# LCPO: TOWARDS ADDRESSING REWARD DEGENERATION IN DIRECT PREFERENCE OPTIMIZATION

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

## ABSTRACT

Direct Preference Optimization (DPO) and its variants are widely used for preference alignment in large language models. However, prior work has shown that DPO often suffers from reward degeneration during training, where the implicit rewards of both chosen and rejected responses drift into extremely negative regions. In this paper, we analyze this phenomenon through the lens of reward margin and reward level. We show that DPO enlarges the reward margin but provides no lower-bound control along the reward-level direction. Consequently, once optimization enters a low-level region, vanilla DPO has no intrinsic mechanism to restore the reward level. We further show that pairwise preference data do not identify the absolute reward value, so correcting reward degeneration necessarily requires introducing additional priors. This motivates us to formulate degeneration correction as a prior design problem. Under this view, we propose three design principles for such priors: level-active, target-preserving, and calibration-free. Guided by these principles, we introduce Level-Constrained Preference Optimization (LCPO), a simple and general solution that imposes a relational constraint between chosen and rejected rewards to counteract reward-level collapse. Empirical results on representative benchmarks show that LCPO effectively mitigates reward degeneration and consistently outperforms existing baselines. Code is provided in the supplementary material.

## 1 INTRODUCTION

Aligning large language models (LLMs) with human preferences is a central goal of modern language model training (Ziegler et al., 2019; Wang et al., 2023; Stiennon et al., 2020; Achiam et al., 2023). While Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017) has long been the dominant paradigm, Direct Preference Optimization (DPO) (Rafailov et al., 2023) and its variants have recently emerged as attractive alternatives due to their simplicity and strong empirical performance. By avoiding explicit reward-model training and reinforcement learning, DPO has been widely adopted across a range of preference alignment settings. However, as DPO is increasingly used in practice, a recurring training pathology has also become apparent: although the preference gap between chosen and rejected responses continues to widen, the implicit rewards of both responses often drift downward simultaneously, sometimes reaching extremely negative values. In this paper, we refer to this phenomenon—where the reward margin increases while the overall reward level collapses—as *reward degeneration*.

Reward degeneration is not merely an isolated empirical anomaly. A growing body of work has examined this issue from different perspectives and proposed diverse remedies, including likelihood constraints (Xu et al., 2024; Pang et al., 2024), zero-point anchoring (Pal et al., 2024; D'Oosterlinck et al., 2025), calibrated objectives (Azar et al., 2024; Xiao et al., 2024), and other auxiliary regularizations to mitigate downward reward drift during training. However, these methods also reflect substantially different assumptions about how reward degeneration should be corrected. This raises a fundamental question that remains largely unexplored: what kind of correction is actually consistent with the structural properties of reward degeneration under pairwise preference learning?

To address this question, we revisit preference optimization through the lens of *reward margin* and *reward level*. The former captures the relative preference gap between chosen and rejected responses, while the latter characterizes the overall vertical placement of their implicit rewards. Under



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

this decomposition, DPO explicitly drives margin growth but exerts only a one-sided influence on reward level: as the margin increases, the feasible upper bound of the reward level is progressively compressed, yet the objective provides no corresponding lower-bound protection. This asymmetry reveals a core limitation of vanilla DPO: once training enters a low-level region, the objective has no intrinsic mechanism to restore the reward level. We further show that the absolute reward value is non-identifiable from pairwise preference data: such data constrain only the relative difference between chosen and rejected responses, but do not determine the absolute offset of the reward function. As a result, no correction for reward degeneration can be uniquely inferred from the data alone; any such correction must inevitably introduce additional priors. This suggests that correcting reward degeneration is fundamentally a prior design problem.

Based on these observations, we propose three key design principles for such priors: (1) *level-active*, meaning that the correction should provide explicit pressure along the reward-level direction; (2) *target-preserving*, meaning that it should mitigate degeneration without redefining the original DPO objective; and (3) *calibration-free*, meaning that it should avoid tying the non-identifiable absolute reward value to an externally specified baseline. Guided by these principles, we introduce *Level-Constrained Preference Optimization* (LCPO), which adds a relational constraint between the implicit rewards of chosen and rejected responses to correct reward-level collapse without imposing externally calibrated absolute targets. We systematically evaluate LCPO across representative preference alignment benchmarks, including instruction-following (Cui et al., 2023) and summarization (Völske et al., 2017), under diverse model architectures and scales. Experimental results show that LCPO not only substantially mitigates reward degeneration, but also consistently delivers robust performance gains over strong baselines.

Our contributions are threefold: (i) we provide a margin-level analysis of DPO and identify a structural asymmetry in reward-level control; (ii) we show that degeneration correction is fundamentally a prior design problem and derive three design principles; and (iii) we introduce LCPO and show that it substantially mitigates reward degeneration while improving downstream alignment performance.

## 2 Related Work

Preference alignment has traditionally been dominated by RLHF (Ziegler et al., 2019; Bai et al., 2022; Ouyang et al., 2022), which typically follows a two-stage pipeline: a reward model is first learned from preference data, and the policy is then optimized with reinforcement learning algorithms such as PPO (Schulman et al., 2017). DPO (Rafailov et al., 2023) recasts this process into a closed-form preference optimization objective, avoiding explicit reward-model training and reinforcement learning. Since its introduction, a large body of work has extended DPO to address issues such as data bias (Meng et al., 2024; Park et al., 2024; Chowdhury et al., 2024), hyperparameter sensitivity (Lee et al., 2026; Chai, 2026), and theoretical generalizations (Chen et al., 2024; Ethayarajh et al., 2024; Zeng et al., 2024). Among these developments, a recurring failure mode has been widely observed: the implicit rewards of both chosen and rejected responses may drift into extremely negative regions during training, often harming optimization stability and alignment performance (Feng et al., 2024; Pang et al., 2024; Chen et al., 2026; Yuan et al., 2025; Razin et al., 2025; Yang et al., 2025). This issue has been discussed under various names in prior work, including *likelihood displacement* (Razin et al., 2025), *squeezing effect* (Ren & Sutherland, 2025), *response suppression* (Chen et al., 2026), and *reward scale mismatch* (Xiao et al., 2024). For clarity and consistency, we refer to this phenomenon throughout the paper as *reward degeneration*.

Some prior studies (Yuan et al., 2025; Razin et al., 2025; Feng et al., 2024) have attempted to explain the causes of reward degeneration from the perspective of optimization dynamics. A larger body of work, however, has focused on the practical question of how to mitigate this issue during training. For example, RPO (Pang et al., 2024) and CPO (Xu et al., 2024) introduce negative log-likelihood (NLL) regularization on the chosen responses; DPOP (Pal et al., 2024) and APO-zero (D'Oosterlinck et al., 2025) penalize or anchor implicit rewards relative to a zero baseline; and IPO (Azar et al., 2024) and Cal-DPO (Xiao et al., 2024) constrain rewards or margins around calibrated target values. These methods differ substantially in how they regularize the reward landscape, reflecting different implicit assumptions about how degeneration should be corrected. In contrast, our work analyzes preference optimization through the lens of reward margin and reward level, and frames degeneration correction as a prior design problem under pairwise preference learning.

# 3 METHOD

## 3.1 PRELIMINARIES

The standard RLHF objective (Jaques et al., 2017; Rafailov et al., 2023) is formulated as

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

where $x$ and $y$ denote the prompt and response, respectively. Here, $\pi_\theta$ denotes the policy model to be aligned, $\pi_{\mathrm{ref}}$ the reference model, $r(x,y)$ the ground-truth reward function reflecting human preferences, and $\beta > 0$ a hyperparameter controlling the KL regularization strength.

Following prior work (Korbak et al., 2022; Go et al., 2023), the optimal policy $\pi^*$ under Equation 1 satisfies

$$\pi^*(y\mid x) = \frac{1}{Z(x)}\,\pi_{\mathrm{ref}}(y\mid x)\exp\left(\frac{r(x,y)}{\beta}\right), \tag{2}$$

where $Z(x) = \sum_y \pi_{\mathrm{ref}}(y\mid x)\exp\big(r(x,y)/\beta\big)$ is the partition function.

Given a prompt $x$ and a pair of responses $(y_w, y_l)$, human preference is typically parameterized via the Bradley-Terry (BT) model (Bradley & Terry, 1952):

$$p(y_w \succ y_l \mid x) = \sigma\big(r(x,y_w) - r(x,y_l)\big), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function. By re-expressing $r(x,y)$ from Equation 2 and substituting it into the BT model, the partition function $Z(x)$ cancels out. Consequently, DPO (Rafailov et al., 2023) reparameterizes the reward with an implicit reward model $r_\theta$:

$$r_\theta(x,y) = \beta\log\frac{\pi_\theta(y\mid x)}{\pi_{\mathrm{ref}}(y\mid x)}. \tag{4}$$

Optimizing the maximum likelihood objective on the preference dataset $\mathcal{D}$ yields the DPO loss:

$$\mathcal{L}_{\mathrm{DPO}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\big(r_\theta(x,y_w) - r_\theta(x,y_l)\big)\right]. \tag{5}$$

In this work, we decouple the implicit reward pair into the reward margin $u_\theta$ and the reward level $v_\theta$, defined respectively as

$$u_\theta(x,y_w,y_l) = r_\theta(x,y_w) - r_\theta(x,y_l), \qquad v_\theta(x,y_w,y_l) = r_\theta(x,y_w) + r_\theta(x,y_l). \tag{6}$$

Under this formulation, the DPO loss can be expressed as

$$\mathcal{L}_{\mathrm{DPO}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\big(u_\theta(x,y_w,y_l)\big)\right]. \tag{7}$$

## 3.2 MARGIN-LEVEL ANALYSIS OF DPO

As established in Equation 7, the DPO loss depends only on the reward margin $u_\theta$ and places no explicit constraint on the reward level $v_\theta$. Nevertheless, because the implicit rewards are induced by normalized policy probabilities, the margin and level are structurally coupled through the following upper bound.

**Proposition 3.1.** *Let $u_\theta$ and $v_\theta$ be defined as in Equation 6. Then, for any preference pair $(x, y_w, y_l)$,*

$$v_\theta \le -u_\theta + C(x,y_w), \tag{8}$$

*where $C(x,y_w) = -2\beta\log\pi_{\mathrm{ref}}(y_w\mid x)$ is a constant independent of $\theta$.*

The proof is provided in Appendix A. Proposition 3.1 shows that increasing the margin $u_\theta$ linearly lowers an upper bound on the feasible reward level $v_\theta$. This observation highlights a one-sided asymmetry in how DPO interacts with reward level:

- **Upper-bound restriction:** as the margin increases, the feasible upper bound of the reward level is progressively compressed.
- **No explicit lower-bound control:** the DPO objective contains no corresponding lower bound or direct level-wise term that penalizes the downward drift of $v_\theta$.

Importantly, this is an objective-level and geometry-level observation rather than a complete characterization of finite-step optimization dynamics under neural parameterization. Still, it suggests a structural vulnerability of vanilla DPO to reward degeneration: although DPO explicitly encourages margin growth, it does not directly distinguish between optimization states that achieve a similar reward margin at different absolute reward values. In particular, if training moves into a low-level region, the DPO objective by itself does not provide an explicit level-targeting signal that prefers a higher reward level over a lower one when the margin is held fixed. In this sense, standard DPO does not explicitly regulate or restore the reward level.

A natural question is then how the reward level should be stabilized or corrected once such degeneration occurs. As we show next, pairwise preference data alone do not identify the absolute offset of the reward function. This means that any method that explicitly selects, stabilizes, or restores the reward level beyond relative differences must rely on additional modeling assumptions or inductive biases beyond pairwise comparisons alone. By substituting Equation 2 into Equation 4, the optimal implicit reward satisfies

$$r^*_\theta(x, y) = r(x, y) - \beta \log Z(x). \tag{9}$$

This expression shows that even under the optimal policy, the implicit reward differs from the underlying reward by a prompt-dependent shift term $\beta \log Z(x)$. Crucially, because pairwise preferences depend only on reward differences, the partition function $Z(x)$ cancels out during training and cannot be identified from pairwise preference data alone.

Therefore, from the perspective of pairwise preference supervision, correcting reward degeneration is underdetermined with respect to the absolute reward value. Pairwise data constrain relative reward differences, but do not determine the absolute placement of rewards along the level direction. As a result, different methods for preventing or alleviating reward degeneration necessarily encode different assumptions, biases, or priors over this non-identifiable degree of freedom. Under this view, the diversity of existing remedies can be understood as reflecting different prior choices for handling the reward level.

### 3.3 Principles for Prior Design

The previous analysis suggests that correcting reward degeneration involves introducing additional assumptions or inductive biases over the non-identifiable reward-level direction. This motivates viewing degeneration correction as a prior design problem. If the goal is to mitigate reward degeneration while introducing as little additional structure as possible beyond what pairwise preference data support, we propose the following three design principles.

**Level-Active.** A correction prior should be *level-active*, meaning that it should provide an explicit optimization signal along the reward-level direction. Since the DPO objective depends only on the margin and is invariant to the level, it offers no direct corrective signal when the level drifts into an undesirable region. Therefore, a method that acts only on the margin may leave the level direction effectively unregulated. From this perspective, a well-matched correction should exert explicit level-wise pressure when the reward level becomes too low.

**Target-Preserving.** A correction prior should be *target-preserving*, meaning that it should correct reward degeneration without globally redefining the original preference optimization objective. If reward degeneration is viewed as a pathology of the optimization trajectory under an otherwise meaningful DPO objective, then it is natural to prefer corrections that intervene only when needed, rather than permanently replacing the target. This does not imply that objective-redefining methods are invalid; rather, target preservation is especially appealing when one aims to remain as close as possible to the original optimization intent of DPO outside degenerate regions.

**Calibration-Free.** A correction prior should also be *calibration-free*, meaning that it should avoid tying implicit rewards to externally specified absolute baselines or fixed target values. Under pairwise preference supervision, rewards are identifiable only up to a prompt-dependent additive shift, so the absolute reward value is not determined by the data alone. For this reason, calibration-free formulations provide a conservative design choice: they express correction through relations internal to the chosen–rejected pair, rather than through externally imposed absolute gauges. This does

Table 1: Comparison of representative methods for mitigating reward degeneration under the proposed prior design principles. This comparison is intended to characterize the type of inductive bias introduced by each method, rather than to serve as an absolute judgment of method quality.

| Method | Level-Active | Target-Preserving | Calibration-Free |
|---|---|---|---|
| DPO+NLL (Pang et al., 2024; Xu et al., 2024) | ✓ | ✗ | ✓ |
| DPOP (Pal et al., 2024) | ✓ | ✓ | ✗ |
| IPO (Azar et al., 2024) | ✗ | ✗ | ✓ |
| APO-zero (D'Oosterlinck et al., 2025) | ✓ | ✗ | ✗ |
| Cal-DPO (Xiao et al., 2024) | ✓ | ✗ | ✗ |

not mean that reward-calibrated methods cannot be effective in practice, but it clarifies the additional assumption they impose.

Based on these principles, we examine several representative methods for mitigating reward degeneration. As summarized in Table 1, existing remedies can be interpreted as making different prior choices, each reflecting a different trade-off under this design lens. We note that not satisfying one or more of these principles does not mean that a method is inherently unreasonable or ineffective. Because degeneration correction is underdetermined from pairwise preference data alone, different priors may be appropriate under different modeling assumptions or practical goals. In this work, these principles serve as a conservative and structurally grounded framework for reasoning about degeneration correction while keeping additional assumptions minimal. This perspective motivates us to propose a new method that satisfies all three principles.

### 3.4 Level-Constrained Preference Optimization

Guided by the three prior design principles above, we introduce *Level-Constrained Preference Optimization* (LCPO), a simple solution to reward degeneration derived from the preceding analysis. LCPO is not the only formulation compatible with these principles; rather, it serves as a minimal and geometrically transparent instantiation.

The central idea of LCPO is to complement vanilla DPO with an explicit constraint along the reward-level direction. In the margin-level view, DPO enlarges the reward margin but provides no lower-bound protection once the reward level drifts into a degenerate region. LCPO addresses this gap by introducing a feasibility boundary in the $(u, v)$ plane, yielding the following constrained preference optimization problem:

$$\begin{aligned} \min_{\theta} \quad & -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\big(u_\theta(x,y_w,y_l)\big)\right] \\ \text{s.t.} \quad & \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[v_\theta(x,y_w,y_l) + k\,u_\theta(x,y_w,y_l)\right] \geq 0. \end{aligned} \tag{10}$$

The geometric intuition of this linear boundary is straightforward: while the reward level $v$ is allowed to decrease as the margin $u$ grows, its downward drift is controlled relative to the achieved margin through the slope parameter $k$. This constraint directly targets reward-level collapse while still allowing continued margin growth.

Importantly, the admissible range of the slope parameter $k$ is not arbitrary; it is jointly constrained by the geometric structure of the feasible $(u, v)$ region and the prior design principles above. When $k = 1$, the constraint simplifies to $\mathbb{E}\big[r_\theta(x, y_w)\big] \geq 0$, which explicitly imposes a zero-valued target on the chosen implicit reward and therefore violates our calibration-free design principle. As further illustrated in Figure 1, when $k < 1$, the resulting boundary intersects the intrinsic probability-induced upper bound within the valid region $u > 0$, thereby imposing a hard upper bound on the achievable reward margin. Since preference optimization is intended to encourage margin growth rather than cap it, such a choice introduces an external restriction that contradicts the original optimization goal. Therefore, we require $k \in (1, +\infty)$.

Within this range, $k$ controls the strength of the restorative prior: a smaller $k$ imposes tighter protection against level collapse, whereas a larger $k$ yields a more relaxed intervention. In the asymptotic limit $k \to \infty$, the constraint reduces to $\mathbb{E}[u_\theta] \geq 0$, so the additional level-wise restriction vanishes and LCPO approaches vanilla DPO.

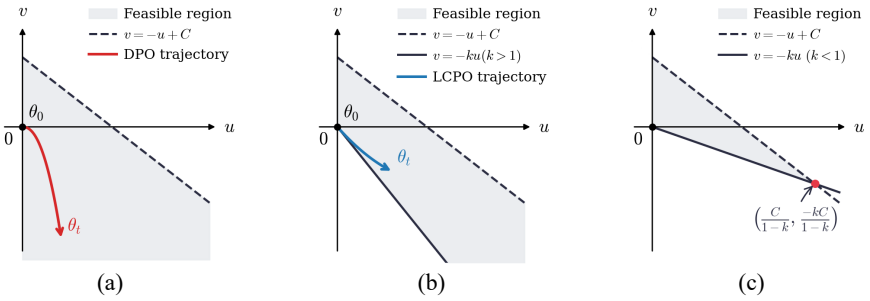



Figure 1: Geometric illustration of reward degeneration and level-constrained correction in the margin-level plane. **(a)** Under vanilla DPO, optimization can continue to increase the reward margin while the reward level drifts into a low-level region, reflecting the absence of lower-bound protection. **(b)** LCPO introduces a feasibility boundary that prevents further downward drift of the reward level while still allowing margin growth within the valid region. **(c)** When $k < 1$, the boundary intersects the intrinsic probability-induced upper bound in the valid region $u > 0$, showing that such a choice imposes an invalid external restriction on the feasible $(u, v)$ domain.

By substituting the definitions of $u_\theta$ and $v_\theta$ from Equation 6, the constrained optimization problem in Equation 10 can be rewritten in a purely reward-relational form:

$$\begin{aligned} \min_{\theta} \quad & -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\big(r_\theta(x,y_w)-r_\theta(x,y_l)\big)\right] \\ \text{s.t.} \quad & \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\gamma\, r_\theta(x,y_l)-r_\theta(x,y_w)\right]\le 0, \end{aligned} \tag{11}$$

where $\gamma = \frac{k-1}{k+1} \in (0,1)$ is the reparameterized constraint coefficient induced by $k$.

To obtain a practical training objective, we adopt a penalty-function formulation with a hinge surrogate and derive the LCPO loss:

$$\mathcal{L}_{\text{LCPO}}(\theta) = \mathcal{L}_{\text{DPO}}(\theta) + \lambda \max\left(0,\, \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\gamma\, r_\theta(x,y_l) - r_\theta(x,y_w)\right]\right), \tag{12}$$

where $\lambda > 0$ denotes the penalty coefficient.

This objective satisfies all three prior design principles. First, when the reward level drifts too far downward relative to the achieved margin, the penalty becomes active and injects an explicit corrective signal along the level direction, making LCPO level-active. Second, as long as the optimization trajectory remains within the feasible region, the hinge term vanishes and the objective reduces exactly to DPO, making LCPO target-preserving. Third, the correction term depends only on the internal relation between chosen and rejected rewards through the coefficient $\gamma$, without enforcing any zero-point baseline or externally specified absolute target, making LCPO calibration-free. We provide a complementary gradient analysis of the active LCPO penalty in Appendix B.

Consequently, LCPO provides a targeted and minimally intrusive correction to reward degeneration: it intervenes only when the optimization trajectory enters a degenerate low-level region, and otherwise preserves the original preference optimization behavior of DPO.

## 4 Experiments

**Baselines.** We compare LCPO against a diverse set of representative preference optimization baselines, including DPO (Rafailov et al., 2023), DPO+NLL (Pang et al., 2024; Xu et al., 2024), DPOP (Pal et al., 2024), IPO (Azar et al., 2024), APO-zero (D'Oosterlinck et al., 2025), and Cal-DPO (Xiao et al., 2024). These baselines cover the main intervention families proposed for mitigating reward degeneration, including likelihood regularization, anchor-based reward shaping, and explicit reward calibration. We adopt Qwen2.5-7B-Instruct (Yang et al., 2024) as the primary backbone. To ensure a fair comparison, we first fine-tune it on chosen responses to obtain a standardized SFT checkpoint, from which all preference optimization methods are initialized. All methods share the same general training hyperparameters, such as learning rate and batch size, while method-specific hyperparameters are separately tuned within reasonable ranges to ensure competitive performance. Comprehensive implementation and hyperparameter details are provided in Appendix C.

**Benchmarks.** We evaluate LCPO on two representative preference alignment benchmarks. UltraFeedback (Cui et al., 2023) is used for general instruction-following. Models trained on this dataset are evaluated on AlpacaEval-2.0 (Dubois et al., 2024), where we report both the raw win rate (WR) and the length-controlled win rate (LC) against GPT-4-Turbo. TL;DR (Völske et al., 2017) serves as a complementary summarization benchmark. Following standard protocols, we conduct pairwise evaluations with GPT-4o on the test set and report win rates against both the dataset's ground-truth chosen responses and the original SFT outputs. In addition to downstream generation quality, we track the test-set implicit rewards for both chosen and rejected responses ($r_w$ and $r_l$) to directly assess whether reward degeneration is mitigated during training. Additional results across diverse model families and scales are provided in Appendix D.

### 4.1 Main Results

Table 2: Comparison of different preference optimization methods on UltraFeedback and TL;DR. Best results are in **bold**, and second-best results are underlined.

| Method | UltraFeedback | | | | TL;DR | | | |
|---|---|---|---|---|---|---|---|---|
| | $r_w$ | $r_l$ | WR (%) | LC (%) | $r_w$ | $r_l$ | vs. chosen (%) | vs. SFT (%) |
| SFT | — | — | 32.8 | 33.5 | — | — | 46.5 | — |
| DPO | -12.58 | -19.12 | 34.1 | 34.7 | -4.78 | -6.31 | 44.5 | 48.4 |
| DPO+NLL | 1.05 | -3.53 | 39.1 | 38.4 | 0.09 | -0.51 | 55.2 | 60.3 |
| DPOP | 0.69 | -3.68 | 39.5 | 39.1 | 0.17 | -1.29 | 52.3 | 56.0 |
| IPO | -0.77 | -1.15 | 34.6 | 33.2 | -0.59 | -0.73 | 43.9 | 47.7 |
| APO-zero | 0.81 | -6.21 | 38.6 | 37.0 | 0.08 | -2.52 | 60.5 | 65.2 |
| Cal-DPO | 0.12 | -0.15 | 36.4 | 36.3 | 0.08 | -0.13 | 49.1 | 54.7 |
| LCPO | -0.46 | -4.32 | **41.4** | **41.8** | -0.23 | -1.76 | **68.4** | **74.8** |

Table 2 presents the main results on two representative preference optimization benchmarks: UltraFeedback, evaluated on AlpacaEval-2.0, and TL;DR, evaluated via GPT-4o pairwise comparison. Across both benchmarks, LCPO achieves the best overall performance among all competing methods. On UltraFeedback, LCPO attains the highest raw win rate and length-controlled win rate, reaching $41.4\%$ and $41.8\%$, respectively. On TL;DR, LCPO again delivers substantial gains, achieving a $68.4\%$ win rate against the ground-truth chosen responses and $74.8\%$ against the initial SFT baseline. These results show that mitigating reward degeneration can translate into substantial improvements in downstream alignment quality.

The direct contrast between LCPO and vanilla DPO most clearly illustrates the necessity of our approach. As shown in Table 2, DPO exhibits severe reward degeneration on both benchmarks: on UltraFeedback, the implicit rewards of both chosen and rejected responses collapse into extremely negative values ($r_w = -12.58$, $r_l = -19.12$), and a similar collapse appears on TL;DR ($r_w = -4.78$, $r_l = -6.31$). Consequently, despite continuing to enlarge the reward margin, DPO yields only modest gains over SFT, suggesting that uncontrolled reward-level collapse can substantially harm generation quality. In contrast, LCPO significantly outperforms DPO across all evaluation metrics while maintaining a substantially more stable reward profile.

This dynamic is further visualized in Figure 2. In the training reward curves, DPO exhibits a continuous, entangled downward collapse of both chosen and rejected rewards, whereas LCPO effectively arrests this trend. In the $(u, v)$ phase plane, the optimization trajectory of DPO drifts steadily into the low-level degenerate region, whereas LCPO prevents such drift by enforcing the level constraint. These dynamics indicate that LCPO improves alignment quality by correcting the level-wise pathology left uncontrolled by vanilla DPO.

Compared to other baselines, LCPO also highlights that not all remedies for reward degeneration operate in the same way. IPO, for instance, provides no corrective pressure along the reward-level direction; instead, it mainly limits further margin growth, which can delay but does not fundamentally prevent downward reward-level drift. By contrast, most other baselines mitigate degeneration by implicitly or explicitly pushing the chosen reward upward, through likelihood regularization, zero-point anchoring, or calibrated target values. While such strategies can alleviate reward degeneration, they also introduce externally specified priors on the absolute reward value, which is non-identifiable

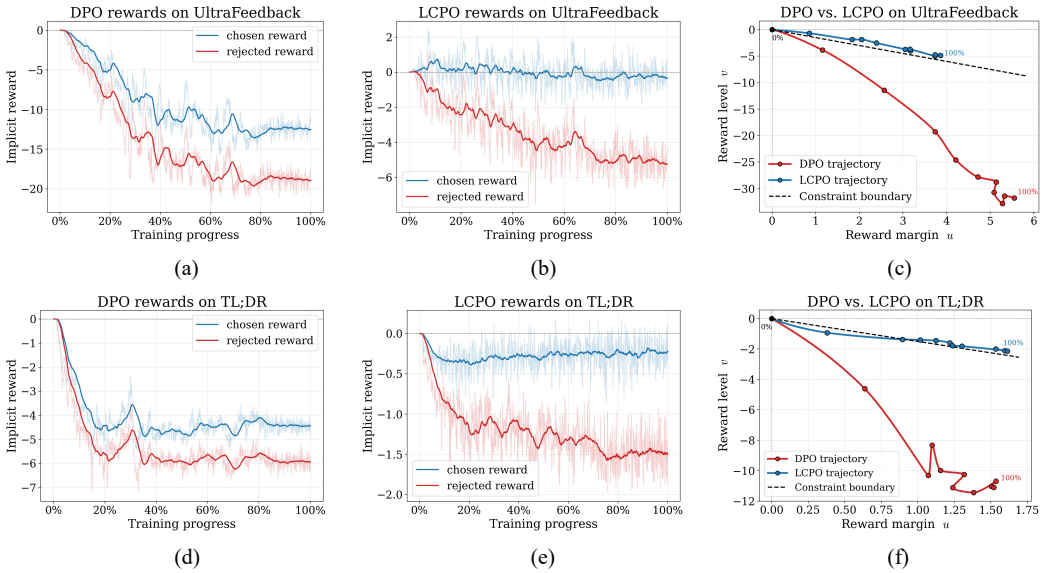



Figure 2: Training dynamics of DPO and LCPO on UltraFeedback (top row) and TL;DR (bottom row). **(a, b, d, e)** Implicit rewards of chosen and rejected responses over training. **(c, f)** Optimization trajectories in the margin-level plane, where each point corresponds to the test-set reward margin and level measured every 10% of training progress.

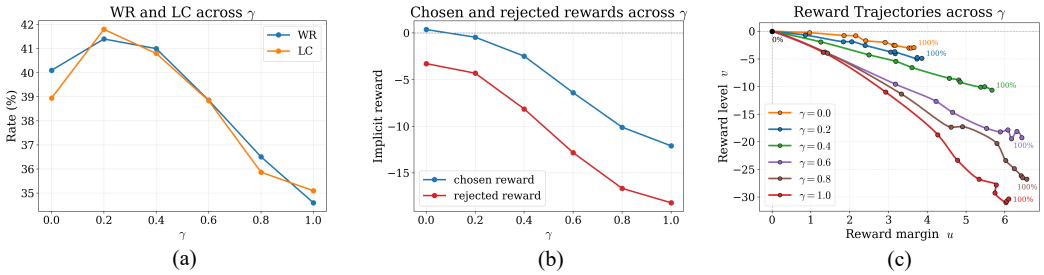



Figure 3: Ablation of the constraint coefficient $\gamma$ on UltraFeedback. **(a)** Downstream win rate (WR) and length-controlled win rate (LC) on AlpacaEval-2.0. **(b)** Test-set implicit rewards ($r_w$ and $r_l$). **(c)** Optimization trajectories in the margin-level plane.

from preference data. Importantly, our results show that neither a higher nor a non-negative chosen reward is sufficient on its own for stronger downstream performance. LCPO differs from these methods by imposing a relative level constraint without restricting the sign of the implicit rewards or hard-coding absolute reward targets. Consistent with this design, LCPO achieves the best overall alignment performance even though its chosen reward remains slightly negative, showing that effective degeneration correction does not require artificially enforcing reward positivity.

In summary, our results lend empirical support to the proposed design principles. In particular, they suggest that reward-degeneration mitigation can benefit from a prior that is level-active, target-preserving, and calibration-free. Consistent with these principles, LCPO substantially mitigates reward degeneration and delivers robust performance gains over existing baselines.

### 4.2 Ablation Study

We conduct ablation studies on UltraFeedback to examine the roles of the two key hyperparameters in LCPO: the constraint coefficient $\gamma$ and the penalty weight $\lambda$. We analyze how they affect downstream alignment performance, test-set reward statistics, and optimization trajectories.

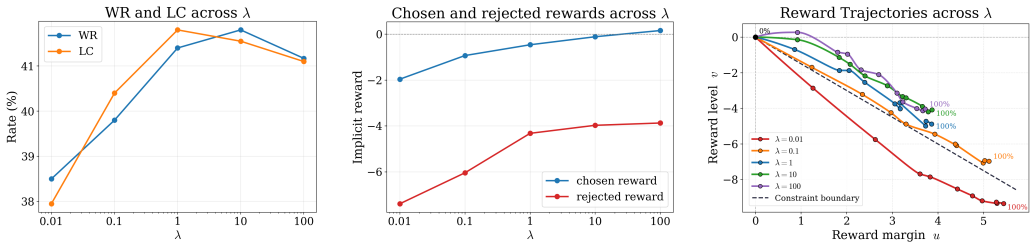



Figure 4: Ablation of the penalty weight $\lambda$ on UltraFeedback. **(a)** Downstream win rate (WR) and length-controlled win rate (LC) on AlpacaEval-2.0. **(b)** Test-set implicit rewards ($r_w$ and $r_l$). **(c)** Optimization trajectories in the margin-level plane.

**Constraint coefficient $\gamma$.** We evaluate the performance of LCPO across a wide range of values $\gamma \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. As shown in Figure 3, $\gamma$ plays a critical role in balancing reward-level control and preference margin optimization. In LCPO, $\gamma = \frac{k-1}{k+1}$ governs the slope of the linear boundary: as $\gamma$ increases from 0.0 to 1.0, the constraint boundary becomes progressively relaxed, and the objective approaches vanilla DPO at $\gamma = 1.0$. This geometric effect is directly reflected in the reward statistics and phase trajectories: as $\gamma$ increases, both test-set chosen and rejected rewards drift downward, and the optimization trajectories move deeper into the low-level region. In terms of downstream performance, setting $\gamma$ too small or too large degrades generation quality. When $\gamma = 0.0$, the constraint reduces to an explicit zero-anchoring of the chosen reward. Although this suppresses degeneration, it can over-restrict optimization and lead to a suboptimal trade-off between level control and preference optimization. Conversely, when $\gamma$ is too large, the constraint becomes too weak to prevent reward degeneration. The best overall performance is achieved at $\gamma = 0.2$, where the boundary provides effective level control without compromising margin growth.

**Penalty weight $\lambda$.** We evaluate the sensitivity of LCPO across a wide range of penalty weights $\lambda \in \{0.01, 0.1, 1, 10, 100\}$. As illustrated in Figure 4, $\lambda$ directly modulates the intensity of the corrective gradient along the reward-level direction. When $\lambda$ is too small, the penalty provides insufficient restoring force to prevent the trajectory from drifting into the low-level region. As a result, the optimization trajectory remains below the linear boundary, exhibiting reward degeneration similar to that of vanilla DPO. Conversely, when $\lambda$ is excessively large, the penalty term dominates the loss function and over-corrects the reward level at the expense of preference optimization, leading to a degraded win rate. The best performance is achieved at $\lambda = 1$, which strikes a desirable balance between suppressing reward degeneration and preserving the original preference objective. This observation further supports our view that degeneration correction should act as a targeted prior on the reward level, rather than redefining preference optimization around a dominant auxiliary target.

## 5 Conclusion

In this work, we presented a principled investigation of reward degeneration in DPO. By analyzing implicit rewards through the lens of reward margin and reward level, we showed that DPO is structurally unconstrained along the reward-level direction and lacks an intrinsic restoring force once training drifts into a low-level region. We further established that the absolute reward value is non-identifiable from pairwise preference data, which recasts the mitigation of reward degeneration as a prior design problem. Based on this perspective, we proposed three design principles for such priors: level-active, target-preserving, and calibration-free. Guided by these principles, we introduced Level-Constrained Preference Optimization (LCPO), a simple yet effective method that imposes a relational constraint between chosen and rejected rewards to counter reward-level collapse without relying on externally calibrated targets. Extensive experiments showed that LCPO substantially mitigates reward degeneration while consistently outperforming preference optimization baselines across diverse benchmarks. We hope that our margin-level perspective and prior design framework can provide guidance for future research on preference optimization and alignment.

## AI USE STATEMENT

In this work, we used generative AI tools for limited language editing, proofreading, and formatting assistance during manuscript preparation. We did not use generative AI tools to generate the core research ideas, theoretical analysis, method design, experimental design, experimental results, or final scientific claims of this paper. We also did not rely on generative AI tools to produce the final evaluation data, numerical results, or conclusions reported in this work. All AI-assisted edits were carefully reviewed by the authors, and any suggested text or formatting changes were manually checked before inclusion. The authors take full responsibility for the final content of this work, including all text, claims, code, and experimental results.

## ETHICS STATEMENT

This work is a methodological study of preference optimization objectives for large language models. It does not involve human subjects, sensitive personal data, or new high-risk applications. All experiments use publicly available models and standard public datasets. We therefore foresee no immediate ethical concerns or potential negative societal impacts arising directly from this work.

## REPRODUCIBILITY STATEMENT

We provide detailed implementation and training configurations in Appendix C, including the LCPO algorithm, baseline formulations, optimizer settings, hardware setup, shared hyperparameters, and method-specific tuning ranges. Additional experimental results across model scales, model architectures, and an extra benchmark are reported in Appendix D. The prompts used for LLM-as-a-Judge evaluation are provided in Appendix E. The complete code is provided in the supplementary material. Together, these materials are intended to support the reproducibility of our method, experimental setup, and evaluation procedure.

## REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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.

Yi Khuen Chai. Adaptive kl control for direct preference optimization in instruction-following llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 41480–41482, 2026.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. Noise contrastive alignment of language models with explicit rewards. *Advances in neural information processing systems*, 37:117784–117812, 2024.

Wei Chen, Yubing Wu, Junmei Yang, Delu Zeng, Qibin Zhao, John Paisley, Min Chen, and Zhou Wang. Towards disentangled preference optimization dynamics: Suppress the loser, preserve the winner. *arXiv preprint arXiv:2604.18239*, 2026.

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

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Karel D'Oosterlinck, Winnie Xu, Chris Develder, Thomas Demeester, Amanpreet Singh, Christopher Potts, Douwe Kiela, and Shikib Mehri. Anchored preference optimization and contrastive revisions: Addressing underspecification in alignment. *Transactions of the Association for Computational Linguistics*, 13:442–460, 2025.

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

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. Towards analyzing and understanding the limitations of dpo: A theoretical perspective. *arXiv preprint arXiv:2404.04626*, 2024.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization. *arXiv preprint arXiv:2302.08215*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*, pp. 1645–1654. PMLR, 2017.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220, 2022.

Sangkyu Lee, Janghoon Han, Hosung Song, Stanley Jungkyu Choi, Honglak Lee, and Youngjae Yu. Kl penalty control via perturbation for direct preference optimization. *Advances in Neural Information Processing Systems*, 38:168095–168121, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

Richard Y Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637, 2024.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4998–5017, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *International Conference on Learning Representations*, volume 2025, pp. 24791–24834, 2025.

Yi Ren and Danica Sutherland. Learning dynamics of llm finetuning. In *International Conference on Learning Representations*, volume 2025, pp. 70523–70563, 2025.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

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

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL `https://github.com/huggingface/trl`.

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. Making large language models better reasoners with alignment. *arXiv preprint arXiv:2309.02144*, 2023.

Teng Xiao, Yige Yuan, Huaisheng Zhu, Mingxiao Li, and Vasant G Honavar. Cal-dpo: Calibrated direct preference optimization for language model alignment. *Advances in Neural Information Processing Systems*, 37:114289–114320, 2024.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024. URL `https://arxiv.org/abs/2412.15115`.

Xiliang Yang, Feng Jiang, Qianen Zhang, Lei Zhao, and Xiao Li. Dpo-shift: Shifting the distribution of direct preference optimization. *arXiv preprint arXiv:2502.07599*, 2025.

Hui Yuan, Yifan Zeng, Yue Wu, Huazheng Wang, Mengdi Wang, and Liu Leqi. A common pitfall of margin-based language model alignment: Gradient entanglement. In *International Conference on Learning Representations*, volume 2025, pp. 5278–5297, 2025.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*, 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 Proof of Proposition

**Proposition 3.1.** *Let $u_\theta$ and $v_\theta$ be defined as in Equation 6. Then, for any preference pair $(x, y_w, y_l)$,*

$$v_\theta \le -u_\theta + C(x, y_w), \tag{8}$$

*where $C(x, y_w) = -2\beta \log \pi_{\text{ref}}(y_w \mid x)$ is a constant independent of $\theta$.*

*Proof.* Recall the definitions of the implicit reward, reward margin, and reward level:

$$r_\theta(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}, \tag{13}$$

$$u_\theta(x, y_w, y_l) = r_\theta(x, y_w) - r_\theta(x, y_l), \tag{14}$$

$$v_\theta(x, y_w, y_l) = r_\theta(x, y_w) + r_\theta(x, y_l). \tag{15}$$

For brevity, we omit the dependence on $(x, y_w, y_l)$ when no confusion arises.

By substituting the definition of $r_\theta$ into $u_\theta$ and $v_\theta$, we obtain

$$u_\theta = \beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}, \tag{16}$$

$$v_\theta = \beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} + \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}. \tag{17}$$

Hence,

$$u_\theta + v_\theta = 2\,\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)}. \tag{18}$$

Now, since $\pi_\theta(\cdot \mid x)$ is a normalized probability distribution, we have

$$\pi_\theta(y_w \mid x) \le 1. \tag{19}$$

Therefore,

$$u_\theta + v_\theta = 2\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} \tag{20}$$

$$\le 2\beta \log \frac{1}{\pi_{\text{ref}}(y_w \mid x)} \tag{21}$$

$$= -2\beta \log \pi_{\text{ref}}(y_w \mid x). \tag{22}$$

Rearranging yields

$$v_\theta \le -u_\theta - 2\beta \log \pi_{\text{ref}}(y_w \mid x). \tag{23}$$

Defining

$$C(x, y_w) = -2\beta \log \pi_{\text{ref}}(y_w \mid x), \tag{24}$$

we obtain

$$v_\theta \le -u_\theta + C(x, y_w), \tag{25}$$

as desired. Since $\pi_{\text{ref}}$ and $\beta$ are fixed during optimization, $C(x, y_w)$ is independent of $\theta$. □

# B Gradient Analysis of the Active LCPO Penalty

To complement the geometric analysis in the main text, we provide a parameter-space interpretation of LCPO when the level constraint is active. Recall that the LCPO objective is

$$\mathcal{L}_{\mathrm{LCPO}}(\theta) = \mathcal{L}_{\mathrm{DPO}}(\theta) + \lambda \max(0, \Delta), \qquad \Delta = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{B}}[\gamma r_\theta(x, y_l) - r_\theta(x, y_w)], \tag{26}$$

where

$$r_\theta(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)}. \tag{27}$$

When $\Delta \le 0$, the hinge term vanishes and LCPO reduces exactly to standard DPO. Therefore, the additional mechanism of LCPO appears only when the batch-level constraint is violated.

When $\Delta > 0$, the penalty becomes active and takes the form

$$\mathcal{L}_{\mathrm{pen}}(\theta) = \lambda\Delta = \lambda\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{B}}[\gamma r_\theta(x, y_l) - r_\theta(x, y_w)]. \tag{28}$$

Since $\pi_{\mathrm{ref}}$ is fixed with respect to $\theta$, we have

$$\nabla_\theta r_\theta(x, y) = \beta \nabla_\theta \log \pi_\theta(y \mid x). \tag{29}$$

Thus, differentiating Equation 28 with respect to $\theta$ yields

$$\nabla_\theta \mathcal{L}_{\mathrm{pen}}(\theta) = -\lambda\beta\, \mathbb{E}_{\mathcal{B}}[\nabla_\theta \log \pi_\theta(y_w \mid x)] + \lambda\gamma\beta\, \mathbb{E}_{\mathcal{B}}[\nabla_\theta \log \pi_\theta(y_l \mid x)]. \tag{30}$$

Under gradient descent, this active penalty contributes the update

$$-\nabla_\theta \mathcal{L}_{\mathrm{pen}}(\theta) = \lambda\beta\, \mathbb{E}_{\mathcal{B}}[\nabla_\theta \log \pi_\theta(y_w \mid x)] - \lambda\gamma\beta\, \mathbb{E}_{\mathcal{B}}[\nabla_\theta \log \pi_\theta(y_l \mid x)]. \tag{31}$$

This admits a simple mechanistic interpretation: once the level constraint is violated, LCPO adds an extra likelihood-style update for the chosen response and an extra unlikelihood-style update for the rejected response. Because $\gamma \in (0, 1)$, the chosen-side restorative update is strictly stronger than the rejected-side suppressive update. In this sense, the active LCPO penalty introduces an explicit upward corrective bias that is absent in vanilla DPO.

This parameter-space view is consistent with the margin-level geometry in the main text. Let

$$u_\theta = r_\theta(x, y_w) - r_\theta(x, y_l), \qquad v_\theta = r_\theta(x, y_w) + r_\theta(x, y_l). \tag{32}$$

Then the active penalty term inside the expectation can be rewritten as

$$\gamma r_\theta(x, y_l) - r_\theta(x, y_w) = -\frac{1-\gamma}{2} v_\theta - \frac{1+\gamma}{2} u_\theta. \tag{33}$$

Therefore, minimizing the active penalty exerts a level-restoring force of magnitude $(1-\gamma)/2 > 0$ along the reward-level direction and a margin-expanding force of magnitude $(1+\gamma)/2 > 0$ along the reward-margin direction. This helps explain why LCPO can allow the rejected reward to remain negative while still preventing the collapse of the chosen reward and the overall reward level, without imposing any externally calibrated absolute reward target.

# C Implementation Details

## C.1 The LCPO Algorithm

Algorithm 1 summarizes the mini-batch optimization procedure of LCPO.

## C.2 Baselines

**DPO+NLL.** The loss function of DPO+NLL (Pang et al., 2024; Xu et al., 2024) is

$$\mathcal{L}_{\mathrm{DPO+NLL}}(\theta) = \mathcal{L}_{\mathrm{DPO}}(\theta) - \alpha\, \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \pi_\theta(y_w \mid x)\right], \tag{34}$$

where $\alpha > 0$ is the regularization coefficient.

Under the margin-level view, this objective is level-active, since the additional NLL term directly increases the log-probability of the chosen response and thus provides explicit upward pressure on the implicit reward level. It is also calibration-free, because it does not impose any externally specified absolute baseline or calibrated reward target. However, it is not target-preserving, because the NLL term remains active throughout training and continuously biases optimization toward the preferred-response distribution, rather than intervening only when degeneration occurs.

In our experiments, we tune $\alpha \in \{0.01, 0.1, 1, 10, 100\}$ and adopt $\alpha = 1$ on both UltraFeedback and TL;DR.

**Algorithm 1** Level-Constrained Preference Optimization (LCPO)

**Require:** Dataset $\mathcal{D}$, reference policy $\pi_{\text{ref}}$, initial policy $\pi_\theta$, KL coefficient $\beta$, coefficient $\gamma \in (0, 1)$, penalty weight $\lambda > 0$, optimizer Opt

**for** each training step **do**
  Sample a mini-batch $\mathcal{B} = \{(x_i, y_i^w, y_i^l)\}_{i=1}^B \sim \mathcal{D}$
  Compute implicit rewards for each pair:
    $r_i^w \leftarrow \beta \log \frac{\pi_\theta(y_i^w|x_i)}{\pi_{\text{ref}}(y_i^w|x_i)}, \quad r_i^l \leftarrow \beta \log \frac{\pi_\theta(y_i^l|x_i)}{\pi_{\text{ref}}(y_i^l|x_i)}, \quad u_i \leftarrow r_i^w - r_i^l$
  Compute DPO loss: $\mathcal{L}_{\text{DPO}}(\theta) \leftarrow -\frac{1}{B}\sum_{i=1}^B \log \sigma(u_i)$
  Compute constraint violation: $\Delta \leftarrow \frac{1}{B}\sum_{i=1}^B \left(\gamma r_i^l - r_i^w\right)$
  Compute total loss: $\mathcal{L}_{\text{LCPO}}(\theta) \leftarrow \mathcal{L}_{\text{DPO}}(\theta) + \lambda \max(0, \Delta)$
  Update policy parameters: $\theta \leftarrow \text{Opt}\left(\theta, \nabla_\theta \mathcal{L}_{\text{LCPO}}\right)$
**end for**
**return** Optimized policy $\pi_\theta$

**DPOP.** The loss function of DPOP (Pal et al., 2024) is

$$\mathcal{L}_{\text{DPOP}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\left(\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right) - \lambda\max\left(0, \log\frac{\pi_{\text{ref}}(y_w \mid x)}{\pi_\theta(y_w \mid x)}\right)\right)\right], \tag{35}$$

where $\lambda > 0$ is the coefficient of the positive-reward correction term.

Under the margin-level view, DPOP is level-active, since the additional correction term explicitly pushes the chosen reward away from excessively low implicit reward regions. It is also target-preserving in the sense of our definition, because the correction becomes inactive once the chosen reward no longer violates the positivity condition, after which the objective reduces to the standard DPO form. However, it is not calibration-free, because the correction is defined relative to the condition $\log \frac{\pi_{\text{ref}}(y_w|x)}{\pi_\theta(y_w|x)} \leq 0$ or equivalently $r_\theta(x, y_w) \geq 0$, which calibrates the chosen implicit reward to an externally specified zero baseline.

In our experiments, we tune $\lambda \in \{0.01, 0.1, 1, 10, 100\}$ and adopt $\lambda = 10$ on UltraFeedback and $\lambda = 1$ on TL;DR.

**IPO.** The loss function of IPO (Azar et al., 2024) is

$$\mathcal{L}_{\text{IPO}}(\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\left(\log\frac{\pi_\theta(y_w \mid x)\pi_{\text{ref}}(y_l \mid x)}{\pi_\theta(y_l \mid x)\pi_{\text{ref}}(y_w \mid x)} - \frac{1}{2\tau}\right)^2\right], \tag{36}$$

where $\tau > 0$ is the temperature parameter in IPO.

Equivalently, under the implicit reward parameterization $r_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$, the IPO objective constrains the reward margin toward a fixed target value, since

$$\log\frac{\pi_\theta(y_w \mid x)\pi_{\text{ref}}(y_l \mid x)}{\pi_\theta(y_l \mid x)\pi_{\text{ref}}(y_w \mid x)} = \frac{1}{\beta}\big(r_\theta(x, y_w) - r_\theta(x, y_l)\big) = \frac{u_\theta(x, y_w, y_l)}{\beta}. \tag{37}$$

Hence IPO is not level-active, because it acts only on the reward margin and provides no explicit optimization signal along the reward-level direction. It is also not target-preserving, since it globally replaces the standard DPO objective with a squared deviation toward a prescribed margin target, rather than intervening only when degeneration occurs. However, IPO is calibration-free, because it does not impose any absolute baseline on the individual implicit rewards.

In our experiments, we tune $\tau \in \{0.01, 0.1, 1, 10, 100\}$ and adopt $\tau = 0.1$ on both UltraFeedback and TL;DR.

**APO-zero.** The loss function of APO-zero (D'Oosterlinck et al., 2025) is

$$\mathcal{L}_{\mathrm{APO-zero}}(\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[-\sigma(r_\theta(x,y_w)) + \sigma(r_\theta(x,y_l))\right], \tag{38}$$

where

$$r_\theta(x,y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} \tag{39}$$

is the implicit reward.

Under the margin-level view, APO-zero is level-active, because it acts directly on the individual implicit rewards rather than only on their difference, thereby providing explicit optimization pressure along the reward-level direction. It is not target-preserving, since the objective globally reshapes optimization around the anchored reward terms instead of preserving the original DPO objective outside degenerate regions. It is also not calibration-free, because both terms are defined relative to the zero point of the implicit reward: the loss explicitly encourages the chosen reward to lie above zero and the rejected reward to lie below zero, thereby imposing an absolute baseline.

This method introduces no additional method-specific hyperparameters and thus follows the shared experimental settings.

**Cal-DPO.** The loss function of Cal-DPO (Xiao et al., 2024) is

$$\begin{aligned}\mathcal{L}_{\mathrm{Cal-DPO}}(\theta) = &- \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\left(\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right)\right] \\ &+ \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\left(\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \frac{1}{2\beta}\right)^2 + \left(\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)} + \frac{1}{2\beta}\right)^2\right].\end{aligned} \tag{40}$$

Equivalently, under the implicit reward parameterization

$$r_\theta(x,y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)}, \tag{41}$$

the objective can be written as

$$\mathcal{L}_{\mathrm{Cal-DPO}}(\theta) = \mathcal{L}_{\mathrm{DPO}}(\theta) + \frac{1}{\beta^2}\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\left(r_\theta(x,y_w) - \frac{1}{2}\right)^2 + \left(r_\theta(x,y_l) + \frac{1}{2}\right)^2\right]. \tag{42}$$

Under the margin-level view, Cal-DPO is level-active, because the additional quadratic terms act directly on the individual implicit rewards and thus provide explicit optimization pressure along the reward-level direction. It is not target-preserving, since these calibration terms remain active throughout training and globally reshape the optimization objective rather than only correcting degenerate trajectories. It is also not calibration-free, because it explicitly calibrates the chosen and rejected rewards toward fixed absolute target values.

This method introduces no additional method-specific hyperparameters and thus follows the shared experimental settings.

### C.3 Training Configurations

**Experimental Setup** Our implementation of LCPO is built upon the Transformer Reinforcement Learning (TRL) (von Werra et al., 2020) library. We fine-tune all parameters of the pretrained model in bfloat16 precision. All experiments are conducted on eight H20 GPUs with DeepSpeed-Zero3 (Rajbhandari et al., 2020). We use the AdamW (Loshchilov & Hutter, 2017) optimizer with a cosine learning rate scheduler. For each benchmark dataset, all methods are trained for one epoch.

**Hyperparameters.** For shared hyperparameters, we adopt a batch size of 64 and a learning rate of $5 \times 10^{-6}$, with the KL coefficient $\beta$ set to 0.1. These parameters are consistently applied across all experiments. All other unspecified hyperparameters follow the default configurations in the TRL library. For LCPO, we tune the two hyperparameters sequentially rather than

through a full two-dimensional grid search. Specifically, we first tune the constraint coefficient $\gamma \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, and then, with $\gamma$ fixed to the selected value, tune the penalty weight $\lambda \in \{0.01, 0.1, 1, 10, 100\}$. We ultimately adopt $\gamma = 0.2$ and $\lambda = 1$ on both UltraFeedback and TL;DR. Similarly, the hyperparameters for all baseline methods are tuned for competitive performance, with detailed configurations specified in Appendix C.2.

# D Additional Experimental Results

## D.1 Generalization Across Model Scales

Table 3: Model scaling results across **1.5B**, **7B**, and **14B** base models on the UltraFeedback and TL;DR benchmarks. Best results are in **bold**, and second-best results are underlined.

| Base Model | Method | UltraFeedback | | | | TL;DR | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $r_w$ | $r_l$ | WR (%) | LC (%) | $r_w$ | $r_l$ | vs. chosen (%) | vs. SFT (%) |
| **Qwen2.5-1.5B -Instruct** | SFT | — | — | 16.0 | 15.6 | — | — | 40.4 | — |
| | DPO | -4.86 | -8.44 | 16.5 | 15.9 | -4.75 | -5.53 | 36.6 | 47.6 |
| | DPO+NLL | 0.59 | -3.16 | 17.3 | 16.5 | 0.27 | -0.31 | 49.2 | 62.6 |
| | DPOP | 0.38 | -2.76 | 18.6 | 18.4 | 0.15 | -0.25 | 46.7 | 57.2 |
| | IPO | -0.63 | -0.97 | 16.3 | 15.7 | -0.65 | -0.85 | 33.9 | 46.1 |
| | APO-zero | 0.64 | -3.51 | 18.1 | 18.0 | -0.85 | -2.64 | 53.0 | 64.4 |
| | Cal-DPO | 0.16 | -0.18 | 17.1 | 16.9 | 0.12 | -0.09 | 45.6 | 59.0 |
| | LCPO | 0.21 | -3.43 | **19.9** | **19.6** | -0.32 | -1.33 | **59.7** | **72.8** |
| **Qwen2.5-7B -Instruct** | SFT | — | — | 32.8 | 33.5 | — | — | 46.5 | — |
| | DPO | -12.58 | -19.12 | 34.1 | 34.7 | -4.78 | -6.31 | 44.5 | 48.4 |
| | DPO+NLL | 1.05 | -3.53 | 39.1 | 38.4 | 0.09 | -0.51 | 55.2 | 60.3 |
| | DPOP | 0.69 | -3.68 | 39.5 | 39.1 | 0.17 | -1.29 | 52.3 | 56.0 |
| | IPO | -0.77 | -1.15 | 34.6 | 33.2 | -0.59 | -0.73 | 43.9 | 47.7 |
| | APO-zero | 0.81 | -6.21 | 38.6 | 37.0 | 0.08 | -2.52 | 60.5 | 65.2 |
| | Cal-DPO | 0.12 | -0.15 | 36.4 | 36.3 | 0.08 | -0.13 | 49.1 | 54.7 |
| | LCPO | -0.46 | -4.32 | **41.4** | **41.8** | -0.23 | -1.76 | **68.4** | **74.8** |
| **Qwen2.5-14B -Instruct** | SFT | — | — | 39.8 | 41.8 | — | — | 47.9 | — |
| | DPO | -12.11 | -18.20 | 42.8 | 44.0 | -4.70 | -6.73 | 45.8 | 46.7 |
| | DPO+NLL | 0.95 | -3.89 | 47.1 | 47.2 | 0.13 | -0.79 | 58.8 | 62.1 |
| | DPOP | 0.62 | -3.43 | 46.9 | 48.4 | 0.28 | -0.64 | 56.6 | 59.5 |
| | IPO | -0.84 | -1.25 | 41.8 | 43.2 | -0.42 | -0.68 | 43.4 | 47.5 |
| | APO-zero | 1.23 | -9.77 | 44.6 | 44.1 | 0.31 | -2.64 | 61.5 | 67.5 |
| | Cal-DPO | 0.11 | -0.20 | 42.5 | 43.0 | 0.11 | -0.14 | 52.4 | 55.2 |
| | LCPO | -0.72 | -5.28 | **49.2** | **49.9** | -0.45 | -1.80 | **70.2** | **75.6** |

To test whether LCPO generalizes across model scales, we evaluate all methods on Qwen2.5-1.5B-Instruct and Qwen2.5-14B-Instruct under the same protocol as in the main text. For each scale, all methods are initialized from the corresponding SFT checkpoint and evaluated on both UltraFeedback and TL;DR. We report downstream alignment metrics together with the test-set implicit rewards of both chosen and rejected responses in Table 3.

LCPO generalizes consistently across all three scales. On UltraFeedback, it achieves the highest WR and LC at every scale, and on TL;DR, it also obtains the best win rates against both the ground-truth chosen responses and the SFT baseline. In contrast, vanilla DPO exhibits severe reward degeneration across all scales, with both chosen and rejected rewards collapsing to extremely negative values, especially for larger models. Notably, LCPO remains the strongest method even when its chosen reward is still slightly negative on the 7B and 14B models (whereas it is slightly positive at $r_w = 0.21$ on the 1.5B model), which further supports our claim that effective degeneration correction does not require consistently enforcing a positive absolute reward value. Overall, these results show that the advantage of LCPO is robust across model scales from 1.5B to 14B parameters.

## D.2 Generalization Across Model Architectures

To examine cross-architecture robustness, we further evaluate all methods on two representative backbones: Llama-3-8B-Instruct (Grattafiori et al., 2024) and Mistral-7B-Instruct (Jiang et al., 2023). We follow the same setup as in the main text: each method is initialized from the corre-

Table 4: Alignment performance across different model architectures (**Qwen2.5-7B-Instruct**, **Llama-3-8B-Instruct**, and **Mistral-7B-Instruct**) on the UltraFeedback and TL;DR benchmarks. Best results are in **bold**, and second-best results are underlined.

| Base Model | Method | UltraFeedback | | | | TL;DR | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $r_w$ | $r_l$ | WR (%) | LC (%) | $r_w$ | $r_l$ | vs. chosen (%) | vs. SFT (%) |
| **Qwen2.5-7B -Instruct** | SFT | — | — | 32.8 | 33.5 | — | — | 46.5 | — |
| | DPO | -12.58 | -19.12 | 34.1 | 34.7 | -4.78 | -6.31 | 44.5 | 48.4 |
| | DPO+NLL | 1.05 | -3.53 | 39.1 | 38.4 | 0.09 | -0.51 | 55.2 | 60.3 |
| | DPOP | 0.69 | -3.68 | 39.5 | 39.1 | 0.17 | -1.29 | 52.3 | 56.0 |
| | IPO | -0.77 | -1.15 | 34.6 | 33.2 | -0.59 | -0.73 | 43.9 | 47.7 |
| | APO-zero | 0.81 | -6.21 | 38.6 | 37.0 | 0.08 | -2.52 | 60.5 | 65.2 |
| | Cal-DPO | 0.12 | -0.15 | 36.4 | 36.3 | 0.08 | -0.13 | 49.1 | 54.7 |
| | LCPO | -0.46 | -4.32 | **41.4** | **41.8** | -0.23 | -1.76 | **68.4** | **74.8** |
| **Llama-3-8B -Instruct** | SFT | — | — | 30.8 | 31.8 | — | — | 47.0 | — |
| | DPO | -8.66 | -16.36 | 29.3 | 29.7 | -5.88 | -6.84 | 46.4 | 49.0 |
| | DPO+NLL | 1.17 | -2.93 | **36.1** | 34.0 | 0.52 | -0.90 | 57.0 | 62.1 |
| | DPOP | 0.26 | -3.67 | 35.1 | 33.8 | -0.21 | -0.69 | 54.4 | 60.5 |
| | IPO | -2.14 | -2.46 | 26.6 | 26.2 | -0.69 | -0.85 | 42.7 | 45.9 |
| | APO-zero | 0.79 | -7.80 | 35.4 | 34.1 | -1.52 | -3.84 | 60.0 | 63.8 |
| | Cal-DPO | 0.16 | -0.28 | 31.2 | 31.1 | 0.15 | -0.11 | 48.5 | 51.6 |
| | LCPO | -0.40 | -5.73 | 35.8 | **35.4** | -0.48 | -1.87 | **67.9** | **75.3** |
| **Mistral-7B -Instruct** | SFT | — | — | 22.6 | 24.8 | — | — | 44.0 | — |
| | DPO | -13.57 | -27.04 | 19.6 | 19.0 | -10.28 | -11.87 | 40.1 | 48.5 |
| | DPO+NLL | 0.55 | -11.79 | 25.4 | 26.5 | -0.38 | -1.70 | 52.8 | 58.2 |
| | DPOP | 0.92 | -3.52 | 26.6 | 25.0 | 0.03 | -1.11 | 46.8 | 53.9 |
| | IPO | -4.87 | -5.24 | 17.4 | 17.9 | -0.86 | -0.99 | 36.2 | 43.7 |
| | APO-zero | 1.55 | -20.64 | 25.4 | 24.6 | -3.28 | -5.63 | 49.7 | 56.4 |
| | Cal-DPO | 0.15 | -0.18 | 20.0 | 19.6 | 0.18 | -0.39 | 39.4 | 48.0 |
| | LCPO | -0.77 | -6.72 | **28.3** | **27.1** | -0.42 | -1.95 | **58.3** | **66.9** |

sponding SFT checkpoint and evaluated on UltraFeedback and TL;DR. Both downstream alignment metrics and test-set implicit reward statistics are reported in Table 4.

LCPO shows strong generalization across architectures. It achieves the best overall results on Qwen2.5-7B-Instruct and Mistral-7B-Instruct, and remains highly competitive on Llama-3-8B-Instruct, achieving the highest length-controlled win rate on UltraFeedback and the top win rates on TL;DR. Across all three backbones, vanilla DPO again suffers from severe reward degeneration, while LCPO consistently avoids extreme reward collapse and delivers the most robust downstream performance. These results suggest that the benefit of LCPO is not tied to a particular model family, but transfers well across distinct model families and architectures.

### D.3 Evaluation on Anthropic-HH Benchmark

Table 5: Evaluation results on the **Anthropic-HH** dataset. Win rates against the chosen responses and the SFT baseline are evaluated using GPT-4o. Best results are in **bold**, and second-best results are underlined.

| Method | Anthropic-HH | | | |
|---|---|---|---|---|
| | $r_w$ | $r_l$ | vs. chosen (%) | vs. SFT (%) |
| SFT | — | — | 40.5 | — |
| DPO | -3.62 | -4.30 | 37.6 | 46.4 |
| DPO+NLL | -0.32 | -0.64 | 44.6 | 52.8 |
| DPOP | 0.10 | -0.12 | 43.4 | 51.3 |
| IPO | -0.27 | -0.37 | 41.2 | 51.2 |
| APO-zero | -2.30 | -3.73 | 46.8 | 56.7 |
| Cal-DPO | 0.06 | -0.12 | 40.0 | 50.5 |
| LCPO | -0.20 | -0.76 | **49.2** | **60.6** |

As an additional preference alignment benchmark, we further test all methods on the Anthropic-HH (Bai et al., 2022) dataset, a widely used helpful–harmless preference benchmark for dialogue alignment. Following the same protocol as in the main text, all methods are initialized from the same SFT checkpoint and evaluated using GPT-4o pairwise comparisons. We report the test-set implicit rewards of both chosen and rejected responses, together with win rates against both the chosen responses and the SFT baseline.

As presented in Table 5, the results show that LCPO also generalizes well to this benchmark. It achieves the best performance among all compared methods, reaching a 49.2% win rate against the chosen responses and 60.6% against the SFT baseline. In contrast, vanilla DPO again exhibits clear reward degeneration, with both chosen and rejected rewards drifting to negative values and downstream performance falling below the SFT baseline when compared against the chosen responses. Other remedies alleviate degeneration to different extents and improve over DPO, but none matches the overall performance of LCPO. These results further demonstrate that the advantage of LCPO is not limited to UltraFeedback and TL;DR, but transfers to an additional preference alignment benchmark as well.

### D.4 Batch-wise vs. Sample-wise Constraint Formulations

In the main paper, we impose the constraint at the expectation level, and implement it in practice via a batch-wise objective as a stochastic approximation to that population-level constraint. Our motivation is that reward degeneration in DPO is primarily observed as a collective downward drift of implicit rewards during training, rather than as a pathology that must be corrected for every individual preference pair at all times. Under this view, applying the correction at the batch level provides a less intrusive intervention: it directly counteracts aggregate reward-level collapse while preserving more flexibility for the original DPO objective to optimize preference margins. By contrast, a sample-wise formulation imposes a stronger local constraint on every pair, which may over-restrict optimization.

To validate this motivation, we compare the batch-wise formulation used in the main paper with a sample-wise alternative on UltraFeedback and TL;DR. The batch-wise variant applies the correction to the batch-average constraint violation:

$$\mathcal{L}_{\text{batch}}(\theta) = \mathcal{L}_{\text{DPO}}(\theta) + \lambda \max\big(0,\, \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{B}}[\gamma r_\theta(x,y_l) - r_\theta(x,y_w)]\big)\,, \tag{43}$$

$$\mathcal{L}_{\text{sample}}(\theta) = \mathcal{L}_{\text{DPO}}(\theta) + \lambda\, \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{B}}[\max(0, \gamma r_\theta(x,y_l) - r_\theta(x,y_w))]\,. \tag{44}$$

Table 6: Comparison between batch-wise and sample-wise constraint formulations on UltraFeedback and TL;DR. Best results are in **bold**, and second-best results are underlined.

| Method | UltraFeedback | | | | TL;DR | | | |
|---|---|---|---|---|---|---|---|---|
| | $r_w$ | $r_l$ | WR (%) | LC (%) | $r_w$ | $r_l$ | vs. chosen (%) | vs. SFT (%) |
| DPO | -12.58 | -19.12 | 34.1 | 34.7 | -4.78 | -6.31 | 44.5 | 48.4 |
| LCPO-Sample | -0.28 | -2.67 | <u>39.8</u> | <u>39.4</u> | -0.12 | -1.08 | <u>63.1</u> | <u>69.7</u> |
| LCPO-Batch | -0.46 | -4.32 | **41.4** | **41.8** | -0.23 | -1.76 | **68.4** | **74.8** |

As shown in Table 6, both variants substantially improve over vanilla DPO, indicating that adding an explicit correction along the reward-level direction is beneficial. At the same time, the two formulations behave differently. The sample-wise variant maintains less negative implicit rewards, suggesting a stronger local correction effect. However, it also yields a smaller reward margin and consistently underperforms the batch-wise formulation on both benchmarks. In contrast, the batch-wise variant achieves the best downstream results while still effectively mitigating reward degeneration. These results support our motivation that, for reward degeneration under DPO, batch-wise correction provides a better trade-off between level control and preference optimization than its sample-wise constraint.

## E Evaluation Prompts

In this section, we present the evaluation prompts used for LLM-as-a-Judge assessments on the TL;DR and Anthropic-HH benchmarks. To ensure fairness and reduce evaluation bias, we adopt a

set of standardized guidelines in all pairwise comparisons. Specifically, the two candidate responses are presented in random order for each evaluation instance, and we report results after accounting for such order randomization to mitigate position bias. We also use the same evaluation prompt, decoding configuration, and judging procedure across all compared methods. In addition, the LLM judge is explicitly instructed not to prefer a response merely because it appears first or is longer, and ties are allowed when the overall quality of the two candidates is effectively indistinguishable. In our win-rate calculations, ties are credited as a 0.5 wins to each side.

### E.1 Evaluation Prompt for TL;DR

**System Prompt: TL;DR Pairwise Evaluation**

You are an impartial expert evaluator of Reddit TL;DR summaries. Compare two candidate summaries using only the source post.

**Prioritize criteria in this order**:

1. **Faithfulness**: all claims must be supported by the source; contradictions and hallucinations are severe errors.
2. **Coverage**: capture the central situation, key actions, and main outcome or question.
3. **Relevance and conciseness**: include important information without unnecessary detail.
4. **Clarity**: be coherent and understandable on its own.

**Evaluation Guidelines**:
Do not favor a candidate because it appears first or is longer. If their overall quality is effectively equal, choose tie.

**Output Format**:
Return only valid JSON strictly matching the following schema:
`{"winner": "A" | "B" | "tie","reason": "brief decisive explanation"}`

### E.2 Evaluation Prompt for Anthropic-HH

**System Prompt: Anthropic-HH Pairwise Evaluation**

You are an impartial expert evaluator of assistant responses in a multi-turn conversation. Compare two candidate responses to the final user message using the full conversation context.

**Prioritize these criteria**:

1. **Harmlessness and safety**: do not endorse or facilitate harmful, illegal, deceptive, abusive, or privacy-violating behavior. A safe refusal is preferable when the request is unsafe.
2. **Helpfulness and relevance**: directly address the user's actual request and provide useful information when it is safe to do so.
3. **Correctness and honesty**: avoid factual errors, fabricated claims, and unjustified certainty; acknowledge important uncertainty or limitations.
4. **Clarity and quality**: be coherent, appropriately detailed, and professional without unnecessary repetition.

**Evaluation Guidelines**:
Judge overall response quality. Do not favor a response because it appears first or is longer. If their overall quality is effectively equal, choose tie.

**Output Format**:
Return only valid JSON strictly matching the following schema:
`{"winner": "A" | "B" | "tie","reason": "brief decisive explanation"}`