arXiv:2412.04628v4 [cs.LG] 19 Jun 2025

Multi-Preference Optimization: Generalizing DPO via Set-Level Contrasts

Taneesh Gupta[*]
Microsoft

Rahul Madhavan[*]
IISc, Bangalore

Xuchao Zhang
Microsoft

Nagarajan Natarajan
Microsoft

Chetan Bansal
Microsoft

Saravan Rajmohan
Microsoft

Abstract

Direct Preference Optimization (DPO) has become a popular approach for aligning language models using pairwise preferences. However, in practical post-training pipelines, on-policy generation typically yields multiple candidate responses per prompt, which are scored by a reward model to guide learning. In this setting, we propose **Multi-Preference Optimization (MPO)**, a generalization of DPO that optimizes over entire sets of responses by extending the Bradley–Terry model to groupwise comparisons between chosen and rejected sets. To further enhance learning, MPO employs deviation-based weighting, which emphasizes outlier responses that deviate most from the mean reward, effectively inducing a self-paced curriculum. We theoretically prove that MPO reduces alignment bias at a rate of $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ with respect to the number of responses per query. Empirically, MPO achieves state-of-the-art performance on the UltraFeedback benchmark and yields up to $\sim$ 17.5% improvement over the state-of-the-art baseline in length-controlled win rate on AlpacaEval2, establishing a new baseline for preference-based alignment.

1 Introduction

Aligning large language models (LLMs) to follow instructions effectively and adhere to desired behaviors is a cornerstone of modern AI development (Ouyang et al., 2022; Christiano et al., 2017). Direct Preference Optimization (DPO) has emerged as a prominent and computationally efficient paradigm for the alignment task. DPO bypasses the need for explicit reward model training by directly optimizing a policy based on pairwise preferences (Rafailov et al., 2024). However, the standard DPO framework is inherently limited to comparing single pairs of (preferred, dispreferred) responses; furthermore, the standard DPO implementations use the most-preferred and the least-preferred responses per query (Wu et al., 2024; Meng et al., 2024) discarding the rest.

This pairwise formulation becomes restrictive as modern alignment pipelines, especially those involving on-policy generation or self-play, often produce a multitude of candidate responses per prompt (Wu et al., 2024; Chen et al., 2024c; Pang et al., 2024; Tang et al., 2025), wasting valuable supervisory signal and computational effort. To overcome this, we propose **Multi-Preference Optimization (MPO)**, which generalizes DPO by using a group-contrastive loss

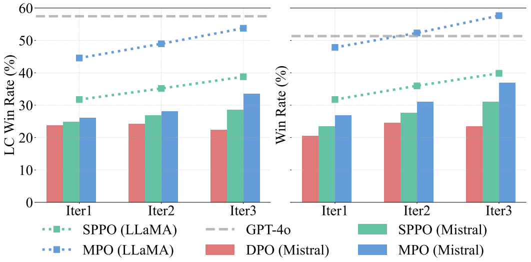

Figure 1: Multi-Preference Optimization (MPO) achieves state-of-the-art results on WR, LC-WR on AlpacaEval2, with performance nearly matching GPT-4o, demonstrating MPO's efficacy for post-training optimization.

---

[*] Equal contribution

Preprint. Under review.

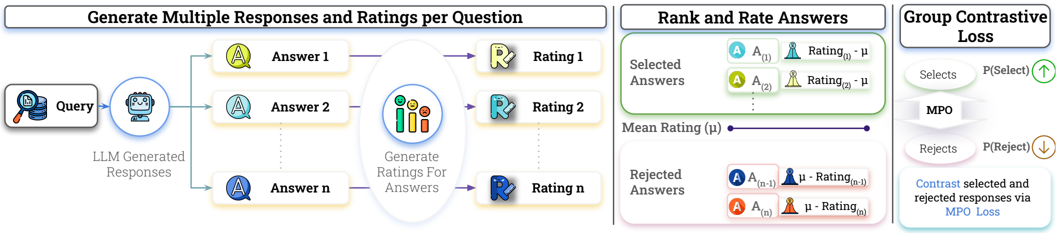
Generate Multiple Responses and Ratings per Question
Query
LLM Generated Responses
Answer 1
Answer 2
Answer n
Generate Ratings For Answers
Rating 1
Rating 2
Rating n
Rank and Rate Answers
Selected Answers
$A_{(1)}$
$\text{Rating}_{(1)} - \mu$
$A_{(2)}$
$\text{Rating}_{(2)} - \mu$
Mean Rating (μ)
Rejected Answers
$A_{(n-1)}$
$\mu - \text{Rating}_{(n-1)}$
$A_{(n)}$
$\mu - \text{Rating}_{(n)}$
Group Contrastive Loss
Selects
P(Select)
MPO
Rejects
P(Reject)
Contrast selected and rejected responses via MPO Loss

Figure 2: MPO pipeline: Responses are split into accepted and rejected sets using human labels or reward model scores. We assign weights based on deviation from the mean and optimize to upweight the accepted group while downweighting the rejected group using our MPO loss.

to compare entire sets of "accepted" versus "rejected" responses. This core idea extends DPO's underlying Bradley-Terry model (Bradley and Terry, 1952) from pairwise to groupwise comparisons.

Leveraging multiple responses for alignment is by no means new. Several alternative strategies have been proposed for post-training optimization using richer preference data. Traditional policy-gradient RLHF methods, such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), can implicitly utilize multiple responses through a learned reward function. More recent methods explicitly target groupwise comparisons, like Group Relative Policy Optimization (GRPO) (Liu et al., 2024a), or aim to integrate explicit reward signals more directly, including Reward-Aware Preference Optimization (RPO) (Sun et al., 2025) and cross-entropy-based methods like InfoNCA (Chen et al., 2024a). In fact, we could apply the standard DPO loss across all possible $\binom{k}{2}$ pairs (Tunstall et al., 2023), without discarding any of the k preferences per query; or use ranking-based methods to leverage full or partial rankings over all k responses (Dong et al., 2023; Yuan et al., 2023).

Many of these approaches, however, face significant hurdles. Methods like PPO, and RPO typically necessitate a well-calibrated reward model to provide the scalar values or advantages crucial for their optimization objectives (Schulman et al., 2017; Liu et al., 2024a; Sun et al., 2025). In fact methods like GRPO only work with verifiable rewards (Liu et al., 2024a). Similarly, InfoNCA relies on the magnitude of explicit reward scores for its target distribution (Chen et al., 2024a). The challenge of obtaining consistently accurate reward scores, particularly given the known difficulties in training stable value functions via regression (Farebrother et al., 2024), can make these approaches sensitive to reward noise or mis-calibration. On the other hand, naively applying DPO to all pairs incurs significant computational overhead and potential optimization instabilities, while Plackett-Luce models demand reliable full rankings, which are often expensive and difficult to obtain (Chen et al., 2024b), and may cause alignment brittleness.

MPO is designed to overcome these specific challenges by retaining DPO's robust contrastive nature while effectively scaling to multiple preferences. It does not need well-calibrated reward models or full ranking over the responses, and instead can work with relative comparisons (ordinal preferences) over exact value regression (Farebrother et al., 2024). When fine-grained scores are available, these sets are partitioned into "accepted" and "rejected" groups based on a query-specific score mean. MPO then employs a novel groupwise contrastive loss, generalizing the Bradley-Terry model (Bradley and Terry, 1952), which is distinctively enhanced by a deviation-based weighting scheme. This weighting prioritizes informative outliers, creating an implicit curriculum (Bengio et al., 2009) and using reward scores primarily to define groups and modulate influence. MPO provides a theoretically grounded method with state-of-the-art results (Section 6).

We provide a detailed illustration of our method in Figure 2. Our primary contributions are:

1. **(Set-Based) Multi-Preference Optimization (MPO):** We introduce **Multi-Preference Optimization (MPO)**, which generalizes DPO's pairwise contrast to efficiently learn from *sets* of preferred and non-preferred responses per query. MPO employs a novel groupwise objective based on the Bradley-Terry model, enabling richer supervision than single positive/negative pairs.

2. **Deviation-Weighted Learning (W-MPO) for Fine-Grained Signals:** We propose **Weighted MPO (W-MPO)**, an extension that incorporates fine-grained reward information by additively adjusting response logits based on their deviation from the query-specific mean reward. This prioritizes informative outliers, creating an implicit data-driven curriculum (Section 4.4).

3. **State-of-the-Art Alignment Performance:** Both MPO and W-MPO achieve state-of-the-art results on diverse alignment benchmarks, including **AlpacaEval 2.0**, **Arena-Hard**, and **MT-Bench**, outperforming established baselines with various LLMs (e.g., Mistral-7B and Llama-3-8B).

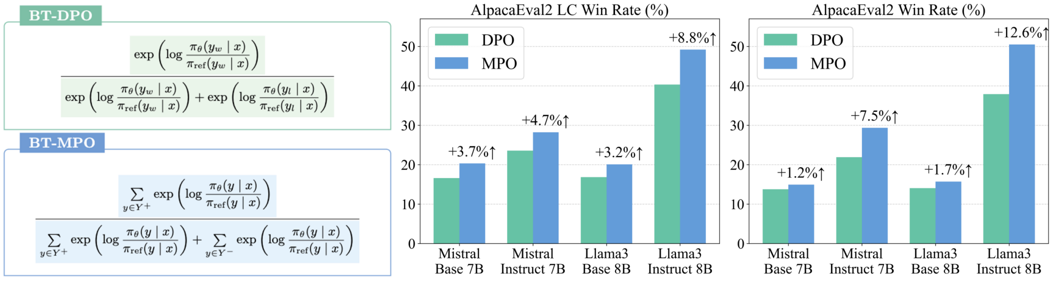

Figure 3: **MPO generalizes DPO** by extending the Bradley-Terry formulation to groupwise comparisons over multiple responses. This richer supervision yields stronger performance across model families. MPO gracefully reduces to DPO when each set contains a single response.

2 Related Work

We briefly situate MPO within the evolving landscape of preference optimization, deferring a more comprehensive survey to Appendix A. The alignment of Large Language Models (LLMs) has significantly advanced from early Reinforcement Learning from Human Feedback (RLHF) frameworks, which typically employ reward models and policy gradient methods like PPO (Schulman et al., 2017; Ouyang et al., 2022). A pivotal shift occurred with Direct Preference Optimization (DPO) (Rafailov et al., 2024), which directly learns from pairwise preferences using a contrastive loss based on the Bradley-Terry model (Bradley and Terry, 1952), simplifying the alignment pipeline. This contrastive learning paradigm, where models learn by distinguishing between positive and negative examples, has foundations in self-supervised learning, notably with objectives like InfoNCE (Oord et al., 2018) which maximize mutual information between related samples.

Many recent works extend DPO's pairwise approach. Some focus on diversified objectives or reward modeling, such as KTO (Ethayarajh et al., 2024) and TDPO (Zeng et al., 2024) for response or token-level alignment. Others modify the DPO framework by altering the reference model, adding regularizers for issues like length bias, or unifying preference learning with supervised objectives, exemplified by SPIN (Chen et al., 2024c), CPO (Xu et al., 2024), ORPO (Hong et al., 2024), SimPO (Meng et al., 2024), R-DPO (Park et al., 2024), and LD-DPO (Liu et al., 2024c). Other works explore different structures for preference signals beyond simple pairs, such as list-wise or rank-based supervision as seen in RAFT (Dong et al., 2023) and RRHF (Yuan et al., 2023).

Leveraging the multiple responses common in on-policy generation or rich datasets (Cui et al., 2023; Gupta et al., 2025) necessitates moving beyond pairwise DPO. While InfoNCA (Chen et al., 2024a) adapts classification objectives for multi-response scalar rewards, MPO provides a distinct paradigm by generalizing DPO's contrastive logic to operate on *sets*. This set-based comparison, especially with W-MPO's deviation-based weighting, robustly utilizes group-level preference signals for post-training optimization.

3 Preliminaries and Technical Background

We frame preference optimization as learning a policy π_θ based on feedback on the candidate responses $\{y_1, \ldots, y_n\}$ to a query x. This feedback often implies a "strength" or "utility" $u(y)$ for each response. In many DPO-style methods, this utility is commonly derived from the policy's log-probabilities relative to a reference model π_{ref}, specifically $u(y) = \exp(\beta r_\theta(y \mid x))$, where $r_\theta(y \mid x) = \log(\pi_\theta(y \mid x)/\pi_{\text{ref}}(y \mid x))$ and β is a positive scaling factor. Different alignment methods incorporate these utilities based on how responses are grouped (or "binned") and how preferences between these responses are modeled. Broadly, these approaches rest on a shared conceptual foundation:

A Unified View of Preference Optimization

Preference optimization can be conceptualized as a probabilistic selection process. The likelihood of favoring certain responses over others is governed by underlying "strength" or "utility" measures associated with each response. A key feature of this optimization process is that these utility measures are linked to the parameters of the language model π_θ being aligned.

3.1 Pairwise and Setwise Preference Models (DPO and MPO)

The foundation for modeling preferences between groups of items can be traced to choice theory (Luce et al., 1959). Consider a scenario with two distinct sets (or "bins") of responses: a preferred

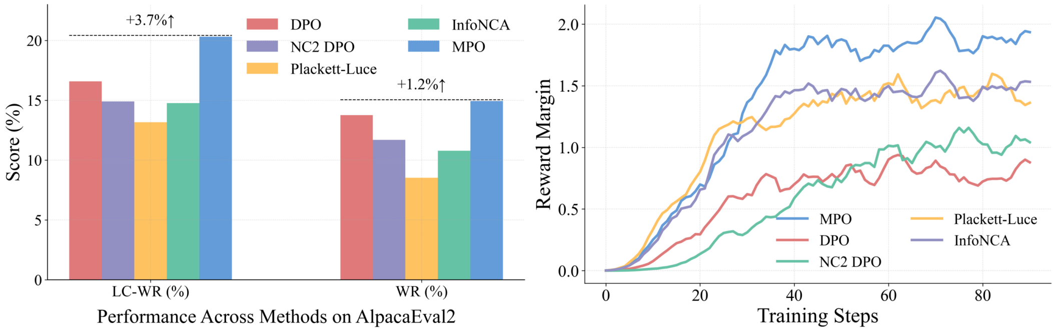

Figure 4: MPO's consistent reward margin advantage during training on Ultrafeedback enables clearer distinction between preferred and rejected responses, which translates into downstream evaluation as well on Mistral-7B.

("chosen") set $\mathcal{Y}^+$ and a dispreferred ("rejected") set $\mathcal{Y}^-$. If we were to select a single representative response from the combined pool $\mathcal{Y}_{\text{all}} = \mathcal{Y}^+ \cup \mathcal{Y}^-$, where the probability of selecting any specific response y is proportional to its intrinsic strength or utility $u(y)$, then the probability that this selected representative comes from the preferred set $\mathcal{Y}^+$ naturally models the collective preference. We use $\mathcal{Y}^+ > \mathcal{Y}^-$ as shorthand for "$\mathcal{Y}^+$ is preferred to $\mathcal{Y}^-$." The probability of this event is expressed as:

$$P(\mathcal{Y}^+ > \mathcal{Y}^-) = \frac{\sum_{y \in \mathcal{Y}^+} u(y)}{\sum_{y' \in \mathcal{Y}_{\text{all}}} u(y')} = \frac{\sum_{y \in \mathcal{Y}^+} u(y)}{\sum_{y \in \mathcal{Y}^+} u(y) + \sum_{y \in \mathcal{Y}^-} u(y)}. \tag{1}$$

In the context of preference optimization and following the Bradley-Terry model (Bradley and Terry, 1952), these utilities $u(y)$ are often defined as an exponential function of an underlying score or logit s_y, i.e., $u(y) = \exp(s_y)$. This exponential transformation ensures positivity of utilities and naturally leads to the logistic (sigmoid) function when comparing two individual items. Eq. 1 forms the conceptual basis for both DPOand our proposed MPO (See comparison in Figure 3).

Direct Preference Optimization (DPO). DPO(Rafailov et al., 2024) is a special case of Eq. 1 where each set contains a single item: $\mathcal{Y}^+ = \{y_w\}$ (winner) and $\mathcal{Y}^- = \{y_l\}$ (loser). The DPOobjective is to maximize $\log P(y_w > y_l)$. Using $u(y) = \exp(\beta r_\theta(y|x))$ implies $\log u(y) = \beta r_\theta(y|x)$, so the standard DPOloss becomes:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)}\left[\log \sigma\big(\beta(r_\theta(y_w \mid x) - r_\theta(y_l \mid x))\big)\right]. \tag{2}$$

Multi-Preference Optimization (MPO): A generalization of DPO. Our work, MPO, directly utilizes the setwise preference model from Eq. 1 where $\mathcal{Y}^+$ and $\mathcal{Y}^-$ can contain multiple responses. The MPO objective is to maximize $\log P(\mathcal{Y}^+ > \mathcal{Y}^-)$, using the same utility definition $u(y) = \exp(\beta r_\theta(y|x))$. The specific construction of $\mathcal{Y}^+, \mathcal{Y}^-$ from data, and the introduction of weighted utilities for the W-MPO variant, are detailed in Section 4.

3.2 Rank-Based and Distributional Preference Models

Plackett-Luce (PL) Model. When a full ranking (permutation) $y_{(1)} \succ y_{(2)} \succ \cdots \succ y_{(n)}$ over n items is available, the PL model (Plackett, 1975) defines its probability as $P_{\text{PL}}(\text{ranking}) = \prod_{i=1}^{n} \frac{u(y_{(i)})}{\sum_{j=i}^{n} u(y_{(j)})}$, where $u(\cdot)$ represents item utilities. This can be conceptualized as sequentially selecting items for n ordered "bins" (ranks), where the probability of selecting an item for the current rank depends on the utilities of the remaining unranked items. PL thus models fine-grained ordinal relationships, contrasting with the binary set partition used in MPO.

Distribution Matching (e.g., InfoNCA). InfoNCA (Chen et al., 2024a) and similar methods align π_θ by matching its output distribution (derived from utilities $u(y_i)$) to a target distribution. For n responses $\{y_i\}$ with scalar quality scores S_i, a target probability $p_{\text{target}}(y_i) = \text{softmax}_i(S_i/\tau)$ is often formed. The model's predicted probability for y_i is $p_{\text{model}}(y_i) = \text{softmax}_i(\log u(y_i))$. The objective then is to minimize the cross-entropy:

$$\mathcal{L}_{\text{InfoNCA}} \propto -\sum_{i=1}^{n} p_{\text{target}}(y_i) \log p_{\text{model}}(y_i). \tag{3}$$

This aims to replicate a detailed reward distribution over individual responses (each potentially a distinct "bin" with its own target weight). We provide the reward margin, a training metric, as well as the downstream performance on AlpacaEval2 for these different related post-training optimization methods in Figure 4. The higher reward margin shows that the distinction between the top chosen and rejected response is higher in MPO vs. other baselines.

4 MPO: Multi-Preference Optimization

Building upon the principles of groupwise preference modeling discussed in Section 3, we introduce Multi-Preference Optimization (MPO), a method designed to align language models using multiple preferred and dispreferred responses per query. MPO generalizes DPO's pairwise comparison to a groupwise contrast using multiple preferred/dispreferred responses per query, enabling richer supervision. We define the MPO objective and its weighted W-MPO variant, which incorporates fine-grained reward signals.

Algorithm 1 Multi-Preference Optimization (MPO)

```
Input: Initial policy $\theta_0$, $\mathcal{D}_{\text{MPO}} = \{(x, \mathcal{Y}_x^+, \mathcal{Y}_x^-)\}$, Reference policy $\pi_{\text{ref}}$, temperature $\beta$, learning-rate $\eta$
Initialise $\theta \leftarrow \theta_0$
for each training iteration do
    for all $(x, \mathcal{Y}_x^+, \mathcal{Y}_x^-) \in \mathcal{D}_{\text{MPO}}$ do
        for $y \in \mathcal{Y}_x^+ \cup \mathcal{Y}_x^-$ do
            $r_\theta(y \mid x) \leftarrow \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$
        end for
        $S_x^+ \leftarrow \sum_{y \in \mathcal{Y}_x^+} \exp\big(\beta\, r_\theta(y \mid x)\big)$
        ▷ Selected responses
        $S_x^{\text{all}} \leftarrow \sum_{y \in \mathcal{Y}_x^+ \cup \mathcal{Y}_x^-} \exp\big(\beta\, r_\theta(y \mid x)\big)$
        $L_x \leftarrow -\log\big(S_x^+ / S_x^{\text{all}}\big)$
        Compute $\nabla_\theta L_x$
        $\theta \leftarrow \theta - \eta\, \nabla_\theta L_x$
    end for
end for
return $\theta$
```

4.1 Problem Setup for MPO

For the MPO framework, we operate on a dataset $\mathcal{D}_{\text{MPO}}$. Each instance in this dataset consists of a query x, a set of n_x^+ chosen (preferred) responses $\mathcal{Y}_x^+ = \{y_{c,1}, \ldots, y_{c,n_x^+}\}$, and a disjoint set of n_x^- rejected (dispreferred) responses $\mathcal{Y}_x^- = \{y_{r,1}, \ldots, y_{r,n_x^-}\}$. These sets $\mathcal{Y}_x^+$ and $\mathcal{Y}_x^-$ can be sourced from direct annotations or derived from finer-grained scalar reward scores, as elaborated for the W-MPO variant (Section 4.4). Our objective is to fine-tune the policy $\pi_\theta(y \mid x)$ using the reference model $\pi_{\text{ref}}(y \mid x)$ to define the implicit log-preference scores $r_\theta(y \mid x)$ (as introduced in Preliminaries, Eq. 3), which are scaled by an inverse temperature $\beta > 0$.

4.2 Base MPO Objective

MPO leverages sets of preferences by modeling the probability $P(\mathcal{Y}_x^+ > \mathcal{Y}_x^- \mid x; \theta, \beta)$ using the groupwise choice principle (Eq. 1 from Preliminaries). For MPO, the utility $u(y)$ of a response y is defined as $u(y) = \exp(\beta \cdot r_\theta(y \mid x))$, where $r_\theta(y \mid x)$ is the implicit log-preference score (Eq. 3). Substituting this utility definition into the general groupwise model (Eq. 1) directly yields the MPO preference probability:

$$P(\mathcal{Y}_x^+ > \mathcal{Y}_x^- \mid x; \theta, \beta) = \frac{\sum_{y \in \mathcal{Y}_x^+} \exp(\beta \cdot r_\theta(y \mid x))}{\sum_{y' \in \mathcal{Y}_x^+ \cup \mathcal{Y}_x^-} \exp(\beta \cdot r_\theta(y' \mid x))}. \tag{4}$$

The MPO objective is to maximize this probability by minimizing its negative log-likelihood:

$$\mathcal{L}_{\text{MPO}}(\theta) = -\mathbb{E}_{(x, \mathcal{Y}_x^+, \mathcal{Y}_x^-) \sim \mathcal{D}_{\text{MPO}}} \left[\log P(\mathcal{Y}_x^+ > \mathcal{Y}_x^- \mid x; \theta, \beta)\right]. \tag{5}$$

Intuitively, this loss function encourages the policy π_θ to assign collectively higher implicit preference scores $r_\theta(y \mid x)$ (and thus higher probabilities $\pi_\theta(y \mid x)$ relative to the reference policy π_{ref}) to the responses in the preferred set $\mathcal{Y}_x^+$ compared to the combined set of preferred and dispreferred responses $\mathcal{Y}_x^+ \cup \mathcal{Y}_x^-$.

4.3 Algorithm for MPO

The training procedure for the core MPO objective is summarized in Algorithm 1. The algorithm iterates through the preference dataset $\mathcal{D}_{\text{MPO}}$. For each data sample, comprising a query x with its associated preferred ($\mathcal{Y}_x^+$) and dispreferred ($\mathcal{Y}_x^-$) response sets, it computes the implicit scores, then the MPO loss term, and updates the policy model π_θ using gradient descent.

4.4 Weighted MPO (W-MPO) Variant

The MPO objective treats all responses within preferred ($\mathcal{Y}_x^+$) or dispreferred ($\mathcal{Y}_x^-$) sets uniformly. However, real-world preference data, often using scalar quality scores $S(y)$, exhibit significant intra-set quality variations. Weighted MPO (W-MPO) is designed to leverage this finer-grained information. By modulating each response's influence based on its deviation from the average quality, W-MPO prioritizes "outlier" responses, creating a data-driven curriculum focused on the most informative examples (further motivation in Appendix C; results in Table 1).

For W-MPO, we assume query x has n_x responses $\mathcal{Y}_x = \{y_1, \ldots, y_{n_x}\}$ with corresponding scalar quality scores $\{S(y_i)\}_{i=1}^{n_x}$. First, we compute the mean score for the query $S_{\text{mean}}(x) = \sum_{i=1}^{n_x} S(y_i)/n_x$. Responses are then partitioned into a preferred set $\mathcal{Y}_x^+ = \{y \in \mathcal{Y}_x \mid S(y) > S_{\text{mean}}(x)\}$ and a dispreferred set $\mathcal{Y}_x^- = \{y \in \mathcal{Y}_x \mid S(y) \leq S_{\text{mean}}(x)\}$. (If $\mathcal{Y}_x^+$ is empty, the sample may be skipped). The *absolute deviation* of each response's score $S(y)$ from this mean is:

$$\Delta W_{\text{abs}}(y) = |S(y) - S_{\text{mean}}(x)|. \tag{6}$$

W-MPO then modifies the logit for each response $y \in \mathcal{Y}_x$ by additively incorporating this absolute deviation:

$$r'_\theta(y \mid x; \alpha_w) = \beta \cdot r_\theta(y \mid x) + \alpha_w \cdot \Delta W_{\text{abs}}(y), \tag{7}$$

where $r_\theta(y \mid x)$ is the base implicit score (Eq. 3) and $\alpha_w > 0$ is a hyperparameter scaling the deviation's impact.

The additive term $\alpha_w \cdot \Delta W_{\text{abs}}(y)$ boosts the logit of outlier responses (those with $S(y)$ far from $S_{\text{mean}}(x)$), thereby increasing their influence. While the base term $\beta \cdot r_\theta(y \mid x)$ and the MPO set-partitioning drive the directional preference (favoring $\mathcal{Y}_x^+$), the absolute deviation term ensures these informative outliers more strongly affect the softmax normalization in the loss.

The W-MPO loss function then uses these modified logits within the MPO framework:

$$\mathcal{L}_{\text{W-MPO}}(\theta) = -\mathbb{E}_{(x, \mathcal{Y}_x, \{S(y)\})}\left[\log\left(\frac{\sum_{y \in \mathcal{Y}_x^+} \exp(r'_\theta(y \mid x; \alpha_w))}{\sum_{y' \in \mathcal{Y}_x^+ \cup \mathcal{Y}_x^-} \exp(r'_\theta(y' \mid x; \alpha_w))}\right)\right]. \tag{8}$$

The expectation is over the original responses and their scores, from which $\mathcal{Y}_x^+$ and $\mathcal{Y}_x^-$ are derived.

W-MPO Robustness via Weighted Contrastive Learning

Unlike methods like GRPO (Shao et al., 2024) that may use signed advantages directly as update targets (ideal with verified rule-based rewards), W-MPO's additive absolute deviation weighting (Eq. 7) is exponentiated within MPO's groupwise contrastive loss (Eq. 8). This optimization is outlier driven and is robust to noise close to median responses, as our experiments demonstrate.

The algorithmic implementation of W-MPO involves calculating these modified logits $r'_\theta(y \mid x; \alpha_w)$ for each response before computing the full loss term, as detailed in Appendix C.3 (Algorithm 2).

4.5 Theoretical Analysis

1. Bias Reduction with Multiple Preferences. Incorporating multiple preferred ($\mathcal{Y}_x^+$) and non-preferred ($\mathcal{Y}_x^-$) responses per query x systematically reduces alignment bias. Alignment bias, in this context, refers to the deviation between the expected characteristics (e.g., helpfulness, adherence to safety guidelines) learned by the model from a finite sample of n responses, and the true characteristics desired from the overall distribution of acceptable responses.

Theorem 1 (Bias Reduction Rate). Under standard assumptions (including finite variance of attributes, independent sampling, sufficient model capacity, and uniform bounded variance across queries), the expected alignment bias $\mathbb{E}[B^{(n)}]$ when using n preferred samples per query decreases as:

$$\mathbb{E}[B^{(n)}] = \tilde{O}\left(\frac{C}{\sqrt{n}}\right), \tag{9}$$

where C is a constant related to the maximum attribute variance.

This result theoretically grounds the benefit of leveraging multiple preferences, as MPO does, over single pairwise comparisons. Detailed assumptions and proof are presented in Appendix D.

2. Stationary Point of the MPO Loss. A crucial property of the MPO loss function (Eq. 5) concerns its optimization landscape. We find that at its stationary points, the model learns to assign vanishing probabilities to non-preferred responses.

Method	**Mistral-Base (7B)**				**Llama-3-Base (8B)**			
	LC (%)	WR (%)	Arena-Hard	MT-Bench	LC (%)	WR (%)	Arena-Hard	MT-Bench
SFT[1]	8.4	6.2	1.3	6.3	6.2	4.6	3.3	6.6
RRHF[1]	11.6	10.2	5.8	6.7	12.1	10.1	6.3	7.0
SLiC-HF[1]	10.9	8.9	7.3	**7.4**	12.3	13.7	6.0	7.6
IPO[1]	10.8	9.0	7.5	7.0	14.4	14.2	17.8	7.4
CPO[1]	9.8	8.9	6.7	6.9	10.8	8.1	5.8	7.4
KTO[1]	13.1	9.1	5.6	6.8	14.2	12.4	12.5	**7.8**
ORPO[1]	14.7	12.2	7.0	7.0	12.2	10.6	10.8	7.6
R-DPO[1]	17.4	12.8	10.5	7.0	17.6	14.4	17.2	7.5
GRPO	10.1	6.7	6.3	6.8	8.2	5.7	6.0	7.1
DPO	16.6	13.8	12.7	6.7	16.9	14.1	**18.5**	7.7
NC2-DPO	14.9	11.7	8.8	6.9	16.3	13.1	13.7	7.5
InfoNCA	14.8	10.8	9.7	7.0	15.9	12.9	14.8	7.6
MPO 1vsk	15.2	11.5	10.1	7.1	17.3	13.5	15.9	7.6
MPO	18.4	14.4	**13.2**	7.2	18.4	15.1	18.4	7.5
W-MPO	**20.3**	**14.9**	12.8	7.3	**20.1**	**15.6**	**18.5**	**7.8**

Table 1: Comparison of methods on Mistral-Base and Llama-Base models in the off-policy setting with Ultrafeedback prompts and responses.

Lemma 1 (Vanishing Probability for Non-Preferred Responses). For the MPO loss function, a stationary point is achieved as the model probabilities for all responses y_i in the non-preferred set $\mathcal{Y}^-$ approach zero, i.e., $P_\theta(y_i \mid x) \to 0$ for all $y_i \in \mathcal{Y}^-$.

This contrasts with objectives like InfoNCA that aim to match a target distribution over all responses. This property encourages MPO to learn a clearer distinction between preferred and non-preferred sets. Detailed proof and characterization are provided in Appendix F.

5 Experimental Setup

We evaluate MPO across three training regimes: Offline, Online, and Iterative, using Mistral-7B and Llama-3-8B base models. Further details, including baselines, are in Appendix I.

Off-policy Setting. Following Tunstall et al. (2023), we first create supervised fine-tuned (SFT) models using UltraChat-200k (Ding et al., 2023). These SFT models initialize MPO and other methods for preference optimization on the UltraFeedback dataset (Cui et al., 2023).

On-policy Setting. We start with instruction-tuned models (Llama-3-8B-Instruct, Mistral-7B-Instruct). To align these models on preferences, we generate 5 responses per UltraFeedback prompt (temperature 1.0) using the instruction-tuned models themselves, similar to Meng et al. (2024). Responses are scored by Skywork-Reward-Llama-3.1-8B Liu et al. (2024b); the top-2 and bottom-2 are selected as preferred ($\mathcal{Y}^+$) and rejected ($\mathcal{Y}^-$) sets for MPO.

Iterative Setting Using the same instruction-tuned base models, we adopt an iterative alignment framework akin to SPPO (Wu et al., 2024). The UltraFeedback prompt set is split for multi-round fine-tuning, with fresh responses (5 per prompt, temp 1.0, scored by Skywork-Reward-Llama-3.1-8B, top-2/bottom-2 selected) generated each round, following Wu et al. (2024).

We provide more details about our experimental setup in Appendix H and I.

6 Experimental Results

SOTA Performance We evaluate MPO against a broad spectrum of preference optimization baselines in three distinct training regimes: offline, online, and iterative. Across all settings, MPO consistently sets a new state-of-the-art. In the offline setup, as seen in Table 1, MPO outperforms established baselines such as R-DPO (Rafailov et al., 2024), InfoNCA (Chen et al., 2024a), and KTO (Ethayarajh et al., 2024). Notably, W-MPO achieves top scores.

---

[1] These are taken directly from the paper SIMPO: Simple Preference Optimization with a Reference-Free Reward (Meng et al., 2024)

[2] These are taken directly from the paper SPPO: Self-Play Preference Optimization for Language Model Alignment (Wu et al., 2024)

Method	**Mistral-Instruct (7B)**				**Llama-3-Instruct (8B)**			
	LC (%)	WR (%)	Arena-Hard	MT-Bench	LC (%)	WR (%)	Arena-Hard	MT-Bench
SFT	17.1	14.7	12.6	7.5	26.0	25.3	22.3	8.1
RRHF[1]	25.3	24.8	18.1	7.6	31.3	28.4	26.5	7.9
SLiC-HF[1]	24.1	24.6	18.9	**7.8**	26.9	27.5	26.2	8.1
DPO[1]	26.8	24.9	16.3	7.6	40.3	37.9	32.6	8.0
IPO[1]	20.3	20.3	16.2	**7.8**	35.6	35.6	30.5	**8.3**
CPO[1]	23.8	28.8	22.6	7.5	28.9	32.2	28.8	8.0
KTO[1]	24.5	23.6	17.9	7.7	33.1	31.8	26.4	8.2
ORPO[1]	24.5	24.9	20.8	7.7	28.5	27.4	25.8	8.0
R-DPO[1]	27.3	24.5	16.1	7.5	41.1	37.8	33.1	8.0
MPO	**28.2**	**29.4**	**22.7**	7.7	**49.0**	**50.6**	**46.2**	8.2

Table 2: Results on Mistral-Instruct and Llama-Instruct models in the on-policy setting with Ultrafeedback prompts.

Method	**Mistral-Instruct (7B)**				**Llama-3-Instruct (8B)**			
	LC (%)	WR (%)	Arena-Hard	MT-Bench	LC (%)	WR (%)	Arena-Hard	MT-Bench
SPPO[2] (Iteration 1)	24.79	23.51	18.7	7.21	31.73	31.74	34.8	8.1
MPO (Iteration 1)	**26.01**	**26.81**	**22.1**	**7.64**	**44.56**	**47.83**	**42.8**	**8.2**
SPPO[2] (Iteration 2)	26.89	27.62	20.4	7.49	35.15	35.98	38.0	8.2
MPO (Iteration 2)	**28.15**	**31.05**	**23.3**	**7.68**	**48.96**	**52.3**	**45.6**	**8.3**
SPPO[2] (Iteration 3)	28.53	31.02	23.3	7.59	38.78	39.85	40.1	8.2
MPO (Iteration 3)	**33.42**	**36.93**	**23.9**	**7.71**	**53.75**	**57.65**	**49.2**	**8.4**

Table 3: Results on Mistral-Instruct and Llama-Instruct models in the iterative on-policy setting with Ultrafeedback prompts.

In the online setup in Table 2, preference optimization is performed on instruction-tuned models. MPO again achieves SOTA performance across all benchmarks and model families. We also evaluate MPO in the iterative setting in Table 3. Compared against SPPO (Wu et al., 2024), the strongest prior iterative baseline, MPO yields significant improvements in every round.

Importance of suboptimal responses in MPO Traditional preference optimization methods often underutilize the rich feedback signal available by relying solely on a single best-worst pair per query. However, models can produce diverse suboptimal responses, and learning from this can improve alignment. In Figure 5, we present results for `Llama-8B-Instruct` in the on-policy setting on AlpacaEval 2. We observe consistent gains as the number of responses per query increases from n=2 to n=8 under MPO, highlighting the benefits of leveraging multiple responses during training.

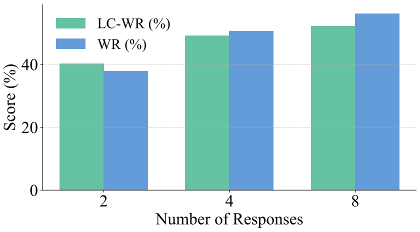

Figure 5: Downstream performance with number of model responses on Alpaca Eval2.

Key Takeaway: *Including additional sub-optimal responses in* MPO *improves policy learning, challenging the convention that only top-ranked answers drive alignment performance.*

Impact of deviation-based weighting in W-MPO We evaluate the impact of W-MPO's core deviation-based weighting by comparing it to MPO. Table 4 underscores that deviation-aware weighting helps the policy learn from more separable supervision signals, leading to improved alignment.

Method	**Mistral (7B)**		**Llama-3 (8B)**	
	LC (%)	**WR (%)**	**LC (%)**	**WR (%)**
MPO	28.2	29.4	49.0	50.6
W-MPO	**29.5**	**30.6**	**52.1**	**52.5**

Table 4: Comparison of MPO and W-MPO on AlpacaEval2 for Mistral and Llama Instruct models.

Key Takeaway: MPO *supports the hypothesis that deviation based contrast amongst responses – not just their absolute reward – plays a critical role in preference optimization.*

MPO achieves superior gains even with limited data We examine how MPO performs under data-constrained settings for Mistral (7B), compared to DPO across varying proportions of training data. As shown in Figure 6, MPO outperforms DPO at varying percentages of training data. For instance, on LC-WR (%), MPO at 25% training data outperforms DPO at 50%. These improvements are a direct result of MPO's ability to leverage multiple responses per query, extracting more supervision signal.

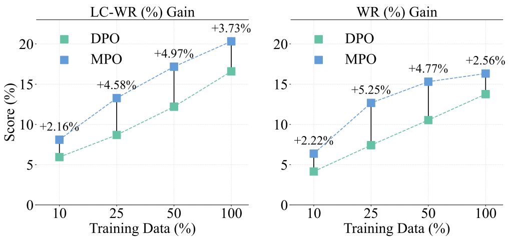

Figure 6: Comparison of MPO and DPO across varying training data sizes on AlpacaEval2.

Key Takeaway: Even with limited training data, MPO *consistently outperforms* DPO *by effectively utilizing multiple responses per query – highlighting the importance of multiple responses per query, over multiple queries, at a constant (query, response) budget.*

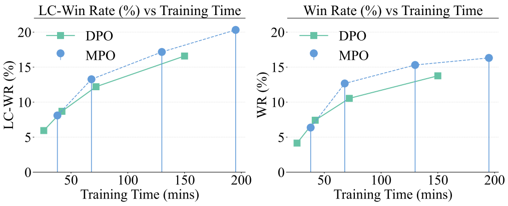

Figure 7: Comparison of MPO and DPO on AlpacaEval2 under fixed training-time budgets.

Efficiency of MPO under fixed training budgets While MPO naturally incurs longer per-step training time because it processes multiple responses per query, we investigate whether this cost translates into superior performance within a fixed wall-clock budget. Figure 7 shows that, despite the higher computation per batch, MPO consistently outperforms DPO at every time checkpoint, achieving substantial gains in both LC-WR and overall WR for Mistral (7B).

Key Takeaway: Even under fixed training time, MPO *outperforms* DPO, *showing its greater sample efficiency and stronger use of richer preference signals.*

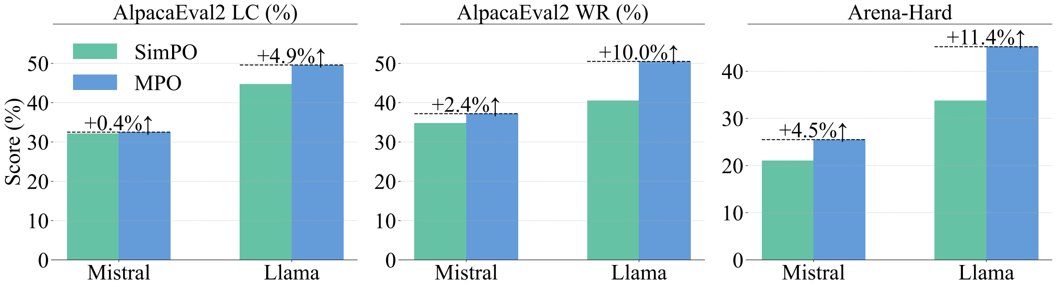

Figure 8: Evaluating MPO (Reference-Free) vs SIMPO on Mistral and Llama instruct models.

Unifying DPO variants under the MPO setting To demonstrate the generality, we explore how MPO can be instantiated on top of existing DPO-based methods, such as KTO (Ethayarajh et al., 2024), R-DPO (Park et al., 2024), and SimPO (Meng et al., 2024). These methods share the same pairwise contrastive structure as DPO, differing primarily in how they weigh or define the reward margin. Since MPO generalizes DPO to groupwise preference comparisons, it is naturally compatible with any loss that depends on contrast over preference data.

In Figure 8, we highlight a concrete instantiation: MPO (Reference-Free), where we extend SimPO's reward-based contrastive loss to operate in the multi-preference regime. This variant benefits from MPO's ability to better utilize responses. We observe consistent improvements across evaluation on downstream datasets. The results indicate that MPO (Reference-Free) leverages responses more effectively than SimPO.

Key Takeaway: MPO *provides a principled generalization of* DPO*-style contrastive objectives, allowing broader preference integration without altering the base reward design.*

References

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1.

Azar, M. G., Guo, Z. D., Piot, B., Munos, R., Rowland, M., Valko, M., and Calandriello, D. (2024). A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

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

Chen, H., He, G., Yuan, L., Cui, G., Su, H., and Zhu, J. (2024a). Noise contrastive alignment of language models with explicit rewards. *arXiv preprint arXiv:2402.05369*.

Chen, Y., Tan, J., Zhang, A., Yang, Z., Sheng, L., Zhang, E., Wang, X., and Chua, T.-S. (2024b). On softmax direct preference optimization for recommendation. *arXiv preprint arXiv:2406.09215*.

Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. (2024c). Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Cheng, P., Yang, Y., Li, J., Dai, Y., and Du, N. (2023). Adversarial preference optimization. *arXiv preprint arXiv:2311.08045*.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., and Sun, M. (2023). Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Ding, N., Chen, Y., Xu, B., Qin, Y., Zheng, Z., Hu, S., Liu, Z., Sun, M., and Zhou, B. (2023). Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., Shum, K., and Zhang, T. (2023). Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

Dubois, Y., Galambosi, B., Liang, P., and Hashimoto, T. B. (2024). Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. (2024). Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Farebrother, J., Orbay, J., Vuong, Q., Taïga, A. A., Chebotar, Y., Xiao, T., Irpan, A., Levine, S., Castro, P. S., Faust, A., Kumar, A., and Agarwal, R. (2024). Stop regressing: Training value functions via classification for scalable deep rl.

Gupta, T., Madhavan, R., Zhang, X., Bansal, C., and Rajmohan, S. (2025). Ampo: Active multi-preference optimization. *arXiv preprint arXiv:2502.18293*.

Hong, J., Lee, N., and Thorne, J. (2024). Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142.

Kim, D., Kim, Y., Song, W., Kim, H., Kim, Y., Kim, S., and Park, C. (2024). sdpo: Don't use your data all at once. *arXiv preprint arXiv:2403.19270*.

Klein, T. and Nabi, M. (2024). Contrastive perplexity for controlled generation: An application in detoxifying large language models. *arXiv preprint arXiv:2401.08491*.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. (2024a). Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Liu, C. Y., Zeng, L., Liu, J., Yan, R., He, J., Wang, C., Yan, S., Liu, Y., and Zhou, Y. (2024b). Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Liu, J., Zhou, Z., Liu, J., Bu, X., Yang, C., Zhong, H.-S., and Ouyang, W. (2024c). Iterative length-regularized direct preference optimization: A case study on improving 7b language models to gpt-4 level. *arXiv preprint arXiv:2406.11817*.

Liu, T., Zhao, Y., Joshi, R., Khalman, M., Saleh, M., Liu, P. J., and Liu, J. (2023). Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Luce, R. D. et al. (1959). *Individual choice behavior*, volume 4. Wiley New York.

Meng, Y., Xia, M., and Chen, D. (2024). Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Ng, A. Y., Russell, S., et al. (2000). Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Pang, R. Y., Yuan, W., Cho, K., He, H., Sukhbaatar, S., and Weston, J. (2024). Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*.

Park, R., Rafailov, R., Ermon, S., and Finn, C. (2024). Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.

Plackett, R. L. (1975). The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202.

Qi, B., Li, P., Li, F., Gao, J., Zhang, K., and Zhou, B. (2024). Online dpo: Online direct preference optimization with fast-slow chasing. *arXiv preprint arXiv:2406.05534*.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2024). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2015). Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Schulman, J. (2015). Trust region policy optimization. *arXiv preprint arXiv:1502.05477*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Song, F., Yu, B., Li, M., Yu, H., Huang, F., Li, Y., and Wang, H. (2024). Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Sun, S., Zhang, Y., Bukharin, A., Mosallanezhad, D., Zeng, J., Singhal, S., Shen, G., Renduchintala, A., Konuk, T., Dong, Y., et al. (2025). Reward-aware preference optimization: A unified mathematical framework for model alignment. *arXiv preprint arXiv:2502.00203*.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Tang, X., Yoon, S., Son, S., Yuan, H., Gu, Q., and Bogunovic, I. (2025). Game-theoretic regularized self-play alignment of large language models. *arXiv preprint arXiv:2503.00030*.

Tang, Y., Guo, Z. D., Zheng, Z., Calandriello, D., Munos, R., Rowland, M., Richemond, P. H., Valko, M., Pires, B. Á., and Piot, B. (2024). Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., et al. (2023). Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Wu, Y., Sun, Z., Yuan, H., Ji, K., Yang, Y., and Gu, Q. (2024). Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*.

Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Durme, B. V., Murray, K., and Kim, Y. J. (2024). Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. *ArXiv*, abs/2401.08417.

Yuan, W., Kulikov, I., Yu, P., Cho, K., Sukhbaatar, S., Weston, J., and Xu, J. (2024). Following length constraints in instructions. *arXiv preprint arXiv:2406.17744*.

Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., and Huang, F. (2023). Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

Zeng, Y., Liu, G., Ma, W., Yang, N., Zhang, H., and Wang, J. (2024). Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

SUPPLEMENTARY MATERIALS

These supplementary materials provide additional details, derivations, and experimental results for our paper. The appendix is organized as follows:

- Section A presents a detailed related works section.
- Section B presents the core limitations of this work, and some avenues for future work.
- Section C presents a detailed overview of a variant of our main method which is weighted MPO.
- Section D presents a detailed bias analysis, demonstrating how incorporating multiple preferences reduces alignment bias.
- Section E provides a comprehensive comparison between the Group Contrastive Loss and InfoNCA Loss, including detailed gradient analyses.
- Section F offers a thorough characterization of stationary points for both the InfoNCA and Weighted Contrastive Loss functions.
- Section H describes the baselines used for comparison in our experimental evaluations, including various DPO implementations and alternative approaches.
- Section I describes the models, dataset and compute details for our experiments.
- Section J provides the implementation details of the reward loss computation, including the actual code used in our experiments.
- Section K provides examples of model responses using our model as compared with the base model.

A Related Work

We will start this literature survey with a high level overview of the RLHF literature and then going deeper into the area of preference, and then multi-preference optimization relevant to our work.

Broader RLHF Literature: Reinforcement Learning through Human feedback (RLHF) has emerged as a robust alignment algorithm for language models. The area broadly started of with works like Trust Region Policy Optimization (TRPO), and Proximal Policy Optimization (PPO) (Schulman, 2015; Schulman et al., 2017) which extend direct RL based methods by constraining the update space to within a trusted region and clipping policy updates to prevent instability respectively. Building upon earlier policy gradient methods (Sutton et al., 1999), PPO has been successfully applied to alignment tasks in Reinforcement Learning from Human Feedback (RLHF), allowing language models to produce outputs aligned with human preferences (Ziegler et al., 2019; Ouyang et al., 2022). Its simplicity and efficiency make it a standard approach for fine-tuning large-scale models. Prior to PPO, Trust Region Policy Optimization (TRPO) (Schulman, 2015) introduced constraints to improve learning stability, influencing the development of PPO. Early applications of policy gradient methods in natural language processing (Ranzato et al., 2015) demonstrated the potential of reinforcement learning for language model training.

Preference Optimization: Direct Preference Optimization (DPO) simplifies the alignment of language models by optimizing a contrastive loss directly over paired preference data, bypassing the intermediate step of reward modeling (Rafailov et al., 2024). Unlike RLHF, DPO does not require explicit reward functions, making it computationally efficient and suitable for limited preference datasets. Recent extensions of DPO, such as Identity Preference Optimization (IPO) (Azar et al., 2024), self-play preference optimization (Wu et al., 2024), preference ranking optimization (Song et al., 2024), rejection sampling optimization (Liu et al., 2023), and generalized preference optimization (Tang et al., 2024) are amongst the other recent works improve on the DPO method.

Beyond the foundational pairwise approaches and their direct extensions, numerous recent works have proposed methods that adapt or refine DPO-like strategies, often eliminating the need for separate reward modeling or reference models.

Alternative Approaches Without Full Reward Modeling. Ethayarajh et al. (2024) propose **KTO**, a framework inspired by prospect theory that directly learns whether a response is globally desirable or undesirable, thereby removing the requirement of having multiple positive examples per instruction. Zeng et al. (2024) focus on token-level alignment in **TDPO**, imposing forward KL divergence constraints for each token rather than solely for the final output. This fine-grained approach can mitigate the mode-collapse issues sometimes observed in sequence-level alignment. Meanwhile, Dong et al. (2023) introduce a list-wise method called **RAFT**, where the model finetunes on the best response from each sampled set of k candidates, iteratively converging toward an optimal subset policy. By contrast, Yuan et al. (2023) center on rank-based supervision through **RRHF**, which combines a rank loss with standard supervised signals to ensure the model maintains stronger probabilities on higher-ranked (i.e., better) responses and less on suboptimal responses.

Enhancing DPO with Additional Objectives and Training Schemes. Other works further modify or reinterpret the DPO loss to incorporate new constraints or to remove the need for a reference model. Chen et al. (2024c) propose **SPIN**, which treats the model as part of a two-player adversarial game, obviating separate reward modeling by training with a discriminator that distinguishes human from machine responses. **CPO** (Xu et al., 2024) reworks the DPO objective by removing the reference-model term and adding a behavior cloning regularizer. Similarly, **ORPO** (Hong et al., 2024) folds preference optimization into a negative log-likelihood objective via an odds-ratio penalty, thereby unifying supervised fine-tuning (SFT) and preference training. In **SimPO**, Meng et al. (2024) remove the reference model and incorporate a length normalization to address verbosity issues that can skew preference data. Likewise, **R-DPO** (Park et al., 2024) and **LD-DPO** (Liu et al., 2024c) specifically tackle length bias by injecting additional regularizers or by explicitly separating length-based preferences from other factors. For instance, LD-DPO modifies the training set to handle length constraints, preventing performance drops on standard benchmarks while mitigating length exploitation in preference tasks.

Refining Training Regimens for Preference Data. A final family of works emphasizes how training procedures or data usage can be systematically improved. For instance, Kim et al. (2024) propose **sDPO**, a step-wise learning method partitioning preference data to stabilize training. **IRPO** (Pang et al., 2024) enhances chain-of-thought reasoning by incorporating a negative log-likelihood term for the chosen solution path, thus nudging LLMs toward robust multi-step reasoning. **OFS-DPO** (Qi et al., 2024) trains two LoRA modules at different paces—one faster, one slower—to sustain gradient momentum and to adapt more efficiently. Lastly, Yuan et al. (2024) tackle verbosity with **LIFT-DPO**, an approach that augments preference data with length-control instructions, ensuring that the model does not exploit response length to inflate its preference scores.

Multi-Preference Optimization: Traditional preference optimization methods, like DPO, consider pairwise comparisons. However, datasets such as UltraFeedback (Cui et al., 2023) highlight the necessity of multi-preference optimization. Multi-preference methods, such as InfoNCA (Chen et al., 2024a), leverage all available positive and negative responses simultaneously, reducing alignment bias and better approximating the true preference distribution. These methods mitigate limitations inherent to pairwise approaches by incorporating the diversity of acceptable and suboptimal responses. Earlier works in search have also used multiple user preferences to optimize models in various applications such as search (Joachims, 2002).

Reward Modeling in Preferences: Reward modeling is essential for translating qualitative human feedback into quantitative metrics that guide AI behavior optimization. Traditional methods, such as Reinforcement Learning from Human Feedback (RLHF), utilize reward models trained on human annotations to inform policy updates (Christiano et al., 2017; Stiennon et al., 2020). Early approaches like inverse reinforcement learning (Ng et al., 2000) and apprenticeship learning (Abbeel and Ng, 2004) demonstrated the feasibility of inferring reward functions from observed behaviors. Recent advancements have diversified reward modeling techniques. For instance, the Adversarial Preference Optimization (APO) framework employs adversarial training to adapt reward models to the evolving generation distribution of language models (Cheng et al., 2023).

Noise Contrastive Estimation and InfoNCA: Contrastive learning, particularly methods like InfoNCE (Oord et al., 2018), maximizes mutual information between positive samples while discriminating against negatives. In the language domain, Klein and Nabi (2024) leverage a perplexity-based contrastive objective to reduce toxic language generation while preserving the model's overall utility. InfoNCA adapts these principles for preference optimization, aligning responses with scalar rewards through noise-contrastive estimation (Chen et al., 2024a). Despite its strengths, InfoNCA can overemphasize less informative negative samples, which motivates methods like MPO that dynamically weigh responses based on deviation from the mean reward.

UltraFeedback Dataset: The UltraFeedback dataset (Cui et al., 2023) is a significant advancement in preference-based training resources. It comprises GPT-4 annotated feedback for over 64,000 instructions, including scalar reward evaluations. UltraFeedback has been pivotal in developing models like UltraLM-13B-PPO and UltraRM, which achieve state-of-the-art performance across benchmarks such as AlpacaEval. This dataset's granularity enables advanced preference optimization methods like MPO to leverage diverse response quality levels effectively.

B Limitations

While MPO and W-MPO demonstrate strong empirical performance and offer theoretical benefits, we identify several areas for further exploration and acknowledge certain design considerations:

B.1 Reward Signal Quality and Partitioning

The partitioning into preferred ($\mathcal{Y}^+$) and dispreferred ($\mathcal{Y}^-$) sets, along with the deviation calculation in W-MPO, utilizes input scalar reward scores $S(y)$. While W-MPO's absolute deviation weighting and the underlying contrastive nature of MPO are designed for greater robustness compared to methods directly regressing on or using signed advantages from potentially noisy reward models, the initial quality of $S(y)$ still influences these preliminary steps. Significant miscalibration or systemic biases in the reward source could affect the optimality of the partitions. Future research could explore adaptive partitioning mechanisms or methods to integrate reward uncertainty more directly.

B.2 Computational Scaling with Extremely Large Response Sets

MPO and W-MPO process all N_x responses available for a given query to compute the loss. This comprehensive approach enables richer supervision than pairwise DPO. However, for scenarios generating an extremely large number of candidate responses per query (e.g., hundreds), the per-step computational demand might increase. Investigating efficient subset selection techniques that identify the most informative responses from very large pools, while retaining the benefits of the MPO framework, presents a valuable avenue for future work to enhance scalability further.

B.3 Exploring Richer Preference Structures Beyond Binary Contrast

The current MPO framework establishes a robust binary contrast between a single "accepted" set $\mathcal{Y}^+$ and a single "rejected" set $\mathcal{Y}^-$. While this generalization of DPO is effective, human preferences can often be more nuanced. Future work could investigate extending MPO to accommodate multiple ordered preference sets (e.g., $\mathcal{Y}^{++} > \mathcal{Y}^+ > \mathcal{Y}^- > \mathcal{Y}^{--}$) or a finer spectrum of preference levels. This might involve further generalizations of the underlying groupwise Bradley-Terry model to capture such richer, multi-level preference structures.

Despite these considerations, MPO and W-MPO offer a robust and effective advancement for leveraging multi-preference data in LLM alignment.

C Detailed Formulation of Weighted MPO (W-MPO)

While the core Multi-Preference Optimization (MPO) framework, as presented in Section 4, effectively leverages sets of preferred ($\mathcal{Y}_x^+$) and dispreferred ($\mathcal{Y}_x^-$) responses, it treats all responses within each set with uniform importance. However, many modern alignment pipelines, particularly those employing on-policy or iterative generation and evaluation, utilize powerful reward models (RMs) that provide fine-grained scalar quality scores $S(y)$ for each generated response y. These scores often reveal significant variance in quality even within the pre-defined preferred or dispreferred sets. Weighted MPO (W-MPO) is designed to exploit this richer, quantitative feedback.

C.1 Motivation for Weighted MPO

In contemporary LLM alignment, especially in on-policy settings (e.g., RLHF with a learned reward model, or iterative DPO-like methods where a model generates responses that are then scored), obtaining exhaustive human pairwise preferences for all generated outputs can be prohibitively expensive or slow. Instead, an automated reward model $RM(y \mid x)$ is often used to assign a scalar quality score $S(y)$ to each response y generated for a prompt x. These scalar scores $S(y)$ offer several advantages:

1. **Granularity:** They provide a more nuanced measure of quality than simple binary (chosen/rejected) or pairwise preferences. A response y_1 might be preferred over y_2, but $S(y_1)$ can also indicate *how much* better it is, or how it compares to a global quality scale.
2. **Efficiency:** RMs can score a large volume of responses quickly, enabling larger-scale and more frequent preference data collection, crucial for iterative alignment.
3. **Handling Multiple Responses:** When multiple ($N_x > 2$) responses are generated per prompt (e.g., via diverse beam search, multiple sampling seeds, or from different model variants), RMs can score all of them, providing a rich landscape of quality.

The core MPO method, by operating on sets $\mathcal{Y}_x^+$ and $\mathcal{Y}_x^-$ (which might themselves be derived from $S(y)$ via a threshold like $S_{\text{mean}}(x)$), does not fully capitalize on the information that some responses in $\mathcal{Y}_x^+$ are "more preferred" (higher $S(y)$) than others, or that some in $\mathcal{Y}_x^-$ are "more dispreferred" (lower $S(y)$). Responses that are significant outliers in terms of their quality scores (i.e., those deviating most from an average or baseline quality) are often the most informative for training. For instance, an exceptionally good response provides a strong positive signal, while a particularly problematic response offers a clear negative signal to learn from.

W-MPO aims to incorporate this finer-grained information by weighting the contribution of each response y in the MPO loss based on its scalar score $S(y)$. Specifically, it uses the deviation of $S(y)$ from a reference point (e.g., the mean score $S_{\text{mean}}(x)$ for the current prompt) to modulate the response's impact. This allows W-MPO to:

- **Prioritize Informative Samples:** Give more importance to responses that are either exceptionally good or exceptionally bad, as these are strong learning signals.
- **Implement a Data-Driven Curriculum:** Naturally focus the model's attention on responses from which it can learn the most about the boundaries of desired behavior.
- **Enhance Stability and Robustness:** By considering the full spectrum of quality scores, W-MPO can potentially lead to more stable and robust alignment compared to methods that only consider coarse preference signals.

This approach is particularly relevant when $S(y)$ comes directly from a reward model, as it allows the alignment process to more faithfully reflect the nuances captured by that RM.

C.2 W-MPO Formulation

Let $\mathcal{D}_{\text{W-MPO}}$ be a dataset where each instance consists of a query x, a set of N_x responses $\mathcal{Y}_x = \{y_1, \ldots, y_{N_x}\}$ for that query, and their corresponding scalar quality scores $\{S(y_1), \ldots, S(y_{N_x})\}$.

Response Partitioning and Deviation Calculation. First, for each query x, we calculate the mean quality score:

$$S_{\text{mean}}(x) = \frac{1}{N_x} \sum_{j=1}^{N_x} S(y_j). \tag{10}$$

The responses are then partitioned into preferred and dispreferred sets:

$$\mathcal{Y}_x^+ = \{y \in \mathcal{Y}_x \mid S(y) > S_{\text{mean}}(x)\}, \tag{11}$$

$$\mathcal{Y}_x^- = \{y \in \mathcal{Y}_x \mid S(y) \leq S_{\text{mean}}(x)\}. \tag{12}$$

If $\mathcal{Y}_x^+$ is empty, the sample for query x is typically skipped.

Next, we define a signed *deviation term* $\Delta W(y)$ for each response $y \in \mathcal{Y}_x$:

$$\Delta W(y) = S(y) - S_{\text{mean}}(x). \tag{13}$$

This term, $\Delta W(y)$, is positive for $y \in \mathcal{Y}_x^+$ and non-positive for $y \in \mathcal{Y}_x^-$. It quantifies how much better or worse a response is compared to the average quality for that query.

Additive Logit Adjustment. W-MPO incorporates this deviation $\Delta W(y)$ by additively adjusting the logit (scaled implicit preference score) of each response. The base implicit score is $r_\theta(y \mid x) = \log(\pi_\theta(y \mid x)/\pi_{\text{ref}}(y \mid x))$, as defined in Eq. 3. The modified logit, $r'_\theta(y \mid x)$, used in W-MPO is:

$$r'_\theta(y \mid x; \alpha_w) = \beta \cdot r_\theta(y \mid x) + \alpha_w \cdot \Delta W(y), \tag{14}$$

where β is the inverse temperature scaling the base DPO-like score, and α_w is a hyperparameter controlling the magnitude of the deviation-based adjustment. A positive α_w will increase the effective logit for responses with $S(y) > S_{\text{mean}}(x)$ and decrease it for responses with $S(y) < S_{\text{mean}}(x)$, with the magnitude of adjustment proportional to $|\Delta W(y)|$.

W-MPO Loss Function. The W-MPO loss function retains the structure of the core MPO loss (Eq. 5) but utilizes these modified logits $r'_\theta(y \mid x; \alpha_w)$:

$$\mathcal{L}_{\text{W-MPO}}(\theta) = -\mathbb{E}_{(x,\mathcal{Y}_x,\{S(y)\})\sim\mathcal{D}_{\text{W-MPO}}}\left[\log\left(\frac{\sum_{y\in\mathcal{Y}_x^+}\exp(r'_\theta(y \mid x; \alpha_w))}{\sum_{y'\in\mathcal{Y}_x^+\cup\mathcal{Y}_x^-}\exp(r'_\theta(y' \mid x; \alpha_w))}\right)\right]. \tag{15}$$

This additive adjustment in the logit space (Eq. 14) is equivalent to applying a multiplicative exponential weight $w_{\text{exp}}(y) = \exp(\alpha_w \cdot \Delta W(y))$ to the original base strength term $\exp(\beta \cdot r_\theta(y \mid x))$ within the sums of the MPO loss. The W-MPO objective thus encourages the policy π_θ to more strongly prefer/disprefer responses that are further above/below the mean quality score, respectively.

C.3 Algorithm for W-MPO

The detailed training procedure for W-MPO is presented in Algorithm 2.

D Bias Analysis:

In the first part of this section section, we analyze how the number of positive and negative examples per query, k, affects the bias with respect to an attribute $a(y)$. We provide a formal theorem establishing the relationship between bias and k, followed by a corollary discussing the behavior as $k \to \infty$.

The reason for this analysis is to show that multi-preference sampling of accepted and rejected answers from a distribution is better than using a single sample as DPO does. The more accepted and rejected samples you have, the lower the bias, provably.

D.1 Assumptions

We make the following assumptions:

1. **Attribute Function**: Let $a(y) : \mathcal{Y}_x \to \mathbb{R}$ be an attribute function mapping responses to real numbers (e.g., response length).
2. **Finite Variance**: The attribute $a(y)$ has finite variance over the acceptable response distribution $\mathcal{A}_x$ for each query x, i.e., $\text{Var}_{y\sim\mathcal{A}_x}[a(y)] = \sigma^2_{\mathcal{A}_x} < \infty$.
3. **Independent Sampling**: Responses are independently sampled from their respective distributions.
4. **Model Capacity**: The model can represent the true distribution given sufficient data.
5. **Uniform Bounded Variance**: There exists a constant $\sigma^2_{\max}$ such that $\sigma^2_{\mathcal{A}_x} \le \sigma^2_{\max}$ for all $x \in \mathcal{X}$.

Algorithm 2 Detailed Weighted Multi-Preference Optimization (W-MPO)

```
Input: Initial policy parameters $\theta_0$; dataset $\mathcal{D}_{\text{W-MPO}}$, where each sample contains $(x, \mathcal{Y}_x, \{S(y)\}_{y \in \mathcal{Y}_x})$; reference model $\pi_{\text{ref}}$; inverse temperature $\beta$; W-MPO weight scaling hyperparameter $\alpha_w$; learning rate $\eta$; number of training iterations/epochs $T$.
Output: Optimized model parameters $\theta_T$.
Initialize policy model parameters $\theta \leftarrow \theta_0$.
for $t = 1$ to $T$ do
    Initialize total loss for epoch/iteration $\mathcal{L}_{\text{epoch}} \leftarrow 0$.
    for all sample $(x, \mathcal{Y}_x, \{S(y)\}_{y \in \mathcal{Y}_x})$ in $\mathcal{D}_{\text{W-MPO}}$ (typically processed in mini-batches) do
        Compute $S_{\text{mean}}(x) \leftarrow \frac{1}{|\mathcal{Y}_x|} \sum_{y_j \in \mathcal{Y}_x} S(y_j)$. (Ref. Eq. 10)
        Define $\mathcal{Y}_x^+ \leftarrow \{y \in \mathcal{Y}_x \mid S(y) > S_{\text{mean}}(x)\}$. (Ref. Eq. 11)
        Define $\mathcal{Y}_x^- \leftarrow \{y \in \mathcal{Y}_x \mid S(y) \leq S_{\text{mean}}(x)\}$. (Ref. Eq. 12)
        if $|\mathcal{Y}_x^+| = 0$ then
            continue {Skip sample if no preferred responses after partitioning}
        end if
        Let $\mathcal{Y}_{\text{all}} = \mathcal{Y}_x^+ \cup \mathcal{Y}_x^-$.
        Create a map $R'_\theta$ to store modified logits.
        for $y \in \mathcal{Y}_{\text{all}}$ do
            Compute base score: $r_\theta(y \mid x) \leftarrow \log\left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)$. (Ref. Eq. 3)
            Compute deviation term: $\Delta W(y) \leftarrow S(y) - S_{\text{mean}}(x)$. (Ref. Eq. 13)
            Compute modified logit: $r'_\theta(y \mid x) \leftarrow \beta \cdot r_\theta(y \mid x) + \alpha_w \cdot \Delta W(y)$. (Ref. Eq. 14)
            $R'_\theta[y] \leftarrow r'_\theta(y \mid x)$.
        end for
        Numerator term $N_x \leftarrow \sum_{y \in \mathcal{Y}_x^+} \exp(R'_\theta[y])$.
        Denominator term $D_x \leftarrow \sum_{y' \in \mathcal{Y}_{\text{all}}} \exp(R'_\theta[y'])$.
        Sample loss $L_x \leftarrow -\log(N_x / D_x)$, ensuring $N_x / D_x \in (0, 1]$. (Ref. Eq. 15)
        Accumulate loss (e.g., $\mathcal{L}_{\text{epoch}} \leftarrow \mathcal{L}_{\text{epoch}} + L_x$).
    end for
    Compute gradient $\nabla_\theta \mathcal{L}_{\text{epoch}}$ (typically averaged over batch size).
    Update model parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{epoch}}$.
end for
return $\theta$.
```

D.2 Bias Definition

The bias with respect to attribute a is defined as:

$$B^{(k)} = \left| \mu_\theta^{(k)} - \mu_{\mathcal{A}} \right|, \tag{16}$$

where:

- $\mu_\theta^{(k)}$ is the expected attribute value under the model after training with k positive and k negative samples per query.
- $\mu_{\mathcal{A}} = \mathbb{E}_{x \sim \mathcal{X}}\left[\mu_{\mathcal{A}_x}\right]$, with $\mu_{\mathcal{A}_x} = \mathbb{E}_{y \sim \mathcal{A}_x}\left[a(y)\right]$.

D.3 Main Bias Result

Theorem 2. Under the stated assumptions, the expected bias $\mathbb{E}[B^{(k)}]$ decreases with the number of samples k as:

$$\mathbb{E}[B^{(k)}] \leq \frac{C}{\sqrt{k}}, \tag{17}$$

where $C = \sigma_{\max}$ is a constant depending on the maximum variance of $a(y)$ over the acceptable responses.

Proof. For each query x, consider the sample mean of the attribute over the k positive responses:

$$\overline{a}_x^{(k)} = \frac{1}{k}\sum_{i=1}^{k} a(y_{x,i}^{+}), \quad y_{x,i}^{+} \sim \mathcal{A}_x. \tag{18}$$

Since the $y_{x,i}^{+}$ are independent and identically distributed samples from $\mathcal{A}_x$, the expected value and variance of $\overline{a}_x^{(k)}$ are:

$$\mathbb{E}\left[\overline{a}_x^{(k)}\right] = \mu_{\mathcal{A}_x}, \tag{19}$$

$$\mathrm{Var}\left(\overline{a}_x^{(k)}\right) = \frac{\sigma_{\mathcal{A}_x}^2}{k} \leq \frac{\sigma_{\max}^2}{k}. \tag{20}$$

Using the fact that for any random variable Z with finite variance, the expected absolute deviation from its mean satisfies:

$$\mathbb{E}\left[|Z - \mathbb{E}[Z]|\right] \leq \sqrt{\mathrm{Var}[Z]}, \tag{21}$$

we have:

$$\mathbb{E}\left[\left|\overline{a}_x^{(k)} - \mu_{\mathcal{A}_x}\right|\right] \leq \sqrt{\frac{\sigma_{\mathcal{A}_x}^2}{k}} \leq \frac{\sigma_{\max}}{\sqrt{k}}. \tag{22}$$

Averaging over all queries $x \in \mathcal{X}$:

$$\mathbb{E}_x\left[\left|\overline{a}_x^{(k)} - \mu_{\mathcal{A}_x}\right|\right] \leq \frac{\sigma_{\max}}{\sqrt{k}}. \tag{23}$$

Since $\mu_\theta^{(k)} = \mathbb{E}_x\left[\overline{a}_x^{(k)}\right]$ and $\mu_{\mathcal{A}} = \mathbb{E}_x\left[\mu_{\mathcal{A}_x}\right]$, the expected bias is:

$$\mathbb{E}[B^{(k)}] = \left|\mu_\theta^{(k)} - \mu_{\mathcal{A}}\right| = \left|\mathbb{E}_x\left[\overline{a}_x^{(k)} - \mu_{\mathcal{A}_x}\right]\right| \tag{24}$$

$$\leq \mathbb{E}_x\left[\left|\overline{a}_x^{(k)} - \mu_{\mathcal{A}_x}\right|\right] \leq \frac{\sigma_{\max}}{\sqrt{k}}. \tag{25}$$

Thus, the expected bias decreases with k as $\frac{1}{\sqrt{k}}$.

$\square$

Corollary 1. As $k \to \infty$, the expected bias $\mathbb{E}[B^{(k)}]$ approaches zero:

$$\lim_{k\to\infty} \mathbb{E}[B^{(k)}] = 0. \tag{26}$$

Implications This theorem establishes a quantitative relationship between the bias $B^{(k)}$ and the number of samples k. It shows that incorporating multiple positive and negative responses per query reduces the bias with respect to attribute $a(y)$ at a rate proportional to $1/\sqrt{k}$. As k increases, the model's expected attribute value converges to the true expected attribute value over acceptable responses, leading to better alignment with human preferences.

E Differentiating the Group Contrastive Loss from InfoNCE Loss

In this subsection, we compare our proposed weighted contrastive loss function with the InfoNCA loss function. We present both loss functions, derive their gradients rigorously, and characterize their stationary points. Based on this characterization, we discuss the properties of the convergence points in terms of what the models learn and their alignment with human preferences.

E.1 Definitions of Loss Functions

InfoNCA Loss Function The InfoNCA loss function is defined as:

$$L_{\text{InfoNCA}} = -\sum_{i=1}^{K} p_i^{\text{target}} \log p_i^{\text{model}},$$

where p_i^{target} represents the target probability for the i-th response, calculated as

$$p_i^{\text{target}} = \frac{e^{r(x,y_i)/\alpha}}{\sum_{j=1}^{K} e^{r(x,y_j)/\alpha}},$$

and p_i^{model} denotes the model's predicted probability for the i-th response, given by

$$p_i^{\text{model}} = \frac{e^{s_\theta(y_i|x)}}{\sum_{j=1}^{K} e^{s_\theta(y_j|x)}}.$$

In this context, x is the instruction or prompt provided to the model, and $\{y_i\}_{i=1}^{K}$ represents a set of K responses generated for the instruction x. The term $r(x, y_i)$ is the reward associated with the response y_i, while $s_\theta(y_i \mid x) = \log\left(P_\theta(y_i \mid x)/P_{\text{ref}}(y_i \mid x)\right)$ is the score for response y_i. The parameter α serves as a temperature parameter that controls the influence of the reward, and K is the total number of responses considered for the instruction x.

Weighted Contrastive Loss Function Our proposed weighted contrastive loss function is expressed as:

$$L_{\text{weighted}} = -\log\left(\frac{\sum\limits_{i \in Y^+} w_i e^{s_\theta(y_i|x)}}{\sum\limits_{j=1}^{K} w_j e^{s_\theta(y_j|x)}}\right),$$

where Y^+ is the set of positive responses with rewards above the mean, defined as $Y^+ = \{y_i \mid S_i > S_{\text{mean}}\}$. Each response y_i is assigned a weight $w_i = e^{\alpha\delta_i}$, where δ_i is the deviation of the reward score S_i from the mean reward score S_{mean}. Specifically, $\delta_i = S_i - S_{\text{mean}}$ for responses in Y^+ and $\delta_i = S_{\text{mean}} - S_i$ for responses not in Y^+. The mean reward score S_{mean} is calculated as

$$S_{\text{mean}} = \frac{1}{K}\sum_{j=1}^{K} S_j,$$

where K is the total number of responses for the query x. The term $s_\theta(y_i \mid x)$ denotes the model's logit for response y_i, and α is a scaling hyperparameter that controls the influence of the deviation δ_i.

E.2 Gradient Analysis

To understand how each loss function influences the model during training, we derive the gradients with respect to the model logits $s_\theta(y_i \mid x)$ for both methods.

Gradient of InfoNCA Loss

Lemma 2. The gradient of the InfoNCA loss with respect to the model logits $s_\theta(y_i \mid x)$ is:

$$\frac{\partial L_{\text{InfoNCA}}}{\partial s_\theta(y_i \mid x)} = p_i^{\text{model}} - p_i^{\text{target}}. \tag{27}$$

Proof. The InfoNCA loss is:

$$L_{\text{InfoNCA}} = -\sum_{k=1}^{K} p_k^{\text{target}} \log p_k^{\text{model}}. \tag{28}$$

Our goal is to compute $\frac{\partial L_{\text{InfoNCA}}}{\partial s_\theta(y_i|x)}$.

Since p_k^{target} does not depend on $s_\theta(y_i \mid x)$ (the rewards are constants with respect to the model parameters), the derivative only affects the terms involving p_k^{model}.

First, express $\log p_k^{\text{model}}$ explicitly:

$$\log p_k^{\text{model}} = s_\theta(y_k \mid x) - \log\left(\sum_{j=1}^{K} e^{s_\theta(y_j|x)}\right). \tag{29}$$

Now, compute the derivative of $\log p_k^{\text{model}}$ with respect to $s_\theta(y_i \mid x)$:

$$\frac{\partial \log p_k^{\text{model}}}{\partial s_\theta(y_i \mid x)} = \frac{\partial s_\theta(y_k \mid x)}{\partial s_\theta(y_i \mid x)} - \frac{\partial}{\partial s_\theta(y_i \mid x)} \log\left(\sum_{j=1}^{K} e^{s_\theta(y_j|x)}\right). \tag{30}$$

Compute each term separately.

First term:

$$\frac{\partial s_\theta(y_k \mid x)}{\partial s_\theta(y_i \mid x)} = \delta_{ik}, \tag{31}$$

where δ_{ik} is the Kronecker delta, equal to 1 if $i = k$ and 0 otherwise.

Second term:

Let $Z = \sum_{j=1}^{K} e^{s_\theta(y_j|x)}$. Then,

$$\frac{\partial}{\partial s_\theta(y_i \mid x)} \log Z = \frac{1}{Z} \frac{\partial Z}{\partial s_\theta(y_i \mid x)}. \tag{32}$$

Compute $\frac{\partial Z}{\partial s_\theta(y_i|x)}$:

$$\frac{\partial Z}{\partial s_\theta(y_i \mid x)} = e^{s_\theta(y_i|x)}. \tag{33}$$

Therefore,

$$\frac{\partial}{\partial s_\theta(y_i \mid x)} \log Z = \frac{e^{s_\theta(y_i|x)}}{Z} = p_i^{\text{model}}. \tag{34}$$

Putting it all together:

$$\frac{\partial \log p_k^{\text{model}}}{\partial s_\theta(y_i \mid x)} = \delta_{ik} - p_i^{\text{model}}. \tag{35}$$

Now, compute the gradient of the loss:

$$\frac{\partial L_{\text{InfoNCA}}}{\partial s_\theta(y_i \mid x)} = -\sum_{k=1}^{K} p_k^{\text{target}} \frac{\partial \log p_k^{\text{model}}}{\partial s_\theta(y_i \mid x)} \tag{36}$$

$$= -\sum_{k=1}^{K} p_k^{\text{target}} \left(\delta_{ik} - p_i^{\text{model}}\right) \tag{37}$$

$$= -\left(p_i^{\text{target}} - p_i^{\text{model}} \sum_{k=1}^{K} p_k^{\text{target}}\right). \tag{38}$$

Since $\sum_{k=1}^{K} p_k^{\text{target}} = 1$, we have:

$$\sum_{k=1}^{K} p_k^{\text{target}} = 1 \implies \sum_{k=1}^{K} p_k^{\text{target}} = 1. \tag{39}$$

Therefore,

$$\frac{\partial L_{\text{InfoNCA}}}{\partial s_\theta(y_i \mid x)} = -\left(p_i^{\text{target}} - p_i^{\text{model}} \cdot 1\right) = p_i^{\text{model}} - p_i^{\text{target}}. \tag{40}$$

□

Gradient of Weighted Contrastive Loss

Lemma 3. The gradient of the weighted contrastive loss with respect to the model logits $s_\theta(y_i \mid x)$ is:

$$\frac{\partial L_{\text{weighted}}}{\partial s_\theta(y_i \mid x)} = p_i^{\text{weighted}} - p_i^{\text{pos}} \tag{41}$$

where:

$$p_i^{\text{weighted}} = \frac{w_i e^{s_\theta(y_i|x)}}{\sum_{j=1}^{K} w_j e^{s_\theta(y_j|x)}}, \qquad p_i^{\text{pos}} = \frac{w_i e^{s_\theta(y_i|x)}}{\sum_{k\in Y^+} w_k e^{s_\theta(y_k|x)}} \cdot \mathbb{I}_{y_i \in Y^+}, \tag{42}$$

and $\mathbb{I}_{y_i \in Y^+}$ is the indicator function, equal to 1 if $y_i \in Y^+$ and 0 otherwise.

Proof. Let us denote:

$$A = \sum_{k\in Y^+} w_k e^{s_\theta(y_k|x)}, \quad Z = \sum_{j=1}^{K} w_j e^{s_\theta(y_j|x)}. \tag{43}$$

The weighted contrastive loss is:

$$L_{\text{weighted}} = -\log\left(\frac{A}{Z}\right) = -\log A + \log Z. \tag{44}$$

Compute the derivative with respect to $s_\theta(y_i \mid x)$:

$$\frac{\partial L_{\text{weighted}}}{\partial s_\theta(y_i \mid x)} = -\frac{1}{A}\frac{\partial A}{\partial s_\theta(y_i \mid x)} + \frac{1}{Z}\frac{\partial Z}{\partial s_\theta(y_i \mid x)}. \tag{45}$$

Compute $\frac{\partial A}{\partial s_\theta(y_i|x)}$:

$$\frac{\partial A}{\partial s_\theta(y_i \mid x)} = w_i e^{s_\theta(y_i|x)} \cdot \mathbb{I}_{y_i \in Y^+}. \tag{46}$$

Compute $\frac{\partial Z}{\partial s_\theta(y_i|x)}$:

$$\frac{\partial Z}{\partial s_\theta(y_i \mid x)} = w_i e^{s_\theta(y_i|x)}. \tag{47}$$

Substitute back into the gradient:

$$\begin{align}
\frac{\partial L_{\text{weighted}}}{\partial s_\theta(y_i \mid x)} &= -\frac{1}{A} w_i e^{s_\theta(y_i|x)} \cdot \mathbb{I}_{y_i \in Y^+} + \frac{1}{Z} w_i e^{s_\theta(y_i|x)} \tag{48}\\
&= w_i e^{s_\theta(y_i|x)} \left(\frac{1}{Z} - \frac{\mathbb{I}_{y_i \in Y^+}}{A} \right). \tag{49}
\end{align}$$

Recognize that:

$$p_i^{\text{weighted}} = \frac{w_i e^{s_\theta(y_i|x)}}{Z}, \quad p_i^{\text{pos}} = \frac{w_i e^{s_\theta(y_i|x)}}{A} \cdot \mathbb{I}_{y_i \in Y^+}. \tag{50}$$

Therefore:

$$\frac{\partial L_{\text{weighted}}}{\partial s_\theta(y_i \mid x)} = p_i^{\text{weighted}} - p_i^{\text{pos}}. \tag{51}$$

Since $p_i^{\text{pos}} = 0$ when $y_i \notin Y^+$, we have:

$$\frac{\partial L_{\text{weighted}}}{\partial s_\theta(y_i \mid x)} = \begin{cases} p_i^{\text{weighted}} - p_i^{\text{pos}}, & \text{if } y_i \in Y^+, \\ p_i^{\text{weighted}} - 0 = p_i^{\text{weighted}}, & \text{if } y_i \in Y^-. \end{cases} \tag{52}$$

However, this suggests that the gradient is always positive for negative examples. In other words, given w_i and Z are positive, $e^{s_\theta(y_i|x)}$ keeps increasing. But note that $s_\theta(y_i \mid x) = -\log(P_\theta(y_i \mid x))$. Hence $\frac{1}{P_\theta(y_i|x)}$ keeps increasing implying that $P_\theta(y_i \mid x)$ keeps decreasing. i.e. at the stationary point, $P_\theta(y_i \mid x) \to 0$ for all negative examples, $y_i \in Y^-$.

Now let us examine the positive examples. The gradient simplifies to $w_i e^{s_\theta(y_i|x)} \left(\frac{1}{Z} - \frac{1}{A}\right)$. Since $Z \geq A$, $\frac{1}{Z} \leq \frac{1}{A}$. Hence the gradient term with respect to $s_\theta(y_i \mid x)$ is negative. Notice that $e^{s_\theta(y_i|x)} = \frac{1}{P_\theta(y_i|x)}$. A negative gradient implies that $\frac{1}{P_\theta(y_i|x)}$ decreases, implying that $P_\theta(y_i \mid x)$ increases for all positive examples $y_i \in Y^+$.

□

We now provide the gradients directly in terms of $P_\theta(y_j \mid x)$ instead of the scores $s_\theta(y_j \mid x)$, for easy interpretibility in terms of the probabilities.

Lemma 4. Let the weighted contrastive loss be defined as:

$$L_{\text{weighted}} = -\log\left(\frac{V}{U}\right) = -\log V + \log U,$$

where

$$U = \sum_{j=1}^{K} u_j P_\theta(y_j \mid x), \quad V = \sum_{i \in Y^+} u_i P_\theta(y_i \mid x),$$

and

$$u_i = \frac{w_i}{P_{\text{ref}}(y_i \mid x)},$$

with $w_i = e^{\alpha\delta_i}$, $P_\theta(y_i \mid x)$ being the model probability for response y_i, and $P_{\text{ref}}(y_i \mid x)$ being the reference model probability.

Then, the gradient of the weighted contrastive loss with respect to $P_\theta(y_i \mid x)$ is given by:

- For positive examples ($y_i \in Y^+$):

$$\frac{\partial L_{\text{weighted}}}{\partial P_\theta(y_i \mid x)} = u_i \left(\frac{1}{U} - \frac{1}{V} \right), \tag{53}$$

- For negative examples ($y_i \notin Y^+$):

$$\frac{\partial L_{\text{weighted}}}{\partial P_\theta(y_i \mid x)} = \frac{u_i}{U}. \tag{54}$$

Proof. Using the score function $s_\theta(y_i \mid x) = \log\left(\frac{P_\theta(y_i|x)}{P_{\text{ref}}(y_i|x)}\right)$, we have $e^{s_\theta(y_i|x)} = \dfrac{P_\theta(y_i \mid x)}{P_{\text{ref}}(y_i \mid x)}$.

The weighted contrastive loss becomes:

$$L_{\text{weighted}} = -\log\left(\frac{\displaystyle\sum_{i \in Y^+} w_i e^{s_\theta(y_i|x)}}{\displaystyle\sum_{j=1}^{K} w_j e^{s_\theta(y_j|x)}} \right) = -\log\left(\frac{\displaystyle\sum_{i \in Y^+} w_i \frac{P_\theta(y_i \mid x)}{P_{\text{ref}}(y_i \mid x)}}{\displaystyle\sum_{j=1}^{K} w_j \frac{P_\theta(y_j \mid x)}{P_{\text{ref}}(y_j \mid x)}} \right) = -\log\left(\frac{V}{U}\right),$$

where $u_i = \dfrac{w_i}{P_{\text{ref}}(y_i \mid x)}$, $V = \sum_{i \in Y^+} u_i P_\theta(y_i \mid x)$, and $U = \sum_{j=1}^{K} u_j P_\theta(y_j \mid x)$.

We compute the gradient of L_{weighted} with respect to $P_\theta(y_i \mid x)$:

$$\frac{\partial L_{\text{weighted}}}{\partial P_\theta(y_i \mid x)} = -\frac{1}{V} \cdot \frac{\partial V}{\partial P_\theta(y_i \mid x)} + \frac{1}{U} \cdot \frac{\partial U}{\partial P_\theta(y_i \mid x)}.$$

Case 1: For $y_i \in Y^+$:

$$\frac{\partial V}{\partial P_\theta(y_i \mid x)} = u_i, \quad \frac{\partial U}{\partial P_\theta(y_i \mid x)} = u_i.$$

Thus,

$$\frac{\partial L_{\text{weighted}}}{\partial P_\theta(y_i \mid x)} = -\frac{u_i}{V} + \frac{u_i}{U} = u_i \left(\frac{1}{U} - \frac{1}{V} \right).$$

Case 2: For $y_i \notin Y^+$:

$$\frac{\partial V}{\partial P_\theta(y_i \mid x)} = 0, \quad \frac{\partial U}{\partial P_\theta(y_i \mid x)} = u_i.$$

Thus,

$$\frac{\partial L_{\text{weighted}}}{\partial P_\theta(y_i \mid x)} = 0 + \frac{u_i}{U} = \frac{u_i}{U}.$$

□

Corollary 2. The sign of the gradient indicates the optimization direction:

- For positive examples ($y_i \in Y^+$), since $V \leq U$, we have $\dfrac{1}{U} - \dfrac{1}{V} \leq 0$. Therefore, the gradient $\dfrac{\partial L_{\text{weighted}}}{\partial P_\theta(y_i \mid x)} \leq 0$, and minimizing L_{weighted} involves **increasing** $P_\theta(y_i \mid x)$.
- For negative examples ($y_i \notin Y^+$), the gradient $\dfrac{\partial L_{\text{weighted}}}{\partial P_\theta(y_i \mid x)} > 0$, and minimizing L_{weighted} involves **decreasing** $P_\theta(y_i \mid x)$.

Proof. As established in the lemma:

For positive examples ($y_i \in Y^+$): Since $V = \sum_{i \in Y^+} u_i P_\theta(y_i \mid x)$ and $U = V + \sum_{j \notin Y^+} u_j P_\theta(y_j \mid x)$, it follows that $V \leq U$ and thus $\frac{1}{U} - \frac{1}{V} \leq 0$.

Therefore, the gradient:

$$\frac{\partial L_{\text{weighted}}}{\partial P_\theta(y_i \mid x)} = u_i \left(\frac{1}{U} - \frac{1}{V} \right) \leq 0.$$

A negative gradient indicates that increasing $P_\theta(y_i \mid x)$ will decrease L_{weighted}. Hence, to minimize the loss, we should **increase** $P_\theta(y_i \mid x)$ for positive examples.

For negative examples ($y_i \notin Y^+$): The gradient is:

$$\frac{\partial L_{\text{weighted}}}{\partial P_\theta(y_i \mid x)} = \frac{u_i}{U} > 0,$$

since $u_i > 0$ and $U > 0$. A positive gradient indicates that decreasing $P_\theta(y_i \mid x)$ will decrease L_{weighted}. Therefore, to minimize the loss, we should **decrease** $P_\theta(y_i \mid x)$ for negative examples. ☐

F Characterization of Stationary Points

We now characterize the stationary points of both loss functions.

F.1 Stationary Points of the InfoNCA Loss Function

Theorem 3. For the InfoNCA loss, the stationary points occur when:

$$p_i^{\text{model}} = p_i^{\text{target}}, \quad \forall i \in \{1, \ldots, K\}. \tag{55}$$

Proof. Stationary points are defined by the condition:

$$\frac{\partial L_{\text{InfoNCA}}}{\partial s_\theta(y_i \mid x)} = 0, \quad \forall i. \tag{56}$$

From the gradient:

$$\frac{\partial L_{\text{InfoNCA}}}{\partial s_\theta(y_i \mid x)} = p_i^{\text{model}} - p_i^{\text{target}}, \tag{57}$$

setting the gradient to zero yields:

$$p_i^{\text{model}} = p_i^{\text{target}}, \quad \forall i. \tag{58}$$

☐

Remark 1. This stationary point is suboptimal because p_i^{model} expands to:

$$p_i^{\text{model}} = \frac{e^{\log P_\theta(y_i|x) - \log P_{\text{ref}}(y_i|x)}}{\sum_{j=1}^{K} e^{\log P_\theta(y_j|x) - \log P_{\text{ref}}(y_j|x)}}$$

Rather than equating the soft-max of the difference between $\log P_\theta(y_i|x)$ and $\log P_{\text{ref}}(y_i|x)$ to p_i^{target}, optimality may require directly setting $\log P_\theta(y|x)$ to match the softmax of the target scores.

F.2 Stationary Points of the Weighted Contrastive Loss under Simplifying Assumptions

Lemma 5. Consider the weighted contrastive loss function in a simplified scenario with the following conditions: There are N^+ positive examples, each with weight w^+, and N^- negative examples, each with weight w^-. All positive examples have the same score $s^{(t)}$ at iteration t, and all negative examples have the same score $s^{(t)}$ at iteration t. Then, the update rule for the score $s^{(t)}$ of the positive examples at iteration $t+1$ is given by

$$s^{(t+1)} = s^{(t)} + \eta \left(\frac{N^- w^-}{N^+(N^+ w^+ + N^- w^-)} \right), \tag{59}$$

where η is the learning rate.

Proof. Let Y^+ denote the set of positive examples and Y^- the set of negative examples, with N^+ and N^- examples respectively for a total of $K = N^+ + N^-$ examples. With weights w^+ and w^- assigned to positive and negative examples respectively, and logits $s^{(t)}$ for both classes at timestep t, the weighted contrastive loss function is defined as:

$$L_{\text{weighted}}(\theta) = -\log \left(\frac{\sum\limits_{i \in Y^+} w_i e^{s_i}}{\sum\limits_{j=1}^{K} w_j e^{s_j}} \right), \tag{60}$$

where $w_i = w^+$ and $s_i = s^{(t)}$ for $i \in Y^+$, and $w_j = w^-$ and $s_j = s^{(t)}$ for $j \in Y^-$.

Compute the numerator A and the denominator Z of the loss function:

$$A = \sum_{i \in Y^+} w_i e^{s_i} = N^+ w^+ e^{s^{(t)}}, \tag{61}$$

$$Z = \sum_{j=1}^{K} w_j e^{s_j} = N^+ w^+ e^{s^{(t)}} + N^- w^- e^{s^{(t)}} = e^{s^{(t)}} (N^+ w^+ + N^- w^-). \tag{62}$$

For positive examples $i \in Y^+$, the weighted probability p_i^{weighted} and the positive probability p_i^{pos} are:

$$p_i^{\text{weighted}} = \frac{w^+ e^{s^{(t)}}}{Z} = \frac{w^+}{N^+ w^+ + N^- w^-}, \tag{63}$$

$$p_i^{\text{pos}} = \frac{w^+ e^{s^{(t)}}}{A} = \frac{w^+}{N^+ w^+} = \frac{1}{N^+}. \tag{64}$$

The gradient of the loss with respect to $s^{(t)}$ for positive examples is:

$$\frac{\partial L_{\text{weighted}}}{\partial s^{(t)}} = p_i^{\text{weighted}} - p_i^{\text{pos}} = \frac{w^+}{N^+ w^+ + N^- w^-} - \frac{1}{N^+}. \tag{65}$$

To simplify this expression, we find a common denominator $D = N^+(N^+ w^+ + N^- w^-)$:

$$\frac{\partial L_{\text{weighted}}}{\partial s^{(t)}} = \frac{w^+ N^+ - (N^+ w^+ + N^- w^-)}{D} \tag{66}$$

$$= \frac{w^+ N^+ - N^+ w^+ - N^- w^-}{N^+(N^+ w^+ + N^- w^-)} \tag{67}$$

$$= \frac{-N^- w^-}{N^+(N^+ w^+ + N^- w^-)}. \tag{68}$$

The update rule for $s^{(t)}$ is then:

$$s^{(t+1)} = s^{(t)} - \eta \frac{\partial L_{\text{weighted}}}{\partial s^{(t)}} = s^{(t)} + \eta \left(\frac{N^- w^-}{N^+(N^+ w^+ + N^- w^-)} \right). \tag{69}$$

This completes the proof. □

Corollary 3. Assuming the initial scores are zero ($s^{(0)} = 0$), the score $s^{(t)}$ of the positive examples at iteration t is given by

$$s^{(t)} = t\eta \left(\frac{N^- w^-}{N^+(N^+ w^+ + N^- w^-)} \right). \tag{70}$$

Proof. From the update rule established in the lemma,

$$s^{(t+1)} = s^{(t)} + c, \tag{71}$$

where

$$c = \eta \left(\frac{N^- w^-}{N^+(N^+ w^+ + N^- w^-)} \right). \tag{72}$$

Since $s^{(0)} = 0$, we have

$$\begin{align} s^{(1)} &= s^{(0)} + c = c, \tag{73} \\ s^{(2)} &= s^{(1)} + c = 2c, \tag{74} \\ &\vdots \tag{75} \\ s^{(t)} &= tc. \tag{76} \end{align}$$

Substituting c back into the expression, we obtain

$$s^{(t)} = t\eta \left(\frac{N^- w^-}{N^+(N^+ w^+ + N^- w^-)} \right). \tag{77}$$

□

Corollary 4. In the special case where there is one positive example ($N^+ = 1$) and one negative example ($N^- = 1$), and the weights are $w^+ = w^- = 1$ (as in Direct Preference Optimization), the score $s^{(t)}$ at iteration t is:

$$s^{(t)} = \frac{\eta t}{2}. \tag{78}$$

Proof. Substituting $N^+ = N^- = 1$ and $w^+ = w^- = 1$ into the expression for $s^{(t)}$:

$$\begin{align} s^{(t)} &= t\eta \left(\frac{1 \times 1}{1 \times (1 \times 1 + 1 \times 1)} \right) \tag{79} \\ &= t\eta \left(\frac{1}{1 \times (1 + 1)} \right) \tag{80} \\ &= t\eta \left(\frac{1}{2} \right) \tag{81} \\ &= \frac{\eta t}{2}. \tag{82} \end{align}$$

□

Lemma 6. Consider the general case where positive examples may have different weights w_i^+, and each positive example i has its own score $s_i^{(t)}$ at iteration t. Assuming initial scores $s_i^{(0)} = 0$ for all positive examples, the score $s_i^{(t)}$ of positive example i at iteration t, up to a linear approximation, is given by

$$s_i^{(t)} = t\eta w_i^+ \left(\frac{B_0}{A_0 Z_0} \right), \tag{83}$$

where $A_0 = \sum_{k \in Y^+} w_k^+$, $B_0 = \sum_{j \in Y^-} w_j^-$, $Z_0 = A_0 + B_0$, and η is the learning rate.

Proof. At iteration $t = 0$, the initial scores are $s_i^{(0)} = 0$ for all $i \in Y^+$. The sums are:

$$A_0 = \sum_{k \in Y^+} w_k^+ e^{s_k^{(0)}} = \sum_{k \in Y^+} w_k^+ = W^+, \tag{84}$$

$$B_0 = \sum_{j \in Y^-} w_j^- e^{s_j^{(0)}} = \sum_{j \in Y^-} w_j^- = W^-. \tag{85}$$

The total sum is $Z_0 = A_0 + B_0 = W^+ + W^-$.

The gradient for each positive example i at $t = 0$ is:

$$\frac{\partial L_{\text{weighted}}}{\partial s_i^{(0)}} = -w_i^+ e^{s_i^{(0)}} \left(\frac{B_0}{A_0 Z_0} \right) = -w_i^+ \left(\frac{B_0}{A_0 Z_0} \right). \tag{86}$$

The update rule is:

$$s_i^{(1)} = s_i^{(0)} - \eta \frac{\partial L_{\text{weighted}}}{\partial s_i^{(0)}} = \eta w_i^+ \left(\frac{B_0}{A_0 Z_0} \right). \tag{87}$$

Assuming that the term $\frac{B_0}{A_0 Z_0}$ remains approximately constant over iterations (which holds when η is small and changes in $s_i^{(t)}$ are small), the score at iteration t is:

$$s_i^{(t)} = t \eta w_i^+ \left(\frac{B_0}{A_0 Z_0} \right). \tag{88}$$

□

Remark 2. The approximation assumes that A_t, B_t, and Z_t remain close to their initial values A_0, B_0, and Z_0 over the iterations considered, and the score values remain small. This is reasonable for small learning rates η and a limited number of iterations t.

F.3 Stationary Points of the Weighted Contrastive Loss

We now analyze the stationary points of our weighted contrastive loss function.

Lemma 7. For the weighted contrastive loss function, the stationary point occurs when the probabilities of the negative samples approach zero, i.e.,

$$P_\theta(y_i \mid x) \to 0 \quad \text{for all } y_i \in Y^-. \tag{89}$$

Proof. From Lemma 3, the gradient of the weighted contrastive loss with respect to the model logits $s_\theta(y_i \mid x)$ is:

$$\frac{\partial L_{\text{weighted}}}{\partial s_\theta(y_i \mid x)} = \begin{cases} p_i^{\text{weighted}} - p_i^{\text{pos}}, & \text{if } y_i \in Y^+, \\ p_i^{\text{weighted}}, & \text{if } y_i \in Y^-. \end{cases} \tag{90}$$

At a stationary point, the gradient must be zero for all y_i. Consider the negative samples $y_i \in Y^-$. Setting the gradient to zero yields:

$$\frac{\partial L_{\text{weighted}}}{\partial s_\theta(y_i \mid x)} = p_i^{\text{weighted}} = 0. \tag{91}$$

Since p_i^{weighted} is the normalized weighted probability of y_i, given by:

$$p_i^{\text{weighted}} = \frac{w_i e^{s_\theta(y_i \mid x)}}{\sum_{j=1}^{K} w_j e^{s_\theta(y_j \mid x)}}, \tag{92}$$

and $w_i > 0$, the only way for p_i^{weighted} to be zero is if $e^{s_\theta(y_i \mid x)} = 0$, which implies:

$$s_\theta(y_i \mid x) \to -\infty \quad \implies \quad P_\theta(y_i \mid x) \to 0 \quad \text{for } y_i \in Y^-. \tag{93}$$

Similarly, for positive samples $y_i \in Y^+$, the gradient is:

$$\frac{\partial L_{\text{weighted}}}{\partial s_\theta(y_i \mid x)} = p_i^{\text{weighted}} - p_i^{\text{pos}} = 0. \tag{94}$$

This implies:

$$p_i^{\text{weighted}} = p_i^{\text{pos}}. \tag{95}$$

Since the probabilities of the negative samples approach zero, the denominator in p_i^{weighted} becomes:

$$\sum_{j=1}^{K} w_j e^{s_\theta(y_j|x)} \approx \sum_{k \in Y^+} w_k e^{s_\theta(y_k|x)}. \tag{96}$$

Therefore, $p_i^{\text{weighted}} \approx p_i^{\text{pos}}$, satisfying the condition for the gradient to be zero for positive samples.

Thus, at the stationary point, the probabilities of the negative samples approach zero. □

Remark 3. When the probabilities of the negative samples approach zero, the scores $s_\theta(y_i \mid x)$ for $y_i \in Y^-$ tend to $-\infty$. Since:

$$e^{s_\theta(y_i|x)} = \frac{P_\theta(y_i \mid x)}{P_{\text{ref}}(y_i \mid x)} \to 0, \tag{97}$$

the weighted contributions of the negative samples to the numerator and denominator of L_{weighted} become negligible.

Consequently, the numerator and denominator of L_{weighted} become equal:

$$\sum_{i \in Y^+} w_i e^{s_\theta(y_i|x)} \approx \sum_{j=1}^{K} w_j e^{s_\theta(y_j|x)}. \tag{98}$$

Therefore:

$$L_{\text{weighted}} = -\log \left(\frac{\sum_{i \in Y^+} w_i e^{s_\theta(y_i|x)}}{\sum_{j=1}^{K} w_j e^{s_\theta(y_j|x)}} \right) \approx -\log 1 = 0. \tag{99}$$

This implies that the loss vanishes when the probabilities of the negative samples approach zero, indicating that the model has successfully minimized the loss by focusing entirely on the positive responses.

G Additional Experiments

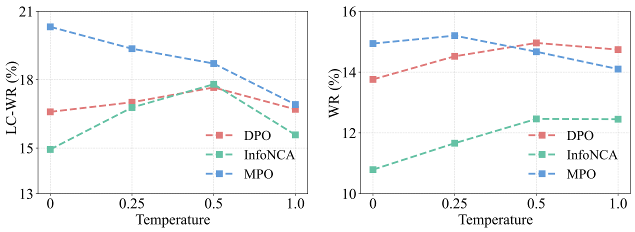

Figure 9: Performance variation of MPO for Mistral-Base with different sampling temperature on AlpacaEval2.

Optimizing for Lower Temperatures Fig. 9 evaluates MPO, DPO, and InfoNCA on AlpacaEval 2 (Mistral-Base 7B) against sampling temperature in the off-policy setting for Mistral-7B. MPO achieves its peak LC-WR of $\sim 20.5\%$ at zero temperature (greedy decoding), declining thereafter. Notably, MPO consistently leads in LC-Win Rate across all temperatures.

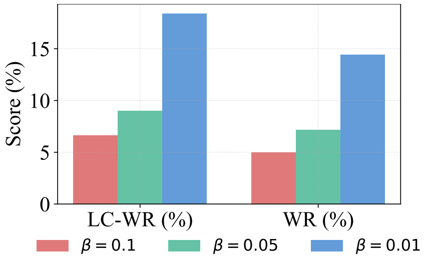

Figure 10: Performance variation of MPO for Mistral-Base with different β values on AlpacaEval2.

Effect of β on Optimization Dynamics. In Figure 10, we conduct an ablation study to investigate the role of the temperature parameter β in our optimization objective. As β increases, the gradient magnitudes decrease due to the flattening of the sigmoid-like objective, effectively pushing the scaled reward term $\beta \cdot \log \frac{\pi_{policy}(y|x)}{\pi_{ref}(y|x)}$ outside the region where the derivative is steepest. This leads to slower updates and consequently a more conservative learning trajectory. In contrast, smaller values of β keep the optimization in the high-sensitivity regime of the sigmoid, allowing for faster convergence. Empirically, we observe that using a smaller β (e.g., $\beta = 0.01$) enables the model to reach comparable performance to higher β settings with significantly fewer epochs and smaller learning rates. This demonstrates that careful tuning of β can not only preserve final accuracy but also substantially reduce training cost.

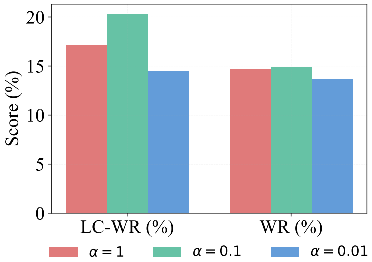

Figure 11: Performance variation of W-MPO for Mistral-Base with different α values on AlpacaEval2.

Effect of α on Optimization Dynamics. In Figure 11, we analyze the impact of the reward normalization coefficient α, which scales the raw reward signal before computing the deviation weight in W-MPO. This parameter plays a crucial role in balancing the sensitivity of the weighting function. A smaller α (e.g., $\alpha = 1$) retains the raw reward's expressivity, but can lead to high variance in gradient updates, potentially destabilizing training. On the other hand, a very large α (e.g., $\alpha = 0.01$) overly compresses the reward differences, weakening the contrast between preferred and dispreferred responses—this leads to underutilized supervision and poor alignment, as evidenced by the drop in LC-WR from 20.3 to 14.43. We find that an intermediate setting (e.g., $\alpha = 0.1$) strikes a favorable balance, providing stable optimization while preserving informative contrast. This setting achieves the highest LC-WR and WR performance, demonstrating the importance of calibrating α to match the scale and distribution of reward scores.

H Baselines Used for Comparison

When dealing with reward datasets where each instruction has more than two $n > 2$ responses, one common approach is to convert the data into pairwise preferences and then apply preference optimization techniques such as Direct Preference Optimization (DPO). Several strategies can be adopted for this purpose, each offering distinct trade-offs in terms of dataset richness and computational overhead.One straightforward method, as implemented by Zephyr Tunstall et al. (2023), involves selecting the response with the highest reward and pairing it with a randomly chosen response from the remaining responses for each instruction. Another variant involves pairing the highest-rewarded response with the lowest-rewarded response for each instruction, ensuring a clear distinction between

Method	Llama3-8B-Instruct			Mistral-7B-Instruct		
	LC-WR (%)	WR (%)	Std. Err.	LC-WR (%)	WR (%)	Std. Err.
DPO	41.1	37.8	1.38	26.8	24.6	1.39
MPO	49.0	50.6	1.52	28.2	29.4	1.34
W-MPO	**52.1**	**52.5**	1.46	**29.5**	**30.6**	1.45

Table 5: Comparison of MPO variants with DPO on **Llama3-8B-Instruct** and **Mistral-7B-Instruct**. We report Length-Controlled Win Rate (LC-WR), Overall Win Rate (WR), and standard error across evaluations.

preferences.Additionally, alternative baselines can be explored to enhance performance by incorporating more suboptimal responses during training. By applying DPO to combinations of responses, we can significantly expand the preference dataset and potentially achieve improved optimization. Baselines in this context are:

DPOx$\binom{N}{2}$: In this approach, all possible pairwise combinations of $\binom{N}{2}$ are generated, and DPO is applied to the entire combinatorial dataset. This method ensures the model is exposed to a comprehensive range of preference relationships, including those involving suboptimal responses.

We implement and evaluated several preference optimization baselines under a consistent offline setting using the ULTRAFEEDBACK dataset. Below, we describe each method in detail:

Group Relative Policy Optimization (GRPO): We implement GRPO in an offline setup, where both the response set (four responses per prompt) and their scalar rewards are fixed, as provided by ULTRAFEEDBACK. GRPO uses a relative advantage formulation that weighs each response based on its normalized reward gap.

Plackett-Luce Ranking Model: We implement the Plackett-Luce model, in which the ranking over responses is determined by the reward scores provided in ULTRAFEEDBACK. The objective maximizes the likelihood of the observed ranking via a sequential softmax over score-ordered responses.

InfoNCA: We adapt InfoNCA to the preference optimization setting by treating each response as a sample in a softmax-based contrastive loss, where rewards from ULTRAFEEDBACK define the similarity scores. All responses are used jointly during training.

MPO-1vsk: In this baseline for each prompt, the response with the highest reward is selected as the positive sample, and the remaining responses form the negative set. This formulation enables multi-response contrastive alignment.

All baselines are trained under a unified offline pipeline to ensure fair and consistent comparison across evaluation benchmarks.

I Experimental Details

Model and Training Setting For our off-policy experiments, we utilized the Ultrafeedback Dataset Cui et al. (2023), an instruction-following benchmark annotated by GPT-4. This dataset consists of approximately 64,000 instructions, each paired with four responses generated by different language models. GPT-4 assigned scalar rewards on a 0-to-10 scale for each response, which prior research has shown to correlate strongly with human annotations. This establishes GPT-4 ratings as a reliable and cost-efficient alternative to manual feedback.

In our broader framework, we first trained a base model (mistralai/Mistral-7B-v0.1) and (meta-llama/Meta-Llama-3-8B) on the UltraChat-200k dataset to obtain an SFT model. This SFT model, trained on open-source data, provides a transparent starting point. Subsequently, we refined the model by performing preference optimization on the UltraFeedback dataset. Once fine-tuned, the model was used for alignment. This two-step process ensures the model is well-prepared for tasks.

In our experiments, we observed that tuning hyperparameters is critical for optimizing the performance of all offline preference optimization algorithms, including DPO, SimPO, MPO and W-MPO. Hyperparameter selection is very crucial for these.For MPO, we found that setting the β parameter in the range of 0.01 to 0.05 consistently yields strong performance. For W-MPO, we found that setting the β parameter in the range of 0.01 to 0.05 and α to 10 consistently yields strong performance. Learning rate for mistral-experiments was fixed to 3e-7 and whereas for llama it was 5e-7 for all the baselines.

For our on-policy and iterative experiments, we utilize a pretrained instruction-tuned model (meta-llama/MetaLlama-3-8B-Instruct) and (mistralai/Mistral-7B-Instruct-v0.2) , as the SFT model. These models have undergone extensive instruction tuning, making them more capable and robust compared to the SFT models used in the Base setup. However, their reinforcement learning with human feedback (RLHF) procedures remain undisclosed, making them less transparent.

To reduce distribution shift between the SFT models and the preference optimization process, we follow the approach in Meng et al. (2024) and generate the preference dataset using the same SFT models. This ensures that our setup is more aligned with an on-policy setting. Specifically, we utilize prompts from the UltraFeedback dataset Cui et al. (2023) and regenerate the resonses using the SFT models. For each prompt x, we produce 5 responses by sampling from the SFT model with a sampling temperature of 1.0 and top-p 1.0. We then use the reward model (Skywork/Skywork-Reward-Llama-3.1-8B-v0.2) Liu et al. (2024b) to score all the responses

Table 6: Hyperparameters β and learning rate α across different training settings.

Setting	**Model**	**Iter**	β	*lr*
Offline	Mistral-Base (7B)	–	0.01	4e-7
	Llama3-Base (8B)	–	0.01	5e-7
Online	Mistral-Instruct (7B)	–	0.01	1.5e-7
	LLama3-Instruct (8B)	–	0.01	3e-7
Iterative	Mistral-Instruct (7B)	1	0.01	2.5e-7
	Llama3-Instruct (8B)	1	0.01	5e-7
	Mistral-Instruct (7B)	2	0.01	2e-7
	Llama3-Instruct (8B)	2	0.01	4e-7
	Mistral-Instruct (7B)	3	0.01	1.5e-7
	Llama3-Instruct (8B)	3	0.01	3e-7

In our online and iterative training experiments, we found that systematic hyperparameter tuning is critical for optimizing performance across diverse datasets. Notably, the choice of the β parameter had a substantial impact, with values in the range of 0.01 to 0.1 consistently yielding strong results. For all experiments, we adopted a fixed learning rate of 1.5e-7 for mistral-based models and 3.0e-7 for llama-based models.

All experiments employed the AdamW optimizer with a cosine annealing learning rate scheduler, incorporating a 0.01% warmup step and a total of 1 training epoch. We used 8×A100 (80GB) GPUs under DeepSpeed with distributed data parallel (DDP) training. The global batch size was set to 128, with a per-device batch size of 2 and a gradient accumulation step of 8 to ensure memory efficiency and training stability.

These findings emphasize the importance of well-calibrated hyperparameters and infrastructure settings to achieve robust and reproducible outcomes in post-training alignment tasks.

Evaluation Benchmarks: We evaluate our models using three widely recognized open-ended instruction-following benchmarks: MT-Bench Zheng et al. (2023), AlpacaEval 2, and Arena-Hard v0.1 Zheng et al. (2023). These benchmarks test the models' conversational versatility across a broad range of queries and are broadly utilized in the research community. AlpacaEval2 Dubois et al. (2024) includes 805 questions derived from five datasets, while MT-Bench spans eight categories with a total of 80 questions. Arena-Hard, a recently updated version of MT-Bench, focuses on 500 well-defined technical problem-solving queries. Scores are reported based on the evaluation protocols of each benchmark. For AlpacaEval2, both the raw win rate (WR) and the length-controlled win rate (LC) are reported, with LC specifically designed to mitigate biases related to model verbosity. For Arena-Hard, the win rate is reported relative to a baseline model. For MT-Bench, the average score is calculated using evaluations by GPT-4 as judge. For decoding details, we generate responses using both greedy decoding and multinomial sampling with temperatures of 0.2, 0.5, and 1.0. To address potential biases introduced by multinomial sampling at varying temperatures, we generate responses three times for each setting at different seed and average their performance across the datasets.

J Reward Loss Computation

In this section we provide the actual code used to compute the reward losses.

```python
import torch

def mpo_loss(pi_logps, ref_logps, rewards, beta, alpha):
    """
    pi_logps: policy logprobs for N responses, shape (Batch_Size, N)
    ref_logps: reference logprobs for N responses, shape (Batch_Size, N)
    rewards: reward labels for N responses, shape (Batch_Size, N)
    beta: Temperature parameter for the MPO loss
    alpha: rating weight
    """

    logits = pi_logps - ref_logps  # Compute logits
    rewards = rewards / alpha # Normalizing the reward value to logits scale

    mean_rewards = torch.mean(rewards, dim=-1)
    if self.norm > 0:
        weights = torch.abs(rewards - mean_rewards.reshape(-1, 1))
    else:
        deviation_reward = 0

    pos_mask = (rewards > mean_rewards.reshape(-1, 1)) * 1
    neg_mask = torch.logical_not(pos_mask) * 1

    eps = 1e-10
    logits = (logits + weights) * beta
    logits = logits - logits.max(dim=-1, keepdim=True)[0]  # Stabilize logits
    softmax_val = torch.softmax(logits + eps, dim=-1)
    pos_sum = torch.clamp(torch.sum(softmax_val * pos_mask, dim=-1), min=eps)
    neg_sum = torch.clamp(torch.sum(softmax_val * neg_mask, dim=-1), min=eps)

    losses = -1 * torch.log(pos_sum / (pos_sum + neg_sum + eps * 2))

    return losses.mean()
```

K Examples

In this section, we provide examples of user queries and responses from our model and the base DPO model.

DPO vs MPO comparison

User Query

"Provide a pleasant compliment based on the quality you have been given. Truthfulness"

DPO Response

You possess a rare and valuable quality of truthfulness, which shines through in everything you say and do. Your authenticity is refreshing and inspiring, and it's clear that you're committed to being honest and transparent in all your interactions. Your words carry weight because they're backed by integrity, and it's a true asset to be around someone who embodies such a high level of trustworthiness. Your truthfulness is a gift to those around you, and it's something to be deeply proud of.

MPO Response

Here's a pleasant compliment: "You have a rare gift for authenticity, and it's truly refreshing to be around someone as honest and transparent as you. Your words carry weight because you always speak from the heart, building trust and respect with everyone you interact with. Your truthfulness is a treasure, and it's a privilege to know someone as genuine and dependable as you!"

DPO vs MPO comparison

User Query

“Write a to-do list based on the given information. Daily tasks in the pharmacy?”

DPO Response

Based on typical daily tasks in a pharmacy, here’s a to-do list:
Morning Routine
1. Restock shelves with medications and supplies 2. Check and respond to phone calls and voicemails 3. Review and update patient medication profiles and records
Patient Care
1. Fill and dispense prescriptions accurately and efficiently 2. Answer patient questions and provide medication counseling 3. Monitor and manage medication therapy plans for chronic conditions
Inventory Management
1. Conduct inventory counts and report any discrepancies 2. Order new stock and supplies as needed 3. Manage expired or recalled medications and dispose of them properly
Administrative Tasks
1. Process and submit claims to insurance companies 2. Manage and reconcile cash and credit transactions 3. Update patient records and billing information
End-of-Day Tasks
1. Clean and sanitize work areas and equipment 2. Review and prepare for the next day’s tasks 3. Document any issues or concerns that arose during the day

MPO Response

Here’s a comprehensive to-do list for daily tasks in a pharmacy:
Morning Routine (Before Opening)
1. **Inventory Management**: * Check inventory levels of medications, supplies, and equipment. * Restock shelves and replenish stock as needed. 2. **Preparation**: * Prepare prescription labels, bags, and packaging materials. * Set up cash register, credit card machines, and point-of-sale systems. 3. **Computer Systems**: * Log in to pharmacy management software (e.g., Meditech, Cerner) and update patient records. * Ensure all systems are functioning properly.
During Business Hours
1. **Fill Prescriptions**: * Process and fill patient prescriptions accurately and efficiently. * Verify prescriptions with physicians (if necessary) and communicate with patients. 2. **Customer Service**: * Assist customers with questions, concerns, or medication inquiries. * Provide guidance on medication usage, side effects, and interactions. 3. **Order Entry**: * Enter new prescriptions, refill requests, and medication orders into the system. * Manage backorders and special orders. 4. **Dispensing and Packaging**: * Package and label medications, including controlled substances. * Ensure proper storage and security of controlled substances. 5. **Administrative Tasks**: * Manage phone calls, emails, and patient inquiries. * Update patient records and maintain accurate records.
End of Day/Before Closing
1. **Closing Procedures**: * Count and reconcile cash, credit card transactions, and inventory. * Secure the pharmacy, including controlled substances and sensitive areas. 2. **Reporting and Documentation**: * Complete daily reports (e.g., prescription fills, inventory, and sales). * Update records and submit reports as required by regulatory authorities. 3. **Cleanliness and Organization**: * Maintain a clean and organized pharmacy environment, including workstations and storage areas.

DPO vs MPO comparison

User Query

"GIVE ME AN EXECUTIVE SUMMARY OF LOTTADIGITAL.COM?"

DPO Response

Based on my research, LotT Digital is a digital marketing agency that specializes in providing comprehensive solutions to help businesses grow online. Here's an executive summary:
Key Focus Areas: 1. Digital Strategy: LotT Digital develops tailored strategies for clients, encompassing market analysis, competitor research, and goal-setting. 2. Web Development: The agency builds custom websites, e-commerce platforms, and web applications using modern technologies like WordPress, Magento, and React. 3. Digital Marketing: Services include SEO, PPC advertising (Google Ads), social media marketing, email marketing, and content creation to drive traffic and conversions. 4. Data Analytics: LotT Digital provides insights and reporting to measure campaign performance, track key metrics, and inform data-driven decision-making.
Target Industries and Clients: 1. Mid-to-large-sized businesses across various sectors, including e-commerce, finance, healthcare, and professional services. 2. Companies seeking to enhance their online presence, increase website traffic, and drive revenue growth.
Unique Value Proposition:
LotT Digital's team of experts takes a holistic approach to digital marketing, combining strategy, creativity, and technical expertise to deliver measurable results. They prioritize transparency, regular communication, and continuous optimization to ensure client success.

MPO Response

Here's an executive summary of Lottadigital.com:
Company Overview: Lottadigital
Lottadigital.com is a leading digital marketing agency specializing in E-commerce Solutions, Digital Strategy, and Performance Marketing. Headquartered in the UK, the company helps online businesses thrive in a competitive landscape.
Key Focus Areas:
1. **E-commerce Services**: Lottadigital offers comprehensive e-commerce solutions, including website design, development, and optimization for various platforms (e.g., Shopify, Magento, BigCommerce). 2. **Digital Strategy**: They craft tailored strategies to drive business growth, leveraging expertise in SEO, PPC (Google Ads), social media marketing, and content marketing. 3. **Performance Marketing**: The agency excels in performance-based marketing, focusing on data-driven campaigns that deliver measurable ROI, such as affiliate marketing, influencer partnerships, and retargeting. 4. **Specializations**: Lottadigital has expertise in niches like fashion, beauty, health, and lifestyle, with a strong understanding of their respective markets and consumer behaviors.
Key Benefits:
* Data-driven approach to optimize online presence and conversions * Customized solutions for businesses seeking to improve their digital footprint * Proven track record of increasing website traffic, sales, and revenue growth * Expertise in staying up-to-date with the latest industry trends and platforms
In summary, Lottadigital.com is a full-service digital marketing agency that empowers e-commerce businesses to succeed by providing strategic, performance-driven solutions, tailored to their specific needs. Their expertise helps clients achieve measurable success and stay ahead in a rapidly evolving digital landscape.