# Margin Adaptive DPO: Leveraging Reward Model for Granular Control in Preference Optimization

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Direct Preference Optimization (DPO) has emerged as a simple and effective method for aligning large language models. However, its reliance on a fixed temperature parameter leads to suboptimal training on diverse preference data, causing overfitting on easy examples and under-learning from informative ones. Recent methods have emerged to counter this. While Identity Preference Optimization (IPO) addresses general overfitting, its uniform regularization can be overly conservative. The more targeted approach of $\beta$-DPO suffers from its own limitations: its batch-level adaptation applies a single, compromised temperature to mixed-margin pairs, its linear update rule can produce unstable negative $\beta$ values, and its filtering mechanism discards potentially useful training signals.

In this work, we introduce Margin-Adaptive Direct Preference Optimization (MADPO), a method that provides a stable, data-preserving, and instance-level solution. MADPO employs a practical two-step approach: it first trains a reward model to estimate preference margins and then uses these margins to apply a continuous, adaptive weight to the DPO loss for each individual training sample. This re-weighting scheme creates an effective target margin that is amplified for hard pairs and dampened for easy pairs, allowing for granular control over the learning signal.

We provide a comprehensive theoretical analysis, proving that MADPO has a well-behaved optimization landscape and is robust to reward model estimation errors. We validate our theory with experiments on a sentiment generation task, where MADPO consistently and significantly outperforms strong baselines across datasets of varying quality. It achieves performance gains of up to +33.3% on High Quality data and +10.5% on Low Quality data over the next-best method. Our results establish MADPO as a more robust and principled approach to preference alignment.

## 1 Introduction

Aligning Large Language Models (LLMs) with human preferences has become a cornerstone of modern AI, enabling models that are more helpful and harmless (Bai et al., 2022), follow complex instructions (Ouyang et al., 2022), and excel at sophisticated tasks (Ziegler et al., 2019). The dominant paradigm for this is learning from preference data, which has given rise to a range of powerful techniques. While early successes were driven by multi-stage methods like Reinforcement Learning from Human Feedback (RLHF), the field has recently seen the emergence of more direct and stable approaches, most notably Direct Preference Optimization (DPO) (Rafailov et al., 2023).

While DPO offers a more direct approach to preference alignment, its effectiveness is constrained by a critical factor: the joint influence of the temperature parameter, $\beta$, and the quality of the preference data. Seminal work in this area by Wu et al. (2024b) demonstrated that the optimal choice of $\beta$ is highly contingent on the reward margin of a given pair. Their analysis revealed that easy pairs with a large margin benefit from a high, conservative $\beta$ to prevent overfitting, whereas hard pairs with a subtle margin require a low, aggressive $\beta$ to ensure the learning signal is captured. The vanilla DPO framework, with its single fixed $\beta$ applied to all samples, is fundamentally unable to reconcile these competing requirements. This inherent tension has motivated recent work on adaptive regularization strategies, which aim to tailor the learning objective to the difficulty of each preference pair.

The challenge of adaptive regularization has led to several innovations that improve upon vanilla DPO. Identity Preference Optimization (IPO) (Azar et al., 2024), for instance, effectively mitigates the general overfitting issue by replacing the loss function with a squared-error objective. While not explicitly designed to resolve the tension between high- and low-margin data, its uniform target margin partially addresses the problem by regularizing easy pairs, though at the risk of being overly conservative on more informative examples. The most direct attempt to solve this is $\beta$-DPO (Wu et al., 2024b), which introduces adaptive, batch-level strategies. However, while demonstrating improved results, its mechanisms introduce significant new challenges. Its $\beta$-batch adaptation, for instance, is potentially unstable—producing a divergent negative $\beta$ for difficult data—and applies a single, compromised temperature to mixed-margin batches. Furthermore, its $\beta$-guided filtering approach can be data-inefficient, as it potentially discards very high and low margin samples that may still contain useful learning signals. These issues of potential instability, coarse granularity, and data inefficiency highlight the need for a solution that is not only instance-level and data-preserving, but also inherently stable.

In this paper, to address these challenges, we introduce Margin-Adaptive Direct Preference Optimization (MADPO), a method that precisely controls the DPO objective through a practical two-step process. First, we train a standard reward model to learn how strongly one response is preferred over another for each training example. Our approach then leverages this reward model to guide the DPO policy, which works by learning to match the preferences captured by the reward model. MADPO strategically modifies the strength of the preference signal from the reward model before showing it to the policy. For hard and informative pairs, it amplifies the signal to make the preference seem stronger, forcing the policy to learn more aggressively, achieving the same effect as a low $\beta$. Conversely, for easy and uninformative examples where the preference is already obvious, it dampens the signal to make the preference seem weaker, which provides a stabilizing, per-sample regularization, achieving the same effect as a high $\beta$. This strategic modification of the preference signal allows for granular, instance-level control, making the alignment process more robust and data-efficient.

Our theoretical analysis validates the design of MADPO. We demonstrate that its instance-level weighting scheme successfully regularizes the learning objective for easy preference pairs while amplifying the signal for difficult ones, all while maintaining a well-behaved and stable optimization landscape with bounded gradients. Crucially, we also prove that this granular control is not brittle; our analysis provides a formal guarantee that the practical two-step algorithm is robust to the estimation errors inherent in reward modeling. These theoretical results establish MADPO as a principled and reliable method, a claim we validate with empirical experiment.

We validate our theoretical claims with a series of experiments on a controlled sentiment generation task using the IMDB dataset. Our results demonstrate that MADPO consistently and significantly outperforms strong baselines, including DPO, IPO, and $\beta$-DPO, across all data quality tiers: High, Medium, and Low. Notably, our method shows strong robustness to degrading data quality, with performance gains over the next-best baseline, $\beta$-DPO, ranging from a significant +10.5% on the challenging Low Quality dataset to +33.3% on the High Quality dataset. Further analysis provides deeper insight into our method's mechanics: a detailed ablation study reveals that this robust performance is primarily driven by the amplification mechanism, while a sensitivity analysis demonstrates that the method's hyperparameters are well-behaved and exhibit clear, predictable trends. These empirical findings confirm the claims of our theoretical analysis and establish MADPO as a more reliable and effective method for preference alignment.

## 2 Preliminaries

The goal of preference alignment is to fine-tune a language model policy $\pi_\theta$, parameterized by $\theta$, using a dataset of human preferences $\mathcal{D} = \{(x, y_w, y_l)\}_{i=1}^{N}$. For each prompt $x$, $y_w$ is the response preferred over the response $y_l$. The alignment process is typically framed by modeling the probability of these preferences.

### 2.1 Reinforcement Learning from Human Feedback (RLHF)

The RLHF paradigm (Ouyang et al., 2022) aligns a policy in two main stages: reward modeling and policy optimization.

**1. Reward Modeling.** This stage aims to learn a reward model $r_\phi(x, y)$, parameterized by $\phi$, that reflects human preferences. The probability that $y_w$ is preferred to $y_l$ is modeled using the Bradley-Terry-Luce (BTL) framework (Bradley & Terry, 1952; Luce et al., 1959):

$$\begin{aligned} P(y_w \succ y_l | x; \phi^*) &= \sigma(h_{\phi^*}(x, y_w, y_l)) \\ &= \sigma(r_{\phi^*}(x, y_w) - r_{\phi^*}(x, y_l)), \end{aligned} \tag{1}$$

where $\sigma(\cdot)$ is the logistic function. The optimal reward model parameters, $\phi^*$, are found by maximizing the likelihood of the preference dataset, which corresponds to minimizing the following negative log-likelihood loss:

$$\mathcal{L}_{\mathrm{RM}}(r_\phi) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(h_\phi(x, y_w, y_l))\right]. \tag{2}$$

**2. Policy Optimization.** In the second stage, the policy $\pi_\theta$ is then fine-tuned using the trained reward model $r_{\hat{\phi}}(x, y)$, where $\hat{\phi}$ is the empirical estimate of the optimal parameters $\phi^*$. The policy is optimized to maximize the expected reward while being regularized by a Kullback-Leibler (KL) divergence penalty against a reference policy $\pi_{\mathrm{ref}}$:

$$\max_\theta \quad \mathbb{E}_{x\sim\mathcal{D}_p, y\sim\pi_\theta(y|x)}[r_{\hat{\phi}}(x, y)] - \beta D_{\mathrm{KL}}(\pi_\theta(y|x)||\pi_{\mathrm{ref}}(y|x)).$$

Here, $\beta$ is a hyperparameter that controls the strength of the KL regularization, and $\mathcal{D}_p$ represents the dataset of prompts.

### 2.2 Direct Preference Optimization (DPO)

DPO (Rafailov et al., 2023) is an alternative that bypasses the explicit reward modeling and reinforcement learning stages. The key insight is that the optimal solution to the KL-regularized objective has a closed-form solution that connects the optimal policy $\pi_{\theta^*}$ to the optimal reward function $r_{\phi^*}$:

$$r_{\phi^*}(x, y) = \beta \log \frac{\pi_{\theta^*}(y|x)}{\pi_{\mathrm{ref}}(y|x)} + \beta \log Z(x), \tag{3}$$

where $Z(x)$ is the partition function that normalizes the distribution.

By substituting this mapping (Eq. 3) into the BTL preference model, the likelihood can be expressed directly in terms of the policy $\pi_\theta$. This allows for end-to-end optimization of the policy by minimizing a single negative log-likelihood loss:

$$\begin{aligned} \mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta h_\theta(x, y_w, y_l)\right)\right] \\ &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}\right)\right]. \end{aligned}$$

For notational clarity, we define the following reward margin functions:

- The **explicit reward margin** from a reward model $r_\phi$:

$$h_\phi(x, y_w, y_l) = r_\phi(x, y_w) - r_\phi(x, y_l).$$

- The **implicit reward margin** from a policy $\pi_\theta$:

$$h_\theta(x, y_w, y_l) = \log \frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}.$$

### 2.3 Limitations of Vanilla DPO

The relationship between the true reward function and the optimal policy is well-defined. From Eq. 3, we can see that the explicit reward margin is proportional to the implicit reward margin:

$$h_{\phi^*}(x, y_w, y_l) = \beta h_{\theta^*}(x, y_w, y_l).$$

Here, the inverse temperature $\beta$ acts as a global hyperparameter. A smaller $\beta$ encourages a larger difference in the policy's log-ratios for a given explicit reward margin, promoting more aggressive, confident updates. Conversely, a larger $\beta$ encourages more conservative updates.

However, recent work has revealed that using a single, static $\beta$ as a global hyperparameter is a significant limitation of the vanilla DPO framework, often leading to overfitting. As argued by Azar et al. (2024), this issue is particularly acute on finite datasets. If all annotators in a sample unanimously prefer one response, the empirical explicit reward margin becomes infinite. To match this, the DPO objective will push the learned log-policy difference, $h_{\hat{\theta}}$, to be arbitrarily large, causing the model to become overconfident and overfit to the winning response.

Complementing this finding, Wu et al. (2024b) suggest that a single $\beta$ is insufficient for handling the diverse quality of preference data. They find that easy pairs with a large explicit reward margin are best handled with a high $\beta$ (a more conservative update) to prevent overfitting. In contrast, hard pairs with a small, subtle margin require a low $\beta$ (a more aggressive update) to effectively learn the preference signal. This tension reveals the need for a more dynamic, instance-aware approach to regularization, which motivated the development of subsequent methods like IPO and $\beta$-DPO.

### 2.4 Identity Preference Optimization (IPO)

Identity Preference Optimization (IPO) (Azar et al., 2024) addresses the overfitting issue by replacing the log-likelihood objective with a squared-error loss,

$$\mathcal{L}_{\text{IPO}}(\pi_\theta, \pi_{ref}) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\left(h_\theta(x, y_w, y_l) - \frac{1}{2\beta}\right)^2\right].$$

The mechanism of IPO can be understood by analyzing the optimality condition of its loss function. The squared-error loss is minimized for any given sample when the implicit reward margin satisfies $\beta h_{\theta^*}(x, y_w, y_l) = 1/2$. Unlike DPO, which attempts to match the true explicit reward margin $h_{\phi^*}$, IPO effectively sets a single, uniform target margin of $1/2$ for every preference pair in the dataset. This has a dual effect: for hard pairs where the true explicit margin is small ($h_{\phi^*} < 1/2$), the higher target margin amplifies the learning signal. Conversely, for easy pairs where the true explicit margin is large ($h_{\phi^*} > 1/2$), the low target margin aggressively dampens the signal, which is the source of the regularization.

### 2.5 $\beta$-DPO

To address the limitations of a fixed temperature, $\beta$-DPO (Wu et al., 2024b) introduces adaptive strategies to modulate the learning process. It proposes two primary mechanisms:

**Batch-Level $\beta$ Adaptation ($\beta$-batch).** This approach adapts the temperature $\beta$ for each training batch using a linear function of the batch's average implicit reward margin, $\bar{h}_\theta$. The batch-specific temperature, $\beta_{\text{batch}}$, is set as:

$$\beta_{\text{batch}} = \beta(1 + m \cdot (\bar{h}_\theta - h_0)), \quad \text{where} \quad \bar{h}_\theta = \frac{1}{|\text{batch}|}\sum_{(x,y_w,y_l)\in\text{batch}} h_\theta(x, y_w, y_l).$$

Here, $\beta$ is a base temperature, $m$ is a scaling factor between zero and one, and $h_0$ is a predetermined threshold. This allows batches with higher average implicit margins to be trained with a higher, more conservative $\beta_{\text{batch}}$.

$\beta$ **guided filtering.** This approach modifies the training data by stochastically filtering each batch. For each sample $(x, y_w, y_l)$ in a batch, a score is computed using the probability density function of a Normal distribution, $\mathcal{N}(\cdot\,; h_0, \sigma^2)$, evaluated at the policy's current implicit margin, $h_\theta(x, y_w, y_l)$. The score is highest for samples whose margin is close to the mean $h_0$. A new, smaller batch is then formed by performing weighted random sampling without replacement from batch, where the probability of selecting a sample is proportional to its score. This method dynamically focuses training on examples of a target difficulty level, effectively down-sampling both overly easy and potentially noisy pairs.

## 3 Margin Adaptive Direct Preference Optimization (MADPO)

In this section, we introduce Margin Adaptive Direct Preference Optimization (MADPO), a method that enhances the DPO objective by adaptively re-weighting each training sample. The core idea is to modulate the loss based on the explicit reward margin, $h_\phi$, to amplify the learning signal from informative low-margin pairs while dampening it for easy, high-margin pairs to prevent overconfidence. This provides a more granular and flexible approach to regularization than IPO and $\beta$-DPO. We begin by detailing the central component of our method: a continuous, margin adaptive weight function. We then define the full MADPO loss function and describe the practical two-step algorithm for its optimization.

### 3.1 Margin Adaptive Weight

The central component of our MADPO method is the weight function which adaptively modulates the learning objective for each training sample based on its preference margin. This subsection provides a detailed exposition of this function's design, its hyperparameters, and the reasoning behind its piecewise structure. To simplify the notation, where the context is unambiguous, any function will be denoted by its symbol alone, suppressing the explicit dependence on its arguments for conciseness.

The core of our method is a coefficient function, $c : \mathbb{R} \to [c_{\min}, c_{\max}]$, which maps the explicit reward margin $h_\phi$ to a modulating scalar. This function is designed to be greater than 1 for low margins and less than 1 for high margins. It is defined as:

$$c(h_\phi) = c_{\min} + \frac{c_{\max} - c_{\min}}{1 + \left(\frac{c_{\max}-1}{1-c_{\min}}\right) \exp\left(\lambda(h_\phi - \tau)\right)},$$

Using this margin-dependent coefficient, we define a piecewise weighting function, $w(h_\phi)$, which selectively modifies the likelihood ratio.

$$w(h_\phi) = \begin{cases} \frac{\sigma(c(|h_\phi|)\cdot h_\phi)}{\sigma(h_\phi)} & \text{if } h_\phi > -\tau \\ 1 & \text{if } h_\phi \leq -\tau \end{cases} \tag{4}$$

The threshold $\tau > 0$ provides a concrete definition for what constitutes a high-margin or easy preference pair. This value can be chosen based on practitioner judgment or derived from the data. Specifically, any pair is classified as high-margin if its absolute explicit reward margin satisfies $|h_\phi| \geq \tau$, and as low-margin if its explicit margin satisfies $|h_\phi| < \tau$.

The parameter $c_{\max}$ acts as the amplifier for low-margin pairs. As the explicit margin $|h_\phi|$ approaches zero, the coefficient approaches $c_{\max}$. This forces the model to learn more aggressively from the most informative and subtle preferences. A higher value provides a stronger signal boost, but risks overfitting to noise in these difficult examples. By definition, this parameter must be greater than one to ensure amplification.

The parameter $c_{\min}$ acts as the dampener for high-margin pairs, setting a floor on the regularization. As the margin grows, the learned target gap is scaled down by a factor approaching $c_{\min}$. A value near 0 instructs the model to almost entirely ignore obvious preferences, preventing overconfidence. A value closer to 1 instructs the model to still learn from them, but with less intensity. By definition, this parameter must be between zero and one, $c_{\min} \in [0, 1)$, to ensure damping.

The parameter $\lambda$ controls the sharpness of the transition around the threshold $\tau$. A large $\lambda$ creates a steep, switch-like change, treating samples on either side of $\tau$ very differently. A small $\lambda$ creates a much more gradual and smooth transition from amplification to dampening.

While these intuitions provide strong starting points for setting the parameters, their optimal values are typically dataset-dependent. Therefore, the most rigorous approach is to perform a hyperparameter search, using a method like cross-validation on a held-out set of preference data to find the combination that yields the best empirical performance.

The piecewise nature of the weight function is a crucial design choice for ensuring training stability. While the re-weighting mechanism works as intended for most pairs, it can lead to undesirable behavior at the extremes of the margin distribution without this piecewise control.

We analyze the behavior of the core ratio $\sigma(c(h_{\phi^*}|) \cdot h_{\phi^*})/\sigma(h_{\phi^*})$, evaluated at the optimal reward model parameter $\phi^*$, in two distinct cases:

- **For large positive margins ($h_{\phi^*} \gg \tau$):** In this region, where $c \approx c_{\min}$, the weight function converges to a small, stable, positive value. This correctly applies a consistent penalty to all easy pairs, achieving the desired regularization effect.
- **For large negative margins ($h_{\phi^*} \ll -\tau$):** Without the piecewise cutoff, the weight function would explode. As $h_{\phi^*} \to -\infty$, the weight can be approximated by $w(h_{\phi^*}) \approx e^{(c_{\min}-1)h_{\phi^*}}$. Since $c_{\min} < 1$ and $h_{\phi^*}$ is negative, the exponent is positive and grows linearly with $|h_{\phi^*}|$. This growth in the weight would assign a massive, potentially infinite loss to these samples, leading to severe gradient instability.

The piecewise definition elegantly solves this problem. By setting $w(h_{\phi^*}) = 1$ for all $h_{\phi^*} \leq -\tau$, we cap this potential explosion. This ensures that samples with very large negative margins—which may be mislabeled or adversarial—are handled by the vanilla, stable DPO loss instead of causing the training to diverge. This design allows us to achieve our desired penalization for high positive margins without sacrificing the stability of the overall training process. The impact of this weighting on the final loss function is discussed in the following sections.

### 3.2 Loss Function and Optimization

Having defined the margin adaptive weight, $w(h_\phi)$, we now formally incorporate it into our final loss function. We then detail the practical, two-step procedure used to train a policy with this new objective.

**The MADPO Loss Function.** The MADPO loss for a single preference pair is the vanilla DPO log-likelihood, re-weighted by our margin-dependent weight function:

$$\mathcal{L}(\theta, \phi; x, y_w, y_l) = -w(h_\phi(x, y_w, y_l)) \log \sigma(\beta h_\theta(x, y_w, y_l)). \tag{5}$$

This loss depends on both the policy parameters, $\theta$, (through $h_\theta$) and the reward model parameters, $\phi$, (through $h_\phi$). To optimize this effectively, we employ a two-step approach.

**Step 1: Reward Model Estimation.** First, we obtain a high-quality estimate of the preference margins. This is achieved by training a standard reward model, $r_\phi$, on the preference dataset $\mathcal{D}$ to find the estimated parameters, $\hat{\phi}$. This step is identical to the reward modeling stage of traditional RLHF:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} \; \mathcal{L}_{\mathrm{RM}}(r_\phi).$$

**Step 2: Margin Adaptive Policy Optimization.** Second, we treat the estimated reward parameters $\hat{\phi}$ as a fixed, ground-truth source of preference margins. These parameters are plugged into our MADPO loss function, which now becomes an objective solely for the policy. We then find the final policy parameters, $\hat{\theta}$, by minimizing the expectation of this loss over the dataset:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \; \mathcal{L}(\theta, \hat{\phi}) = \underset{\theta}{\operatorname{argmin}} \; \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \mathcal{L}(\theta, \hat{\phi}; x, y_w, y_l) \right].$$

This two-step process provides a stable and practical method for training a policy that is explicitly aware of the nuance and difficulty of the preference data it learns from.

## 4 Theoretical Analysis

In this section, we provide the theoretical justification for the MADPO algorithm. Our analysis proceeds in three parts. First, we establish that our method achieves its primary goal: in an idealized oracle setting, MADPO encourages the policy to be confident on informative, low-margin pairs while achieving the desired regularization for high-margin pairs. Second, we prove a formal performance guarantee for our practical two-step algorithm, showing that the loss function is Lipschitz continuous and therefore robust to errors from the reward estimation stage. Finally, we demonstrate that the gradient and Hessian of the MADPO loss are scaled versions of their vanilla DPO counterparts, which shows that our method provides controllable training stability while retaining the well-behaved optimization landscape of DPO.

### 4.1 Oracle Characterization of Margin-Adaptive Regularization

In this section, we formally characterize the behavior of the MADPO loss function, showing how it achieves the dual goals of amplifying the learning signal for informative low-margin pairs and regularizing the objective for high-margin pairs. To isolate this mechanism, we conduct our analysis in an oracle setting, assuming access to the true optimal reward parameters, $\phi^*$.

We begin with our first proposition, which characterizes how MADPO aggressively learns from low-margin pairs by optimizing the policy towards an amplified preference target.

**Proposition 4.1.** *Under the BTL model with an optimal reward model $r_{\phi^*}$, the optimal policy parameter $\theta^*$ that minimizes the MADPO loss $\mathcal{L}(\theta, \phi^*; x, y_w, y_l)$ satisfies the following for any preference pair $(x, y_w, y_l) \in \mathcal{D}_{low}$:*

$$\beta h_{\theta^*}(x, y_w, y_l) = c(|h_{\phi^*}(x, y_w, y_l)|) \cdot h_{\phi^*}(x, y_w, y_l),$$

*where the low-margin subset $\mathcal{D}_{low}$ is defined as:*

$$\mathcal{D}_{low} = \{(x, y_w, y_l) \in \mathcal{D} \mid |h_{\phi^*}(x, y_w, y_l)| < \tau\}.$$

This proposition formalizes the amplification mechanism of MADPO for low-margin data. For any sample in the subset $\mathcal{D}_{\text{low}}$, the coefficient $c(|h_{\phi^*}|)$ is greater than one by construction. Consequently, the optimality condition reveals that the policy is not optimized to simply match the true explicit reward margin, $h_{\phi^*}$, but rather an amplified version of it, $c(|h_{\phi^*}|) \cdot h_{\phi^*}$. This encourages the policy to learn more aggressively from these subtle and informative examples, increasing the separation in its log-ratios to a greater degree than what the true reward margin alone would suggest. This is the core mechanism by which MADPO boosts the learning signal for hard, informative pairs.

Having characterized the amplification mechanism for low-margin data, we now turn our attention to the complementary case: how MADPO achieves regularization for high-margin pairs.

**Proposition 4.2.** *Under the BTL model with an optimal reward model $r_{\phi^*}$, the optimal implicit reward $\beta h_{\theta^*}$ is monotonically increasing with respect to the coefficient $c \equiv c(|h_{\phi^*}|)$ for any preference pair $(x, y_w, y_l) \in \mathcal{D}_{high}$. Formally:*

$$\frac{\partial(\beta h_{\theta^*})}{\partial c} > 0,$$

*where the high-margin subset $\mathcal{D}_{high}$ is defined as:*

$$\mathcal{D}_{high} = \{(x, y_w, y_l) \in \mathcal{D} \mid |h_{\phi^*}(x, y_w, y_l)| \geq \tau\}.$$

This proposition provides the formal justification for the regularization mechanism of MADPO on high-margin data. It establishes that the learned implicit reward margin, $\beta h_{\theta^*}$, is directly and monotonically controlled by the coefficient $c$. For any preference pair in the high-margin subset $\mathcal{D}_{\text{high}}$, our method sets $c(|h_{\phi^*}|)$ to a value less than one. The monotonic relationship guarantees that this shrinks the optimization target, $\beta h_{\theta^*}$, relative to the true explicit reward margin, $h_{\phi^*}$. This acts as a powerful regularization tool, dampening the learning signal for easy pairs and preventing the policy from becoming overconfident or overfitting to these less informative examples.

Our theoretical results establish that MADPO offers a more granular, per-pair control over the learned preference margin compared to the global mechanisms of IPO and $\beta$-DPO. Propositions 4.1 and 4.2 formalize this: for each preference pair, the optimal policy learns to match a dynamically scaled target, $c(|h_{\phi^*}|)h_{\phi^*}$, and this target margin can be monotonically controlled by the coefficient $c$. This provides a sharp, per-example comparative-statics guarantee.

In contrast, IPO regularizes globally (a uniform mechanism that does not adapt to each pair's margin), and $\beta$-DPO adapts a batch-level temperature $\beta$ shared across all samples in a batch. Neither provides the sample-specific, per-pair control formalized by Propositions 4.1 and 4.2.

### 4.2 Lipschitz Continuity and Robustness to Reward Estimation Error

Having analyzed our method in an ideal oracle setting, we now establish its robustness to the reward estimation errors that occur in practice. To do so, we prove that the MADPO loss function is Lipschitz continuous with respect to the reward model parameters, culminating in a formal performance guarantee that bounds the impact of these errors.

Our analysis in this section follows the theoretical framework established by Chowdhury et al. (2024). We adopt the following assumptions from their work to analyze the stability of our method.

**Assumption 4.3.** *We assume the following constraints on the reward model's parameter space, $\Phi$:*

- *The parameter space is defined as $\Phi = \{\phi \in \mathbb{R}^\delta \mid \sum_{i=1}^{\delta} \phi_i = 0\}$.*
- *For any parameter vector $\phi \in \Phi$, there exists a constant $B > 0$ such that its Euclidean norm is bounded: $\|\phi\| \leq B$.*

This assumption places two standard constraints on the reward model's parameter space. The first, the zero-mean condition ($\sum \phi_i = 0$), is necessary for identification. Because the preference probability in the BTL model depends only on the difference in rewards, $r_\phi(x, y_w) - r_\phi(x, y_l)$, the underlying reward function is only unique up to an arbitrary constant shift. This constraint resolves the inherent shift ambiguity, ensuring the identification of a reward function. The second condition, boundedness ($\|\phi\| \leq B$), is a standard regularity assumption required in most theoretical analyses to ensure the parameter space is well-behaved, forming a necessary prerequisite for the performance guarantees that follow.

**Assumption 4.4.** *The reward function $r_\phi(x, y)$ is assumed to be well-behaved with respect to its parameters. Specifically, there exist constants $\alpha_0, \alpha_1, \alpha_2 > 0$ such that for any $\phi \in \Phi$ and any sample $(x, y)$, the function, its gradient, and its Hessian are uniformly bounded:*

$$|r_\phi(x, y)| \leq \alpha_0, \ \|\nabla_\phi r_\phi(x, y)\| \leq \alpha_1, \ \nabla_\phi^2 r_\phi(x, y) \preceq \alpha_2 I.$$

This assumption imposes standard smoothness and boundedness conditions on the reward function, $r_\phi$. These conditions are crucial as they ensure that the explicit reward margin function, $h_\phi(x, y_w, y_l) = r_\phi(x, y_w) - r_\phi(x, y_l)$, is also well-behaved. Specifically, they imply that $h_\phi$ is bounded and Lipschitz continuous with respect to its parameters $\phi$, and that its gradient is also Lipschitz continuous. Such regularity assumptions are a common prerequisite for establishing performance guarantees in the analysis of policy optimization algorithms and are consistent with the theoretical frameworks used in related work (Agarwal et al., 2021; Chowdhury et al., 2024).

**Assumption 4.5.** *There exists a constant $L_\theta > 0$ such that for any policy parameters $\theta$ in the parameter space $\Theta$ and for any sample $(x, y_w, y_l)$, the absolute value of the log-likelihood term is bounded:*

$$|\log \sigma(\beta h_\theta(x, y_w, y_l))| \leq L_\theta.$$

This is a standard technical assumption that is well-justified. It is a mild condition, as it is fundamentally a constraint that the policy, $\pi_\theta$, cannot assign a probability of exactly 0 or 1 to any response. Furthermore, this condition is not arbitrary; it can be derived by imposing boundedness constraints on the policy function, $\pi_\theta(y|x)$, that are directly analogous to those we placed on the reward model in Assumption 4.4. This approach is consistent with similar assumptions made in prior theoretical analyses of preference-based learning, including the framework established by Chowdhury et al. (2024).

The following theorem provides a high-probability bound that connects the estimation error of the reward model to the stability of our final loss function. This error is measured in a data-dependent semi-norm, $\|\cdot\|_{\hat{\Sigma}_\phi}$, which is induced by the empirical covariance of the reward gradients.

Let the gradient vectors $\mathbf{z}_i$ be defined with respect to the reward model parameters $\phi$:

$$\mathbf{z}_i = \nabla_\phi h_\phi(x_i, y_{w,i}, y_{l,i}).$$

Then, the empirical covariance matrix $\hat{\Sigma}_\phi$ is given by:

$$\hat{\Sigma}_\phi = \frac{1}{N}\sum_{i=1}^{N} \mathbf{z}_i \mathbf{z}_i^\top.$$

**Theorem 4.6.** *Under Assumptions 4.3, 4.4 and 4.5, let $\rho \in (0,1]$ and $\kappa > 0$. Then, with a probability of at least $1-\rho$, the following bound holds for the MADPO loss function, for all $(x, y_w, y_l) \in \mathcal{D}$*

$$\begin{aligned}\left|\mathcal{L}(\theta,\phi^*;x,y_w,y_l) - \mathcal{L}(\theta,\hat{\phi};x,y_w,y_l)\right| &\le L\cdot\|\hat{\phi}-\phi^*\|_{\hat{\Sigma}_{\phi^*}+\kappa I} \\ &\le \frac{L\cdot C}{\gamma}\sqrt{\frac{\delta+\log(1/\rho)}{N}} + C'\cdot B\sqrt{\kappa+\frac{\alpha_2}{\gamma}+\alpha_1\alpha_2 B},\end{aligned}$$

*where $L$ is the Lipschitz constant, $C$ and $C'$ are absolute constants, and $\gamma$ is a constant dependent on the bound of the reward function:*

$$\gamma = \frac{1}{2+e^{-4\alpha_0}+e^{4\alpha_0}}.$$

This result certifies the plug-in stability of our practical, two-stage MADPO pipeline. It guarantees that the training objective remains stable even when using an estimated reward model, $\hat{\phi}$, instead of the optimal, $\phi^*$. Concretely, Theorem 4.6 shows that for any given preference pair, the gap between the oracle loss $\mathcal{L}(\theta,\phi^*;x,y_w,y_l)$ and the plug-in loss $\mathcal{L}(\theta,\hat{\phi};x,y_w,y_l)$ is controlled linearly by the reward-parameter error. This guarantee is per-sample and uniform over the dataset, which is a stronger claim than a statement about the average-case error. This means every mini-batch, curriculum subset, or the full empirical objective inherits the same deviation control.

The bound is also operational, as it reveals the key levers that ensure MADPO is robust in practice. The two terms in the bound expose what makes the plug-in procedure reliable:

- **Data Quality and Quantity:** The first term decays at the familiar $O(\sqrt{(\delta+\log(1/\rho))/N})$ rate with more data. It also improves with more informative data that leads to a well-conditioned covariance matrix $\hat{\Sigma}_\phi$.
- **Regularization and Boundedness:** The second term shows that gentle regularization $\kappa$ stabilizes learning in directions where data is sparse. Furthermore, the constant $\gamma$, derived from the bounded-reward assumption, prevents the logistic loss from saturating, ensuring that small errors in $\hat{\phi}$ do not cause disproportionately large swings in the objective.

Finally, this theorem complements our earlier results. Propositions 4.1 and 4.2 characterize what MADPO learns at the pair level (amplifying low margins, shrinking high ones); Theorem 4.6 guarantees how reliably that mechanism survives the reality of an estimated reward model. In short, the pairwise control that defines MADPO is not brittle. Under the stated regularity conditions, the two-stage procedure is uniformly robust to reward estimation error, making the method dependable for the training regimes used in practice.

### 4.3 Smoothness & Curvature: MADPO vs. DPO

In this subsection, we compare the smoothness and curvature of the MADPO loss to the vanilla DPO loss. We show that the gradient and Hessian of our objective are simply a re-scaled version of the DPO derivatives, which confirms that our method has a stable and well-behaved optimization landscape.

**Proposition 4.7.** *Let $\mathcal{L}(\theta, \phi; x, y_w, y_l)$ be the MADPO loss function with bounded, $\theta$ independent weight $0 < w(h_\phi) \leq w_{max}$. Then, for any sample $(x, y_w, y_l) \in \mathcal{D}$, the first and second derivatives of the loss with respect to the implicit reward margin, $h_\theta$, satisfy the following bounds:*

1. ***Bounded Gradient:***
$$\left| \frac{\partial \mathcal{L}}{\partial h_\theta} \right| \leq w_{max}\beta.$$

2. ***Bounded Hessian:***
$$\left| \frac{\partial^2 \mathcal{L}}{\partial h_\theta^2} \right| \leq \frac{w_{max}\beta^2}{4}.$$

This proposition reveals that MADPO is a principled modification of the vanilla DPO framework. It shows that the scalar gradient and Hessian of the MADPO loss are simply the vanilla DPO derivatives multiplied by our bounded weight, $w(h_\phi)$. This direct scaling is crucial because it ensures MADPO preserves the benign optimization geometry of the original DPO objective, which has intrinsically capped sensitivity and curvature. For practitioners, this translates to predictable gradients that are compatible with standard control techniques like learning rate tuning or gradient clipping. Crucially, because these derivatives are bounded on a per-sample basis, it guarantees that the instance-level amplification and regularization effects established in our prior propositions are applied in a stable and reliable manner.

## 5 Experiment

In this section, we present the empirical evaluation of our proposed method, MADPO. We first detail our experimental setup, which is designed to test the performance and robustness of our algorithm against strong baselines on datasets of varying quality. We then present and thoroughly analyze the quantitative and qualitative results of these experiments.

### 5.1 Experiment Setup

The goal of our experiments is to evaluate how effectively different preference alignment algorithms can teach a base language model a specific stylistic trait: to consistently generate positive-sentiment responses. To test this in a controlled manner, we design a synthetic environment inspired by the methodology of Chowdhury et al. (2024), which allows us to compare our method against standard baselines across datasets of varying difficulty. All models were trained using LoRA (Hu et al., 2022).

**Models and Datasets.** Our setup uses a combination of a base language model for fine-tuning, a powerful existing model to serve as a ground-truth reward oracle, and a standard text dataset for prompts and content.

- **Base Language Model:** We use `google/gemma-3-270M` (Team, 2025) as the base model for all fine-tuning tasks.

- **Ground-Truth Reward Model:** To simulate human preferences with a known reward function, we use `cardiffnlp/twitter-roberta-base-sentiment-latest` (Loureiro et al., 2022), a strong sentiment analysis model, as our oracle.

- **Text Corpus:** All prompts and responses are derived from the `stanfordnlp/imdb` dataset (Maas et al., 2011).

**Training Procedure** Our full experimental pipeline consists of four sequential stages: supervised fine-tuning of a base model, generation of synthetic preference data, training a reward model on this data, and finally, fine-tuning a policy using a preference alignment algorithm.

1. **Supervised Fine-Tuning (SFT):** We first create a base policy with a strong stylistic prior. To do this, we fine-tune the model on the final 12,000 positive-sentiment reviews from the IMDB training set. The model is trained to generate positive-sentiment text when prompted with the beginning of a review. This SFT model serves as the starting point for all subsequent policy fine-tuning.
2. **Synthetic Data Generation:** The SFT model is then used to generate a preference dataset of 12,000 pairs. We prompt the SFT model with the first 12,000 negative-sentiment reviews from the IMDB training dataset, forcing it to generate positive text in a negative context. These generations are then used to construct three distinct preference datasets of varying quality (High, Medium, and Low). The precise methodology for this process is detailed in the following paragraph.
3. **Reward Model Training:** For each of the three quality-tiered datasets, we train a separate reward model. The reward model is trained on the first 10,000 preference pairs of its corresponding dataset. The final 2,000 pairs are held out as a validation set to implement early stopping, ensuring the reward model does not overfit to the training data.
4. **Policy Fine-Tuning:** In the final stage, we take the SFT model from Step 1 and fine-tune it using one of the preference alignment algorithms (DPO, IPO, $\beta$-DPO, and our proposed MADPO). Each policy is trained for two epochs on the same 10,000-pair training set that was used to train its corresponding reward model for that quality tier.

**Data Generation and Quality Tiers.** To create a challenging and diverse preference dataset, we generate our preference data by prompting this optimistic SFT model with the 12,000 negative-sentiment reviews. For each negative prompt, the SFT model generates two distinct positive-leaning responses. From these, we construct three dataset versions with varying quality:

- **High Quality:** Both responses in each pair, $(y_1, y_2)$, are generations from the SFT model.
- **Medium Quality:** For the first 6,000 pairs, one response is a high-scoring generation from the SFT model and the other is a real, negative review from the IMDB dataset. The remaining 6,000 pairs are high-quality (SFT vs. SFT).
- **Low Quality:** For all 12,000 pairs, one response is a high-scoring SFT generation and the other is a real, negative review.

For each generated pair, we score both responses using the ground-truth RoBERTa model. Specifically, we extract the probability of the 'positive' label, $p$, and apply the linear transformation $f(p) = 6(p - 0.5)$ to map it to a reward value in the range $[-3, 3]$. This range was chosen to be wide enough to generate a meaningful distribution of reward margins, yet narrow enough that the BTL model in our choice simulation remains impactful, ensuring the resulting preferences are probabilistic rather than deterministic. We then simulate a human choice by adding independent and identically distributed Gumbel noise to the reward score of each of the two responses. The response with the highest resulting noisy score is then selected as the winner for that preference pair, a process which is consistent with the BTL preference model (McFadden, 2001). To ensure a controlled comparison, all three 12,000-pair datasets are randomly shuffled using the same fixed permutation, guaranteeing that the training and evaluation splits contain the identical set of prompts across each quality tier. From each shuffled dataset, the first 10,000 pairs are used for training, and the prompts from the remaining 2,000 are reserved for evaluation.

**Baselines and Hyperparameters.** We compare MADPO against several standard baselines: DPO, IPO, and $\beta$-DPO (applying both $\beta$-batch and $\beta$ guided filtering). For all methods, we use a fixed temperature of $\beta = 0.1$.

We perform a targeted hyperparameter search for both $\beta$-DPO and our proposed method, MADPO. For $\beta$-DPO, we tune the $\beta$-batch scaling factor $m \in \{0.4, 0.6, 0.8\}$ and the $\beta$-filtering proportion $p \in \{0.4, 0.6, 0.8\}$. Instead of a full Cartesian product, we conduct a search over 5 configurations for each dataset quality: we first fix $m = 0.6$ while varying $p$, and then fix $p = 0.8$ while varying $m$. For the filtering mechanism, the target margin is set to $h_0 = 0$ with an initial standard deviation of one and a momentum coefficient of 0.9. For our method, MADPO, we perform a similar targeted search over 6 configurations. The steepness is fixed at $\lambda = 1$ and the minimum coefficient is set as the reciprocal of the maximum, $c_{\min} = 1/c_{\max}$. We search over the threshold $\tau \in \{2, 4, 7, 10\}$ and the maximum coefficient $c_{\max} \in \{2, 3, 4\}$. Specifically, we first fix $\tau = 7$ while varying $c_{\max}$, and then fix $c_{\max} = 2$ while varying $\tau$.

**Evaluation.** For each method and on each of the three quality-tiered datasets, we fine-tune a separate policy. We evaluate the performance of each final policy by using it to generate responses for the 2,000 held-out evaluation prompts. The primary metric for comparison is the mean ground-truth reward of these generated responses, as scored by our RoBERTa oracle. A higher mean reward indicates a better-aligned policy.

## 5.2 Experiment Result

In this subsection, we present a series of experiments designed to empirically validate the effectiveness and robustness of our proposed method. We compare MADPO against strong preference alignment baselines on synthetic datasets of varying quality to specifically test its performance in challenging, noisy settings. The section is organized as follows: first, we present the main quantitative results comparing all methods across the three data quality tiers. Second, we provide a detailed sensitivity analysis on MADPO's key hyperparameters. Third, we conduct an ablation study to disentangle the contributions of our method's core components.

**(a) Main Results Table**

| Method | High | Medium | Low |
|---|---|---|---|
| DPO | 1.62<br>(0.03) | 1.71<br>(0.03) | 1.48<br>(0.04) |
| IPO | 0.35<br>(0.05) | 0.31<br>(0.05) | 0.10<br>(0.05) |
| $\beta$-DPO | 1.67<br>(0.03) | 1.84<br>(0.03) | 1.76<br>(0.03) |
| **MADPO** | **2.23**<br>(0.02) | **2.23**<br>(0.02) | **1.95**<br>(0.03) |

**(b) Performance Visualization**

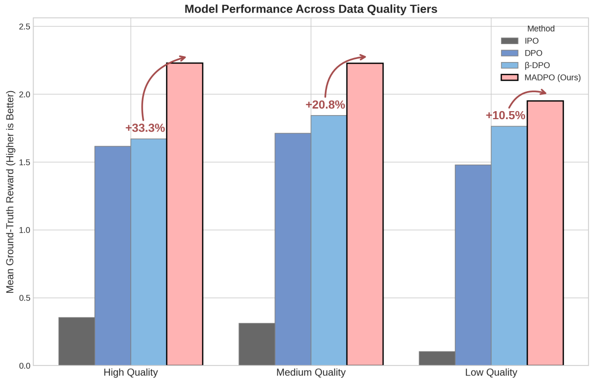


Figure 1: Main experimental results. **(a)** Table of mean rewards (standard error) for all methods across three data quality tiers. **(b)** Bar chart visualizing the mean rewards, clearly showing MADPO's superior performance and robustness compared to baselines. For $\beta$-DPO and MADPO, we report the performance of the best hyperparameter configuration found for each individual tier.

**Main Result.** The main experimental results, visualized in Figure 1, clearly demonstrate the superior performance and robustness of our proposed method, MADPO. Our method achieves the highest mean reward across all three data quality tiers, substantially outperforming all baselines. The annotations on the chart highlight the significant margin of victory over the next-best method, $\beta$-DPO, with performance gains of +33.3% on High Quality, +20.8% on Medium Quality, and +10.5% on Low Quality data. Notably, MADPO's absolute performance is also remarkably stable, showing no degradation between the High and Medium quality settings and maintaining a strong lead on the most challenging Low Quality dataset.

The performance of the baseline models validates our experimental design. The vanilla DPO model's score is volatile and drops on the Low Quality set, confirming the difficulty of this tier. While $\beta$-DPO consistently improves upon DPO, it is clearly outperformed by MADPO's instance-level approach and shows a sensitivity to the data mixture, with its peak performance occurring on the Medium Quality dataset. IPO performs poorly in all scenarios, indicating that its uniform regularization strategy is ill-suited for this task.

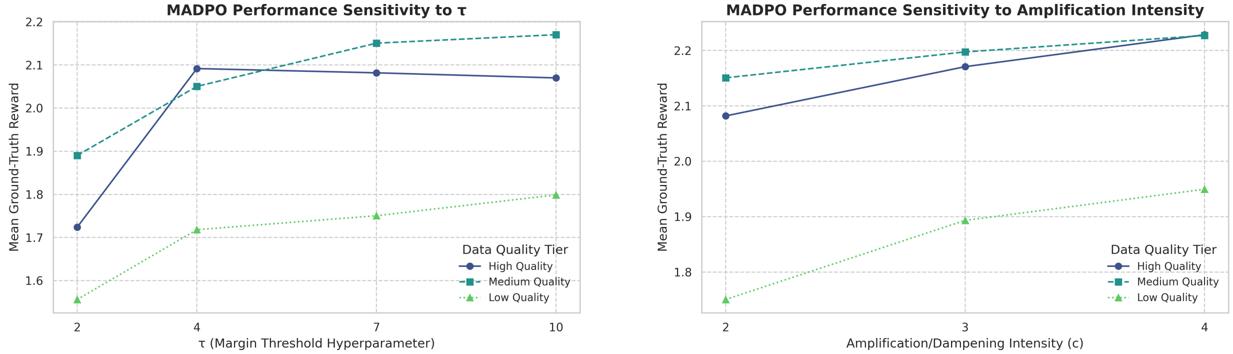


Figure 2: Sensitivity analysis for MADPO's key hyperparameters across the three data quality tiers. **(Left)** Performance as a function of the margin threshold, $\tau$. Higher values are generally better, though performance plateaus on High Quality data. **(Right)** Performance as a function of the amplification intensity, $c$, where we set $c_{\max} = c$ and $c_{\min} = 1/c$. Performance consistently improves with higher intensity across all tiers.

**Sensitivity Analysis.** To understand the behavior of MADPO with respect to its key hyperparameters, we conduct a sensitivity analysis on the margin threshold, $\tau$, and the amplification intensity, $c$. The results, shown in Figure 2, reveal several key insights into the method's mechanics.

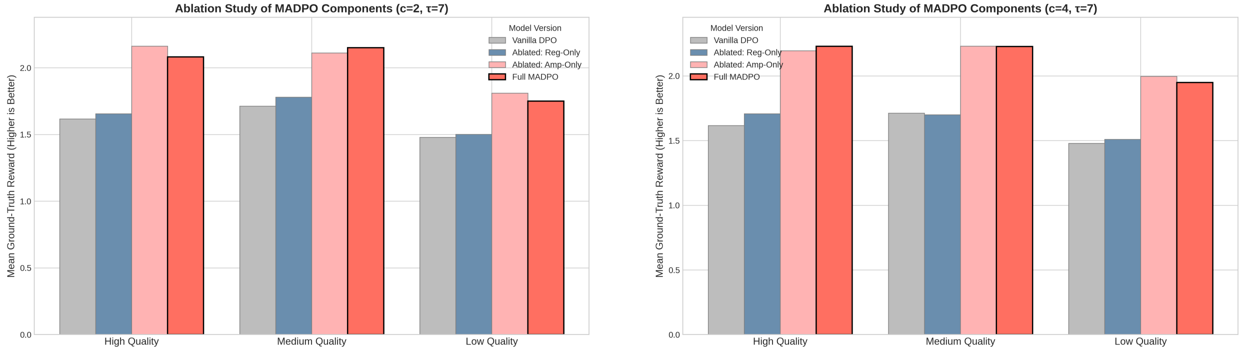
The analysis for the threshold $\tau$ (Figure 2a) reveals a nuanced interaction between the threshold and the underlying data quality, highlighting the importance of both of MADPO's components. For the High Quality dataset, performance peaks at an optimal value ($\tau = 4$) before plateauing. This indicates that setting the threshold too high on clean data is suboptimal; it misclassifies easy-to-learn pairs as hard, applying unnecessary amplification where regularization would be more beneficial. This result underscores the value of the regularization component when data quality is high.

Conversely, for the Low and Medium quality datasets, performance consistently improves with a larger $\tau$. This suggests that when the data is noisy, a higher threshold is advantageous as it ensures the truly informative, low-margin pairs are amplified. The large gradients from these amplified pairs then dominate the training update, providing an implicit regularization effect that prevents the model from overfitting to the uninformative, high-margin pairs present in lower-quality data. Therefore, the optimal setting for $\tau$ is dependent on the expected data quality, with noisier datasets benefiting from a more aggressive and wide-ranging amplification strategy.

The sensitivity to the amplification intensity $c$ (Figure 2b), where we set $c_{\max} = c$ and $c_{\min} = 1/c$, shows a more uniform trend. Across all three data quality tiers, a higher intensity leads to stronger performance. This robustly demonstrates the effectiveness of our core mechanism: aggressively amplifying the learning signal for informative pairs while simultaneously and strongly dampening it for uninformative ones is a beneficial strategy regardless of the overall data quality. Overall, this analysis confirms that MADPO is not overly sensitive to its hyperparameter choices and that clear trends can guide practitioners toward an optimal configuration.

**Ablation Study** The results of our ablation study, presented in Figure 3, reveal that the amplification mechanism is the primary driver of MADPO's performance gains.

To isolate the two core components of our method, we analyze two ablated models. The first is an Amplification-Only version, where the weight is set to one for high-margin pairs ($|h_{\hat{\phi}}| \geq \tau$) to disable the regularization



**(a) Ablation Study with** $c = 2, \tau = 7$

**(b) Ablation Study with** $c = 4, \tau = 7$

Figure 3: Ablation study of MADPO's amplification and regularization components. The study compares the full MADPO model against vanilla DPO and two ablated versions: Amp-Only, which only amplifies low-margin pairs (by setting $w(h_{\hat{\phi}}) = 1$ for $|h_{\hat{\phi}}| \geq \tau$), and Reg-Only, which only regularizes high-margin pairs (by setting $w(h_{\hat{\phi}}) = 1$ for $|h_{\hat{\phi}}| < \tau$). The comparison is shown under two hyperparameter settings: **(Left)** a moderate amplification intensity ($c = 2, \tau = 7$), and **(Right)** a high amplification intensity ($c = 4, \tau = 7$).

component. This model consistently achieves the highest or near-highest performance across all settings, suggesting that aggressively learning from informative, low-margin pairs is the most critical factor for success. One interpretation is that strong amplification also serves as a form of implicit regularization; by forcing the optimization to prioritize hard examples, the gradients from easy examples have less influence on the overall update, which may prevent overfitting.

The second model is a Regularization-Only version, where the weight is set to one for low-margin pairs ($|h_{\hat{\phi}}| < \tau$) to disable amplification. The results for this model show that when tuned properly, explicitly dampening high-margin pairs provides a consistent improvement over the vanilla DPO baseline across all data quality tiers. This confirms that the regularization component is a beneficial mechanism in its own right, even if its impact is secondary to that of amplification.

While the Amplification-Only model performs exceptionally well, this does not render the explicit regularization component meaningless. The full MADPO model, which includes both mechanisms, represents a more complete and theoretically robust solution designed for general-purpose application. The regularization component provides a crucial safeguard against overfitting on easy examples, a problem which may be more or less severe depending on the specific dataset and base model. Our results show that the full model's performance is highly competitive with the 'Amplification-Only' version, indicating that the regularization term offers this theoretical robustness without a significant performance trade-off in practice.

## 6 Conclusion

In this work, we addressed a key limitation in the vanilla Direct Preference Optimization (DPO) framework: its reliance on a single, fixed temperature parameter that struggles with preference data of varying quality. We introduced Margin-Adaptive Direct Preference Optimization (MADPO), a method that applies an instance-level, adaptive weight to the DPO loss. Our theoretical analysis proved that MADPO is a principled modification that maintains a stable optimization landscape and is robust to the errors inherent in practical two-step training pipelines. Our empirical results on a sentiment generation task confirmed these findings, demonstrating that MADPO consistently outperforms strong baselines, particularly on challenging, low-quality datasets. Our analysis revealed that this success is primarily driven by MADPO's ability to amplify the learning signal for informative, low-margin examples.

We acknowledge two primary limitations in our current study. First, our experiments were conducted on a 270M-parameter language model. While this allowed for a controlled and thorough analysis of the method's mechanics, further research is needed to validate whether our findings generalize to larger, state-of-the-art models. Second, our work relies on a synthetic preference dataset generated from an oracle reward model. This provided a clean, controlled environment for analysis, but the dynamics of training on real-world, human-annotated preference data—which can be inherently noisier and more inconsistent—remain an important area for future investigation.

## References

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22 (98):1–76, 2021.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

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

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust dpo: Aligning language models with noisy feedback. *arXiv preprint arXiv:2403.00409*, 2024.

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

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 251–260. Association for Computational Linguistics, 2022. doi:10.18653/v1/2022.acl-demo.25.

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

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, 2011.

Daniel McFadden. Economic choices. *American economic review*, 91(3):351–378, 2001.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

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

Shangpin Peng, Weinong Wang, Zhuotao Tian, Senqiao Yang, Xing Wu, Haotian Xu, Chengquan Zhang, Takashi Isobe, Baotian Hu, and Min Zhang. Omni-dpo: A dual-perspective paradigm for dynamic preference learning of llms. *arXiv preprint arXiv:2506.10054*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 53728–53741, 2023.

Gemma Team. Gemma 3. 2025. URL `https://arxiv.org/abs/2503.19786`.

Junkang Wu, Xue Wang, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. $\alpha$-dpo: Adaptive reward margin is what direct preference optimization needs. *arXiv preprint arXiv:2410.10148*, 2024a.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. $\beta$-dpo: Direct preference optimization with dynamic $\beta$. *Advances in Neural Information Processing Systems*, 37:129944–129966, 2024b.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. Wpo: Enhancing rlhf with weighted preference optimization. *arXiv preprint arXiv:2406.11827*, 2024.

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

# A Proofs

*Proof of Proposition 4.1.* For mathematical representation in this proof, we adopt the notation $(x, y, y', d) \in \mathcal{D}$, where $y, y'$ are responses and $d \in \{0, 1\}$ is a binary preference indicator such that $d = 1$ indicates $y \succ y'$ (i.e., $y = y_w$, $y' = y_l$) and $d = 0$ indicates $y' \succ y$ (i.e., $y' = y_w$, $y = y_l$). The introduction of $d$ allows us to flexibly model preference directions using the BTL model, where $E_{d\sim P_{\phi^*}}[d|x, y, y'] = \sigma(h_{\phi^*})$. We now analyze the expected loss for any triplet $(x, y, y', d) \in \mathcal{D}_{\text{low}}$. The analysis begins by taking the expectation of the sample-level loss over the optimal preference distribution $d \sim P_{\phi^*}$. This expectation simplifies to a new cross-entropy objective.

The derivation proceeds as follows:

$$\begin{aligned}\mathcal{L}(\pi_\theta, \phi^*; x, y, y') &= -\mathbb{E}_{d\sim P_{\phi^*}}\left[d \cdot w(h_{\phi^*}) \log \sigma(\beta h_\theta) + (1 - d) \cdot w(-h_{\phi^*}) \log \sigma(-\beta h_\theta)|x, y, y'\right] \\ &= -\left[\sigma(h_{\phi^*}) w(h_{\phi^*}) \log \sigma(\beta h_\theta) + \sigma(-h_{\phi^*}) w(-h_{\phi^*}) \log \sigma(-\beta h_\theta)\right] \\ &= -\left[\sigma(c(|h_{\phi^*}|) \cdot h_{\phi^*}) \log \sigma(\beta h_\theta) + \sigma(-c(|h_{\phi^*}|) \cdot h_{\phi^*}) \log \sigma(-\beta h_\theta)\right].\end{aligned}$$

The second equality holds by BTL where $E_{d\sim P_{\phi^*}}[d|x, y, y'] = \sigma(h_{\phi^*})$. The last equality is satisfied by the construction of $w(h)$. This final form is the cross-entropy loss between the policy's distribution, $P_\theta$, and a margin-aware target distribution, $P'_{\phi^*}$, with logits defined by $c(|h_{\phi^*}|) \cdot h_{\phi^*}$. Since the total loss, $\mathcal{L}(\theta, \phi^*)$, is the expectation of these non-negative, instance-level cross-entropy losses over the dataset, the global minimum is achieved when the loss for each instance is minimized simultaneously. This occurs if and only if the distributions are identical for each instance, which requires their logits to be equal. Therefore, the optimal solution satisfies:

$$\beta h_{\theta^*} = c(|h_{\phi^*}|) \cdot h_{\phi^*}.$$

□

*Proof of Proposition 4.2.* As we did for the proof of Proposition 4.1, we adopt the notation $(x, y, y', d) \in \mathcal{D}$. Without loss of generality, we assume $h_{\phi^*} \equiv h_{\phi^*}(x, y, y') = r_{\phi^*}(x, y) - r_{\phi^*}(x, y') > 0$ by swapping $y$ and $y'$ if necessary. So the weighting function is

$$w(h_{\phi^*}) = \frac{\sigma(c(h_{\phi^*}) \cdot h_{\phi^*})}{\sigma(h_{\phi^*})} \quad \text{and} \quad w(-h_{\phi^*}) = 1$$

We derive the expected loss for a triplet $(x, y, y', d) \in \mathcal{D}_{\text{high}}$ with $E_{d\sim P_{\phi^*}}[d|x, y, y'] = \sigma(h_{\phi^*})$ from BTL model:

$$\mathcal{L}(\pi_\theta, \phi^*; x, y, y') = -\left[\sigma(c(|h_{\phi^*}|) \cdot h_{\phi^*}) \log \sigma(\beta h_\theta) + \sigma(-h_{\phi^*}) \log \sigma(-\beta h_\theta)\right]. \tag{6}$$

Now, we take derivatives of Eq. equation 6 with respect to $\beta h_\theta$, which results in

$$\frac{\partial \mathcal{L}(\pi_\theta, \phi^*; x, y, y')}{\partial \beta h_\theta} = -\left[\sigma(c(|h_{\phi^*}|) \cdot h_{\phi^*}) \sigma(-\beta h_\theta) - \sigma(-h_{\phi^*}) \sigma(\beta h_\theta)\right].$$

At the optimum $\theta^*$, we have

$$F(c, \beta h_{\theta^*}) \equiv \left.\frac{\partial \mathcal{L}(\pi_\theta, \phi^*; x, y, y')}{\partial \beta h_\theta}\right|_{\theta=\theta^*} = -\left[\sigma(c(|h_{\phi^*}|) \cdot h_{\phi^*}) \sigma(-\beta h_{\theta^*}) - \sigma(-h_{\phi^*}) \sigma(\beta h_{\theta^*})\right] = 0.$$

We denote $c \equiv c(|h_{\phi^*}|)$ for notation simplicity. By the implicit function theorem, $\frac{\partial \beta h_{\theta^*}}{\partial c} = -\frac{\partial F/\partial c}{\partial F/\partial \beta h_{\theta^*}}$. Compute the partial derivatives:

$$\begin{aligned}\frac{\partial F}{\partial c} &= \frac{\partial}{\partial c}\left[\sigma(c \cdot h_{\phi^*}) \sigma(-\beta h_{\theta^*}) - \sigma(-h_{\phi^*}) \sigma(\beta h_{\theta^*})\right] \\ &= h_{\phi^*} \sigma(c \cdot h_{\phi^*}) \sigma(-c \cdot h_{\phi^*}) \sigma(-\beta h_{\theta^*}). \\ \frac{\partial F}{\partial \beta h_{\theta^*}} &= \sigma(c \cdot h_{\phi^*}) \cdot \left[-\sigma(\beta h_{\theta^*}) \sigma(-\beta h_{\theta^*})\right] - \sigma(-h_{\phi^*}) \cdot \left[\sigma(\beta h_{\theta^*}) \sigma(-\beta h_{\theta^*})\right] \\ &= -\sigma(\beta h_{\theta^*}) \sigma(-\beta h_{\theta^*}) \left[\sigma(c \cdot h_{\phi^*}) + \sigma(-h_{\phi^*})\right].\end{aligned}$$

Thus:

$$\frac{\partial \beta h_{\theta^*}}{\partial c} = -\frac{h_{\phi^*}\sigma(c \cdot h_{\phi^*})\sigma(-c \cdot h_{\phi^*})\sigma(-\beta h_{\theta^*})}{-\sigma(\beta h_{\theta^*})\sigma(-\beta h_{\theta^*})\left[\sigma(c \cdot h_{\phi^*}) + \sigma(-h_{\phi^*})\right]} = \frac{h_{\phi^*}\sigma(c \cdot h_{\phi^*})\sigma(-c \cdot h_{\phi^*})\sigma(-\beta h_{\theta^*})}{\sigma(\beta h_{\theta^*})\sigma(-\beta h_{\theta^*})\left[\sigma(c \cdot h_{\phi^*}) + \sigma(-h_{\phi^*})\right]}.$$

Since $h_{\phi^*} > 0$, and all sigmoid terms are strictly positive, the numerator and denominator are positive, so $\frac{\partial \beta h_{\theta^*}}{\partial c} > 0$. Thus, $\frac{\partial(\beta h_{\theta^*})}{\partial c} > 0$, proving the proposition. □

**Lemma 1.** *Let $w(h)$ be the weight function defined in Eq. 4. The function is differentiable almost everywhere, and its derivative, $w'(h)$, is uniformly bounded. That is, there exists a constant $L_w > 0$ such that for all $h \in \mathbb{R}$ where the derivative is defined:*

$$|w'(h)| \leq L_w.$$

*Proof.* For the purpose of this analysis, we can simplify the expression by fixing the hyperparameters to representative values. Hyperparameters' specific value does not affect the following stability analysis, so for simplicity we let $c_{\min} = 0$, along with $c_{\max} = 2$ and $\lambda = 1$.

By construction, the weight function $w(h)$ is continuous everywhere and differentiable almost everywhere. The only non differentiable points occur at $h = 0$ due to the absolute value in the coefficient function $c$ and at $h = -\tau$ due to the piecewise definition.

- **Case 1: For $h \in (0, \infty)$.**
  Let $k(h) = c(h) \cdot h$. The weight function and its derivative are:

  $$w(h) = \frac{\sigma(k(h))}{\sigma(h)}, \quad w'(h) = \frac{\sigma'(k(h))k'(h)\sigma(h) - \sigma(k(h))\sigma'(h)}{\sigma^2(h)}$$

  The sigmoid function $\sigma(\cdot)$ and its derivative $\sigma'(\cdot)$ are universally bounded (by 1 and 1/4, respectively). For $h > 0$, the denominator $\sigma^2(h)$ is bounded away from zero. Therefore, to show that $w'(h)$ is bounded, we only need to show that $k'(h)$ is bounded.

  By the product rule, $k'(h) = c(h) + c'(h)h$. By definition, $c(h)$ is bounded between $[c_{\min}, c_{\max}]$. The term $c'(h)h$ involves the derivative of the coefficient function, which contains an exponential decay term that dominates the linear growth of $h$. As $h \to \infty$, $c'(h)h \to 0$. Since $k'(h)$ is continuous on $(0, \infty)$ and has finite limits at its boundaries ($h \to 0^+$ and $h \to \infty$), it is bounded on this interval. Thus, $w'(h)$ is also bounded for $h > 0$.

- **Case 2: For $h \in (-\tau, 0)$.**
  On this bounded open interval, the derivative $w'(h)$ is a continuous function. As established in our analysis of the boundaries, the one-sided limits of $w'(h)$ as $h \to -\tau^+$ and as $h \to 0^-$ are both finite. A function that is continuous on a bounded open interval and has finite limits at its endpoints is necessarily bounded. Thus, $w'(h)$ is bounded on this interval.

- **Case 3: For $h < -\tau$.**
  In this region, $w(h) = 1$. Therefore, its derivative is $w'(h) = 0$, which is trivially bounded.

Since the derivative $w'(h)$ is bounded on all three regions that cover its domain, we conclude that it is uniformly bounded. □

**Lemma 2.** *The weight function $w$ defined in Eq. 4 is $L_w$-Lipschitz continuous. That is, for any $h, h' \in \mathbb{R}$, the following inequality holds:*

$$|w(h) - w(h')| \leq L_w|h - h'|. \tag{7}$$

*Proof.* The proof relies on Lemma 1, which establishes that the derivative of the weight function is bounded almost everywhere, i.e., $|w'(t)| \leq L_w$. A function with a bounded derivative (a.e.) is absolutely continuous, which is the required condition to apply the Fundamental Theorem of Calculus for Lebesgue Integrals.

For any two points $a, b \in \mathbb{R}$ with $a < b$, the theorem states:

$$w(b) - w(a) = \int_a^b w'(t)dt. \tag{8}$$

We can now take the absolute value of both sides and apply the bound from our lemma:

$$\begin{aligned}
|w(b) - w(a)| &= \left| \int_a^b w'(t)dt \right| \\
&\leq \int_a^b |w'(t)|dt && \text{(Triangle inequality for integrals)} \\
&\leq \int_a^b L_w dt && \text{(By Lemma 1, } |w'(t)| \leq L_w\text{)} \\
&= L_w(b-a).
\end{aligned}$$

Since this holds for any $a < b$, we can generalize to $|w(h) - w(h')| \leq L_w|h - h'|$ for any $h, h' \in \mathbb{R}$. This is the definition of $L_w$-Lipschitz continuity. □

*Proof of Theorem 4.6.* The proof proceeds in two main parts. First, we establish that the margin function $h_\phi$ is Lipschitz continuous with respect to its parameters $\phi$. Second, we leverage this result to show that the full loss function, $\mathcal{L}(\theta, \phi; x, y_w, y_l)$, is also Lipschitz continuous, which allows us to invoke the final result from prior work.

**Part 1: Lipschitz Continuity of the Margin Function ($h_\phi$).** We begin with the definition of the margin function: $h_\phi(x, y_w, y_l) = r_\phi(x, y_w) - r_\phi(x, y_l)$. By the Mean Value Theorem, there exists a $\bar{\phi}$ on the line segment between $\phi^*$ and $\hat{\phi}$ such that:

$$h_{\hat{\phi}}(x, y_w, y_l) - h_{\phi^*}(x, y_w, y_l) = \left(\nabla_\phi r_{\bar{\phi}}(x, y_w) - \nabla_\phi r_{\bar{\phi}}(x, y_l)\right)^\top (\hat{\phi} - \phi^*).$$

Taking the absolute value and applying the generalized Cauchy-Schwarz inequality with the semi-norm $\|\cdot\|_{\Sigma_\phi + \kappa I}$ gives:

$$|h_{\hat{\phi}} - h_{\phi^*}| \leq \left\|\nabla_\phi r_{\bar{\phi}}(x, y_w) - \nabla_\phi r_{\bar{\phi}}(x, y_l)\right\|_{(\Sigma_\phi + \kappa I)^{-1}} \cdot \|\hat{\phi} - \phi^*\|_{\Sigma_\phi + \kappa I}.$$

Under Assumption 4.4, the gradient of the reward function is bounded. This implies that the term involving the gradients is also bounded by some constant, which we will call $L_\phi$. Thus, the margin function is $L_\phi$-Lipschitz continuous with respect to $\phi$:

$$|h_{\hat{\phi}}(x, y_w, y_l) - h_{\phi^*}(x, y_w, y_l)| \leq L_\phi \|\hat{\phi} - \phi^*\|_{\Sigma_\phi + \kappa I}.$$

**Part 2: Lipschitz Continuity of the Full Loss Function ($\mathcal{L}$).** Now we analyze the full loss function, $\mathcal{L}(\theta, \phi; x, y_w, y_l) = -w(h_\phi(x, y_w, y_l)) \log \sigma(\beta h_\theta(x, y_w, y_l))$.

$$\begin{aligned}
\left|\mathcal{L}(\theta, \phi^*; x, y_w, y_l) - \mathcal{L}(\theta, \hat{\phi}; x, y_w, y_l)\right| &= |-\log \sigma(\beta h_\theta(x, y_w, y_l)| \cdot \left|w(h_{\phi^*}(x, y_w, y_l)) - w(h_{\hat{\phi}}(x, y_w, y_l))\right| \\
&\leq L_\theta \cdot \left|w(h_{\phi^*}(x, y_w, y_l)) - w(h_{\hat{\phi}}(x, y_w, y_l))\right| \\
&\leq L_\theta \cdot L_w \cdot |h_{\phi^*}(x, y_w, y_l) - h_{\hat{\phi}}(x, y_w, y_l)| \\
&\leq L_\theta \cdot L_w \cdot L_\phi \cdot \|\hat{\phi} - \phi^*\|_{\Sigma_\phi + \kappa I}.
\end{aligned}$$

The second inequality holds by Assumption 4.5. The third inequality holds by Lemma 2. The last inequality holds by Part 1 of this proof. By defining the final Lipschitz constant as $L = L_\theta L_w L_\phi$, we have shown that our loss function is L-Lipschitz continuous:

$$\left|\mathcal{L}(\theta, \phi^*; x, y_w, y_l) - \mathcal{L}(\theta, \hat{\phi}; x, y_w, y_l)\right| \leq L \cdot \|\hat{\phi} - \phi^*\|_{\Sigma_\phi + \kappa I}.$$

With this result established, the final statistical bound on the estimation error follows directly by invoking the general framework for two-step estimators, such as Theorem 4.2 in Chowdhury et al. (2024). This completes the proof. □

*Proof of Proposition 4.7.* For readability in this proof, we suppress the explicit dependence on the sample $(x, y_w, y_l)$ and parameters $(\theta, \phi)$ for functions like $\mathcal{L}$, $h_\theta$, and $h_\phi$, unless required for clarity. The proof for both claims relies on the fact that the weight function, $w(h_\phi)$, is uniformly bounded. From its construction in Eq. 4, there exists a constant $w_{\max}$ such that $|w(h_\phi)| \leq w_{\max}$ for all inputs.

1. **Bounded Gradient:** The first derivative of the loss is given by:

$$\frac{\partial \mathcal{L}}{\partial h_\theta} = -w(h_\phi) \cdot \beta\sigma(-\beta h_\theta).$$

   This expression is a product of the weight function, a constant $\beta$, and the sigmoid function, all of which are bounded. Therefore, their product is uniformly bounded.

2. **Bounded Hessian:** The second derivative of the loss is given by:

$$\frac{\partial^2 \mathcal{L}}{\partial h_\theta^2} = w(h_\phi) \cdot \beta^2\sigma(-\beta h_\theta)\sigma(\beta h_\theta).$$

   This is a product of the bounded weight function, a constant $\beta^2$, and the derivative of the sigmoid function, $\sigma'(\cdot) = \sigma(\cdot)\sigma(-\cdot)$, which is famously bounded by 1/4. Therefore, the second derivative is also uniformly bounded.

□

## B Related Work

Recent research in preference alignment has sought to address the limitations of DPO's fixed temperature, which can lead to overfitting on easy, high-margin pairs. Prominent approaches such as IPO (Azar et al., 2024) and $\beta$-DPO (Wu et al., 2024b) tackle this problem with mechanisms that are applied at a coarse granularity. IPO proposes a uniform target margin for all samples, while $\beta$-DPO's strategies operate at the batch level.

While an improvement, the batch-level mechanisms of $\beta$-DPO have several notable drawbacks. First, its adaptation is a coarse approximation of the instance-level ideal. A single training batch can easily contain a mix of high- and low-margin pairs, yet the $\beta$-batch method applies a single, compromised temperature to all of them. Second, the linear adaptation rule, $\beta_{\text{batch}} = \beta_0(1 + m \cdot (\bar{h}_\theta - h_0))$, can be unstable; for difficult batches where the average margin $\bar{h}_\theta$ is negative, the resulting temperature $\beta_{\text{batch}}$ can also become negative, leading to a divergent objective that actively learns to prefer the dispreferred response. Finally, the batch-dependent temperature complicates hyperparameter selection, as the absence of a fixed $\beta$ makes it difficult to perform reliable cross-validation to find the optimal tuning parameters.

Our work, MADPO, is distinct in that it provides a fully instance-level and data-preserving solution that avoids these issues. By applying a continuous, adaptive weight to each training sample based on its unique reward margin, our method can granularly control the learning signal. This allows it to be aggressive on hard pairs and conservative on easy ones within the same batch, providing a more flexible and stable approach to preference alignment.

Beyond the methods discussed above, other recent works have extended the DPO framework in various directions. For instance, methods like SimPO (Meng et al., 2024) and $\alpha$-DPO (Wu et al., 2024a) focus on simplifying the objective by removing the need for an explicit reference policy. Another line of work has explored reweighting preference data. Omni-DPO (Peng et al., 2025) dynamically weights pairs based on both their inherent quality and the model's current learning state, while WPO (Zhou et al., 2024) reweights off-policy preference data to more closely resemble the on-policy distribution. While these methods also use a reweighting scheme, their motivation is distinct from that of MADPO. Whereas these approaches weight data based on the policy's dynamic state or distributional properties, MADPO's weighting is based on a static, external signal of sample difficulty derived from the reward margin, $h_\phi$. Our goal is not to correct for distributional shift, but to granularly control the regularization strength for each individual sample based on its intrinsic difficulty.