade-offs considered in our setting.

Constitutional AI and rule-guided alignment. Constitutional AI replaces repeated human supervision with a written set of high-level principles that guides self-critique, response revision, and AI-generated preference labels (Bai et al., 2022). Lee et al. (2024) further study empirically whether an external language model can provide preference supervision at a quality comparable to human feedback, while collective constitutional AI obtains the constitution through a public deliberation process rather than solely from model developers (Huang et al., 2024). These methods demonstrate that explicit rules can make alignment supervision more scalable and interpretable. However, their constitutions are specified as natural-language principles and are typically shared across interactions.

Pareto learning. The preference Pareto cone originates from Censor (1977) and has recently been used by Rame et al. (2023) for reward interpolation and by Ma et al. (2026) in multi-objective bandits. We adopt the same geometric abstraction but differ in both construction and use: our cone is estimated online from noisy pairwise comparisons via a consistency check, and is adaptively updated as new comparisons arrive to actively guide response selection.

B ADDITIONAL EXPERIMENTAL DETAILS

B.1 SETUP DETAILS

Experimental Setup. We run all online DPO variants for a fixed number of steps with a matched global batch size across all settings. Specific hyperparameters are listed in Table 5.

For MPG-DPO, we use additional hyper-parameters of pool size $N = 8$, $d = 5$ sub-objectives and $K = 8$. We further use $\tau = 0.1$ and $\lambda = 1.0$ by default. We use $\delta = 0.1$, $S = 1.0$, with $\alpha_t(\delta)$ computed as detailed in Section 3.3.

Table 5: Online DPO hyperparameters

Hyperparameter	Value	Hyperparameter	Value
LoRA rank (r)	32	Epochs (query budget)	3
LoRA (α)	64	Global Batch Size	64
Learning Rate	5×10^{-5}	Warm-up Ratio	0.05
Optimizer	AdamW	Max Sequence Length	256
Precision	`bf16`	Gradient Clipping	1.0

Table 6: Results using `Kimi-K2.6` sub-objective oracle. Each entry reports mean $\pm$ standard deviation over three seeds.

LLM	Method	IMDB Average Reward	TL;DR Win Rate (%)	ALPACAEVAL2 LC Win Rate (%)	ARENA-HARD Win Rate (%)
`Qwen2.5-7B`	Online DPO	81.6$\pm$ 0.55	63.38$\pm$ 0.24	59.03$\pm$ 0.18	54.68$\pm$ 0.17
	XPO	81.8$\pm$ 0.66	63.82$\pm$ 0.27	59.14$\pm$ 0.20	54.74$\pm$ 0.24
	APL	82.1$\pm$ 0.61	64.02$\pm$ 0.26	59.13$\pm$ 0.22	54.94$\pm$ 0.21
	APL+MD	82.8$\pm$ 0.66	64.37$\pm$ 0.29	59.51$\pm$ 0.27	55.21$\pm$ 0.26
	MPG-DPO	**85.1**$\pm$ 0.63	**66.26**$\pm$ 0.26	**61.73**$\pm$ 0.25	**58.13**$\pm$ 0.25

The human preference oracle is queried only once per update to determine the winning/losing response in the chosen pair. For W_t initialization, a fixed amount of preference responses can be provided for initial computation, which can be conducted as an offline pre-processing step.

Computational Environment. All experiments were conducted on a single machine with 4 NVIDIA H100 80GB GPUs. Experiments were run in an environment using CUDA version 12.8 and PyTorch 2.7.1.

B.2 ADDITIONAL RESULTS

Query efficiency. We next verify the importance of the warm-up step to construct W_t. Here, we use `Qwen2.5-7B-Instruct` and `Mistral-7B-Instruct-v0.3` and vary the amount of initial data necessary for W_t construction. We re-use the same alignment procedure as described before. In Figure 3, we can see that a moderate amount of pre-labeled samples is important for properly initializing W_t. While the complete lack of any initial queries causes training to progress more slowly, which is sensible due to the need to construct W_t completely from online pair-wise comparisons, using a moderate amount of warm-up labels can ensure high performance.

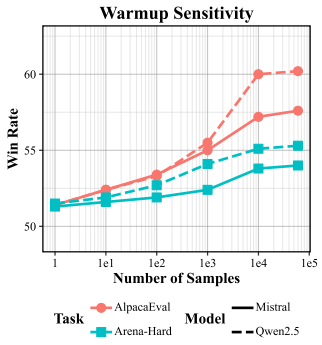


Figure 3: Number of queries for W_t warm-up.

Cone construction. As a final verification, we also audit the cone construction process in Eq. 2. We take the tuned model at different stages of tuning (25%, 50%, 75%, and 100%) and measure the filtering rate (removal of responses in the pool). We observe in Figure 4 that as the tuning progresses and both the cone as well as W_t can encode increasing information about the model preferences, the model is able to filter out an increasing number of potentially less-informative responses within the pool.

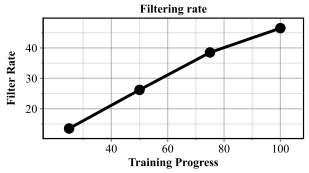


Figure 4: Filtering rate of responses during alignment.

Varying Sub-Objective Oracle. We also run a comparison by changing the use of the sub-objective oracle. As a simple replacement, we prompt a `Kimi-K2.6` using the same FLASK objectives and templates used for Prometheus, obtaining scores from the new model for training and tuning with the setup described for the main experimental results.

C Proof of Theorem 1

We first review the notations. At round t, the learner uses C_{t-1} and $\mathbf{W}_{t-1}$ constructed from previous feedback, together with V_t and $\widehat{\theta}_t$ fitted to noisy scores from rounds $s < t$. The current queried scores and preference label are observed after selection. The operational confidence bounds use

$$\widehat{m}_{t,k}(y \mid y') = \frac{\mathbf{w}_{t-1,k}^\top \widehat{\theta}_t^\top [\phi(y) - \phi(y')]}{U_{t-1,k} - L_{t-1,k}}, \qquad c_{t,k}(y, y') = \frac{\alpha_t(\delta) \|\phi(y) - \phi(y')\|_{V_t^{-1}}}{U_{t-1,k} - L_{t-1,k}},$$

with $k \in [K_{t-1}]$ and $m_{t,k}$ obtained by replacing $\widehat{\theta}_t$ with θ^*. All optimal sets are evaluated on the latent scores $\mathbf{x}_{t,i} = \mu(\mathbf{x}_t, y_t^i)$ in the current candidate pool. Regret, $\Delta_{t,i}$, and $\mathcal{D}_t(i)$ in Assumption 1 all use the same cone C_{t-1} as selection. In particular, $P^*_{C_{t-1}}$ is evaluated on the current pool, before observing the current preference label. We use $q_t = \sum_{i \in B_t} \|\phi(y_t^i)\|^2_{V_t^{-1}}$ as defined in Section 3.4.

C.1 Technical Lemmas

In this part, we introduce necessary technical lemmas.

Lemma 1 (Simultaneous directional confidence bound) *Under the score model in Section 3.1, for any $\delta \in (0,1)$, with probability at least $1-\delta$, simultaneously for every round t, every $\mathbf{v} \in \mathbb{R}^p$, and every nonzero $\mathbf{w} \in \mathbb{R}^d$,*

$$\frac{|\mathbf{w}^\top (\widehat{\theta}_t - \theta^*)^\top \mathbf{v}|}{\|\mathbf{w}\|_1} \le \alpha_t(\delta) \frac{\|\mathbf{w}\|_2}{\|\mathbf{w}\|_1} \|\mathbf{v}\|_{V_t^{-1}}. \tag{9}$$

Consequently, $\mathrm{LCB}_{t,k}(y \mid y') \le m_{t,k}(y \mid y') \le \mathrm{UCB}_{t,k}(y \mid y')$ *simultaneously for all rounds, response pairs, and rows of* $\mathbf{W}_{t-1}$.

Proof of Lemma 1. Order the queried score vectors chronologically, using a fixed order within each pair. The corresponding feature vectors are known before each score is observed. By the self-normalized concentration inequality of Abbasi-Yadkori et al. (2011, Theorem 1) and a union bound over the d coordinates, with probability at least $1-\delta$, simultaneously for all t and $q \in [d]$,

$$\left\| \sum_{s<t} \sum_{i \in B_s} \phi(y_s^i) \varepsilon_s(y_s^i)[q] \right\|_{V_t^{-1}} \le \sigma_z \sqrt{2 \log \frac{d}{\delta} + \log \frac{\det V_t}{\det(\lambda I_p)}}.$$

Substitution of the noisy observations into the ridge estimator gives

$$\widehat{\theta}_t - \theta^* = V_t^{-1} \left(\sum_{s<t} \sum_{i \in B_s} \phi(y_s^i) \varepsilon_s(y_s^i)^\top - \lambda \theta^* \right).$$

Combining the coordinate bounds in Frobenius norm and using $V_t \succeq \lambda I_p$ yields

$$\|V_t^{1/2}(\widehat{\theta}_t - \theta^*)\|_F \le \sigma_z \sqrt{d \left(2 \log \frac{d}{\delta} + \log \frac{\det V_t}{\det(\lambda I_p)} \right)} + \lambda \|V_t^{-1/2} \theta^*\|_F \le \alpha_t(\delta).$$

Thus, for every $\mathbf{v}$ and $\mathbf{w}$, Cauchy–Schwarz implies

$$|\mathbf{w}^\top (\widehat{\theta}_t - \theta^*)^\top \mathbf{v}| \le \|\mathbf{w}\|_2 \|V_t^{1/2}(\widehat{\theta}_t - \theta^*)\|_F \|\mathbf{v}\|_{V_t^{-1}} \le \alpha_t(\delta) \|\mathbf{w}\|_2 \|\mathbf{v}\|_{V_t^{-1}}.$$

Dividing by $\|\mathbf{w}\|_1 > 0$ proves (9). Each unit-norm row satisfies

$$U_{t-1,k} - L_{t-1,k} = \sum_{q=1}^{d} |\mathbf{w}_{t-1,k}[q]| = \|\mathbf{w}_{t-1,k}\|_1 \ge \|\mathbf{w}_{t-1,k}\|_2 = 1.$$

Taking $\mathbf{w} = \mathbf{w}_{t-1,k}$ and $\mathbf{v} = \phi(y) - \phi(y')$ gives $|m_{t,k}(y \mid y') - \widehat{m}_{t,k}(y \mid y')| \le c_{t,k}(y, y')$. The confidence event is uniform over time and all directions, so adaptive candidate pools and preference matrices require no additional union bound.

Remark 2 *Noisy scores and preference labels determine the realized cones through (2). Lemma 1 controls score-estimation error uniformly over time and directions, and the regret analysis uses the gap and coverage assumptions under the cones available before querying, without a separate concentration bound for preference labels. The resulting guarantee is relative to the learned cones; recovery of a latent preference cone or convergence of the true reward requires additional conditions.*

Lemma 2 (Filtering with the previous preference matrix) *On the event of Lemma 1, the filter evaluated using $\mathbf{W}_{t-1}$ retains every Pareto C_{t-1}-optimal response in the current pool. If $i \in \mathcal{N}_t$ and some response j satisfies*

$$\min_{k\in[K_{t-1}]} \frac{\mathbf{w}_{t-1,k}^\top[\mu(y_t^j)-\mu(y_t^i)]}{U_{t-1,k}-L_{t-1,k}} \geq \Delta,$$

then

$$\Delta \leq 2\alpha_t(\delta)\|\phi(y_t^j)-\phi(y_t^i)\|_{V_t^{-1}}. \tag{10}$$

If this j also satisfies (7), then $\Delta^2 \leq 4\alpha_t^2(\delta)\kappa q_t$.

Proof of Lemma 2. If the filter removes i, another response j has $\mathrm{LCB}_{t,k}(y_t^j \mid y_t^i) > 0$ for every $k \in [K_{t-1}]$. Lemma 1 implies $\mathbf{W}_{t-1}[\mu(y_t^j)-\mu(y_t^i)] > \mathbf{0}$, and hence $j \succ_{C_{t-1}} i$. Thus no Pareto C_{t-1}-optimal response in the current pool is removed.

For a retained i and a j satisfying the stated margin condition, some $k \in [K_{t-1}]$ has $\mathrm{LCB}_{t,k}(y_t^j \mid y_t^i) \leq 0$. The operational true margin $m_{t,k}$ uses $\mathbf{w}_{t-1,k}$, so Lemma 1 gives

$$0 \geq \mathrm{LCB}_{t,k}(y_t^j \mid y_t^i) \geq m_{t,k}(y_t^j \mid y_t^i) - 2c_{t,k}(y_t^j, y_t^i) \geq \Delta - 2c_{t,k}(y_t^j, y_t^i).$$

Since $U_{t-1,k} - L_{t-1,k} = \|\mathbf{w}_{t-1,k}\|_1 \geq 1$,

$$\Delta \leq \frac{2\alpha_t(\delta)\|\phi(y_t^j)-\phi(y_t^i)\|_{V_t^{-1}}}{U_{t-1,k}-L_{t-1,k}} \leq 2\alpha_t(\delta)\|\phi(y_t^j)-\phi(y_t^i)\|_{V_t^{-1}}.$$

Squaring and applying (7) proves the final claim. No current-round score is required to evaluate the filter; true margins are used only in the analysis.

Remark 3 (A sufficient condition for coverage) *Assumption 2 holds with $\kappa = 4\gamma$ if, for a constant $\gamma > 0$ independent of t,*

$$\max_{j\in[N]} \|\phi(y_t^j)\|^2_{V_t^{-1}} \leq \gamma q_t.$$

For any selected response i and any gap-achieving dominator j, the triangle inequality gives

$$\|\phi(y_t^j)-\phi(y_t^i)\|_{V_t^{-1}} \leq \|\phi(y_t^j)\|_{V_t^{-1}} + \|\phi(y_t^i)\|_{V_t^{-1}} \leq 2\sqrt{\gamma q_t}.$$

The maximum ranges over the full candidate pool to cover every possible gap-achieving dominator under C_{t-1}. Coverage remains a separate assumption and is not implied by the trade-off selection criterion.

Lemma 3 (Cumulative feature uncertainty) *For $\lambda \geq 1$, $|B_t| \leq 2$, and $\|\phi(y_t^i)\|_2 \leq 1$,*

$$\sum_{t=1}^{T} q_t \leq 3\log\frac{\det V_{T+1}}{\det V_1} \leq 3p\log\left(1+\frac{2T}{\lambda p}\right). \tag{11}$$

Proof of Lemma 3. Let $H_t = V_t^{-1/2}(\sum_{i\in B_t}\phi(y_t^i)\phi(y_t^i)^\top)V_t^{-1/2} \succeq 0$. Its eigenvalues $\mu_{t,1},\ldots,\mu_{t,p}$ are nonnegative, with $\mathrm{tr}(H_t) = \sum_{r=1}^p \mu_{t,r} = q_t$. Moreover, $V_t \succeq \lambda I_p$ gives

$$0 \leq q_t \leq \lambda^{-1}\sum_{i\in B_t}\|\phi(y_t^i)\|_2^2 \leq 2.$$

Since $V_{t+1} = V_t^{1/2}(I_p + H_t)V_t^{1/2}$, we have

$$\log\frac{\det V_{t+1}}{\det V_t} = \log\prod_{r=1}^{p}(1+\mu_{t,r}) \geq \log(1+q_t) = \int_0^{q_t}\frac{du}{1+u} \geq \frac{q_t}{3}.$$

The product inequality follows because all higher-order terms in its expansion are nonnegative; the last inequality uses $q_t \le 2$. Summing over t telescopes the log determinants and proves the first bound. For the second, $V_1 = \lambda I_p$ and

$$\operatorname{tr}(V_{T+1}) = \lambda p + \sum_{t=1}^{T} \sum_{i \in B_t} \|\phi(y_t^i)\|_2^2 \le \lambda p + 2T.$$

Applying the arithmetic–geometric mean inequality to the positive eigenvalues of V_{T+1} yields

$$\det V_{T+1} \le \left(\frac{\operatorname{tr}(V_{T+1})}{p} \right)^p \le \left(\lambda + \frac{2T}{p} \right)^p.$$

Dividing by $\det V_1 = \lambda^p$ and taking logarithms completes the proof.

C.2 Proof of Theorem 1

In this part, we provide a detailed proof of Theorem 1. We work on the event of Lemma 1 and fix arbitrary choices $a_t \in B_t$. At every query with $a_t \notin P^*_{C_{t-1}}$, choose a gap-achieving $j \in \mathcal{D}_t(a_t)$ satisfying Assumption 2. Assumption 1 gives

$$\min_{k \in [K_{t-1}]} \frac{\mathbf{w}_{t-1,k}^\top [\mu(y_t^j) - \mu(y_t^{a_t})]}{U_{t-1,k} - L_{t-1,k}} = \Delta_{t,a_t} \ge \Delta.$$

Since a_t survived the filter based on $\mathbf{W}_{t-1}$, Lemma 2 and coverage yield $\Delta^2 \le 4\alpha_t^2(\delta)\kappa q_t$. For any generated query prefix of length T, the determinant bound in Lemma 3 gives $\alpha_t(\delta) \le \bar{\alpha}_T(\delta)$ for $t \le T$. Summing over nonoptimal selections and applying the same lemma yields

$$\operatorname{Regret}^{01}(T) \le \frac{4\bar{\alpha}_T^2(\delta)\kappa}{\Delta^2} \sum_{t=1}^{T} q_t \le \frac{12\bar{\alpha}_T^2(\delta)\kappa p}{\Delta^2} \log\left(1 + \frac{2T}{\lambda p}\right).$$

This proves (8) on the simultaneous confidence event, whose probability is at least $1 - \delta$.

The choices a_t were arbitrary on the same confidence event. In particular, choose a nonoptimal member whenever B_t contains one. This gives the same bound for $\sum_{t=1}^{T} \mathbb{I}\{B_t \not\subseteq P^*_{C_{t-1}}\}$. Since each pair contains at most two responses, $\sum_{t=1}^{T} \sum_{i \in B_t} \mathbb{I}\{i \notin P^*_{C_{t-1}}\}$ is at most twice this count.

For fixed $\lambda, S, \sigma_z, \kappa, p, d, \Delta, \delta$, we have $\bar{\alpha}_T^2(\delta) = O(\log T)$. If queries continue indefinitely, the bound holds simultaneously for all T on the same event, so $\operatorname{Regret}^{01}(T)/T \to 0$ with probability at least $1 - \delta$.

C.3 Full Statement and Proof of Proposition 1

In this part, we provide the full statement of Proposition 1 and corresponding proof. Under the conditions of Theorem 1, let $\mathcal{R}_T(\delta)$ denote the right-hand side of (8), and let $G_T = \sum_{t=1}^{T} \mathbb{I}\{B_t \subseteq P^*_{C_{t-1}}\}$ count queries whose two responses are Pareto-optimal in their respective pools under the cones available before querying. For each integer $M \ge 1$, let $Q_M = \inf\{T \in \mathbb{N} : G_T \ge M\}$ be the number of expensive queries needed to collect M such comparisons, with the infimum taken over generated query prefixes and $\inf \varnothing = +\infty$. With probability at least $1 - \delta$, simultaneously for all generated query prefixes $T \ge 1$, and for all integers $M \ge 1$ such that at least $2M$ queries are generated,

$$\frac{G_T}{T} \ge 1 - \frac{\mathcal{R}_T(\delta)}{T}, \qquad M \le Q_M \le M + \lceil \mathcal{R}_{2M}(\delta) \rceil \quad \text{if } \mathcal{R}_{2M}(\delta) \le M. \tag{12}$$

If queries continue indefinitely, then for fixed problem parameters and δ, $Q_M = M + O(\log^2 M)$ as $M \to \infty$.

Proof of Proposition 1. We work on the same simultaneous confidence event as in Theorem 1. For each round, choose $a_t \in B_t \setminus P^*_{C_{t-1}}$ when this set is nonempty, and choose any $a_t \in B_t$ otherwise.

Since the theorem holds simultaneously for all choices of a_t, including these choices made for evaluation, we have

$$T - G_T = \sum_{t=1}^{T} \mathbb{I}\{B_t \not\subseteq P^*_{C_{t-1}}\} = \sum_{t=1}^{T} \mathbb{I}\{a_t \notin P^*_{C_{t-1}}\} \leq \mathcal{R}_T(\delta).$$

Dividing by T proves the first inequality in (12). Each round produces at most one counted comparison, so $Q_M \geq M$.

The bound $\mathcal{R}_T(\delta)$ is nondecreasing in T. If $\mathcal{R}_{2M}(\delta) \leq M$, then the integer $T = M + \lceil \mathcal{R}_{2M}(\delta) \rceil$ satisfies $T \leq 2M$, and hence

$$G_T \geq T - \mathcal{R}_T(\delta) \geq M + \lceil \mathcal{R}_{2M}(\delta) \rceil - \mathcal{R}_{2M}(\delta) \geq M.$$

It follows that $Q_M \leq M + \lceil \mathcal{R}_{2M}(\delta) \rceil$. For fixed problem parameters and δ, $\mathcal{R}_{2M}(\delta) = O(\log^2 M) = o(M)$, so, if queries continue indefinitely, the condition holds for all sufficiently large M, giving $Q_M = M + O(\log^2 M)$ on the same confidence event. No additional union bound is needed.

The count G_T evaluates each comparison using its own round's candidate pool and the cone available before querying; previously collected comparisons are not reevaluated under later cones. The proposition quantifies query allocation to nondominated pairs, without assuming that all such comparisons have equal information value or establishing a label complexity for a target policy reward.