GapPO: Gradient-Adaptive Pairwise Preference Optimization

Anonymous ACL submission

Abstract

Aligning large language models to human preferences requires training on pairwise comparisons between candidate responses. Existing preference optimization methods assign equal gradient weight to every pair, regardless of whether the quality difference is large or negligibly small. We introduce GapPO (Gradient-Adaptive Pairwise Preference Optimization), a preference optimization method designed to directly improve pairwise ranking accuracy in large language models. Standard methods currently treat all pairs equally: a pair scoring 4.8 vs. 1.2 receives the same gradient weight as one scoring 3.2 vs. 2.9, diluting clear signal with annotator noise. GapPO corrects this by weighting each pair by the absolute quality-score gap $|\delta| = |\texttt{score}_{\text{chosen}} - \texttt{score}_{\text{rejected}}|$, so that gradient mass concentrates on the most discriminative comparisons. Since the model is shaped more by reliable comparisons, its implicit reward function better separates high-quality from low-quality responses at test time. Beyond improving pairwise accuracy (PWA), score-gap weighting improves Spearman rank correlation between model rewards and annotation scores, which is the calibration property required to scale from pairwise to listwise ranking. Evaluated on UltraFeedback binarized across Qwen2.5-0.5B, Gemma-2-2B, and Mistral-7B, GapPO consistently outperforms SimPO, CPO, IPO, and AlphaPO baselines.

1 Introduction

Ranking is a core problem in document retrieval (Manning et al., 2008; Liu, 2009) and language model alignment (Christiano et al., 2017; Ouyang et al., 2022). Given a query and a set of candidate responses, a model must assign implicit scores that correctly order them by quality. Large language models (LLMs) are increasingly being used as rankers (Sun et al., 2023; Hou et al., 2024; Qin et al., 2024), but standard preference optimization methods are designed for general alignment, not specifically targeting ranking accuracy. They optimize a binary preference signal (chosen vs. rejected) and treat every pair with the same gradient weight, regardless of how different the two responses actually are.

This uniformity is a poor fit for ranking. A pair where one response scores 4.8 and another scores 1.2 is an unambiguous signal where any reasonable judge would agree on the ordering. Conversely, a pair with scores 3.2 and 2.9 is near a tie, which suggests that the label may reflect annotator noise more than a genuine quality difference. Training a ranker on both pairs with equal weight dilutes the strong signal with the weak. The model spends as much gradient on borderline cases as on clear-cut ones.

GapPO corrects this by deriving a per-pair training weight directly from the quality-score gap $|\delta^{(i)}| = |s_w^{(i)} - s_l^{(i)}|$, where s_w and s_l are the human annotator scores for the chosen and rejected responses. Pairs with large $|\delta|$ receive proportionally more gradient while pairs with small $|\delta|$ are down-weighted but not discarded. This is structurally different from the alternative of hard filtering (discarding all pairs with $|\delta| < \tau$). The continuous weight in GapPO preserves moderate-$|\delta|$ pairs at reduced contribution, while a hard filter treats them identically to high-$|\delta|$ pairs after thresholding, discarding useful signal below the cutoff.

GapPO introduces score-gap weighting as a general mechanism that can be applied to any per-sample preference loss. We build on SimPO's reference-free, length-normalized reward (Meng et al., 2024) and multiply each pair's loss by $|\delta^{(i)}|$, shifting gradient mass toward the most reliable comparisons. Weights are left unnormalized, so batches rich in high-$|\delta|$ pairs produce larger update steps, allowing the training signal to be strongest when the evidence is clearest.

The implicit reward $r(y|\mathbf{x})$ learned during pref-

erence optimization is a ranking function: it assigns a score to every response, and those scores induce an ordering. Pairwise accuracy (PWA), which is the fraction of held-out pairs the model correctly ranks, directly measures whether that ordering agrees with human preferences. While PWA is necessary, correctly ordering isolated pairs does not guarantee a globally consistent ranking across all responses to a prompt. We also measure Spearman rank correlation between model rewards and annotation scores, a continuous indicator of reward calibration and a direct bridge to listwise ranking, where all k responses must be jointly ordered.

Our main contributions are:

- **Identifying a fundamental gap in preference optimization**: existing methods treat all training pairs with equal gradient weight, ignoring the reliability signal embedded in annotation score differences. A pair with $|\delta| = 4.5$ is treated identically to one with $|\delta| = 0.3$, causing models to spend as much gradient on annotator noise as on unambiguous quality signal.

- **GapPO**: a novel preference optimization loss that directly addresses this gap by weighting each training pair by its annotation score gap $|\delta|$. GapPO concentrates gradient on the most reliable comparisons and weights are left unnormalized so that batch informativeness modulates the effective learning rate, allowing training to be strongest precisely when the evidence is clearest.

- **Consistent empirical gains across models and fine-tuning regimes**: GapPO outperforms SimPO, CPO, IPO, and AlphaPO on peak pairwise accuracy (PWA) and Spearman rank correlation across Qwen2.5-0.5B, Gemma-2-2B, and Mistral-7B under both full and MLP-only fine-tuning, demonstrating that score-gap weighting generalizes across model families, scales, and parameter-efficient trainings.

2 Preliminaries

2.1 Problem Definition

Let $\mathcal{D} = \{(\mathbf{x}^{(i)}, y_w^{(i)}, y_l^{(i)}, s_w^{(i)}, s_l^{(i)})\}_{i=1}^N$ be a preference dataset where $\mathbf{x}^{(i)}$ is a prompt, $y_w^{(i)}$ and $y_l^{(i)}$ are the chosen and rejected responses, and $s_w^{(i)}, s_l^{(i)} \in \mathbb{R}$ are their corresponding human annotation scores. Each trained policy π_θ induces an implicit reward function $r(y|\mathbf{x})$ that assigns a scalar score to every response. We define the score gap as:

$$\delta^{(i)} = \left| s_w^{(i)} - s_l^{(i)} \right|. \tag{1}$$

The goal is to learn π_θ such that the induced reward correctly ranks responses according to human preference. Formally, we aim to minimize the expected ranking error:

$$\mathcal{R}(\theta) = \mathbb{E}_{(\mathbf{x}, y_w, y_l)\sim\mathcal{D}}[\mathbf{1}[r(y_w|\mathbf{x}) \leq r(y_l|\mathbf{x})]], \tag{2}$$

which is the complement of pairwise accuracy (PWA). Since this objective is non-differentiable, preference optimization methods minimize a surrogate loss. The key limitation of existing surrogates is that they weight every pair equally, making the optimization indifferent to $\delta^{(i)}$. Pairs where annotation scores are nearly identical carry as much gradient as pairs with large, unambiguous score gaps.

2.2 LLMs as Rankers

Using LLMs to rank candidate responses is a growing area of research (Sun et al., 2023; Ma et al., 2023). Most approaches either prompt a frozen LLM in a listwise or pointwise fashion, or fine-tune on binary preference labels. A key limitation of these approaches is that they treat the ranking problem as a classification task: the model learns to distinguish chosen from rejected responses without access to how large the quality difference between them actually is. GapPO addresses this by fine-tuning the implicit reward function directly on graded annotation scores, making ranking accuracy an explicit training objective rather than a byproduct of binary preference learning.

2.3 Reference-Free Preference Optimization

The dominant paradigm for aligning LLMs with human preferences is reinforcement learning from human feedback (RLHF) (Christiano et al., 2017), which trains a reward model on pairwise comparisons and then optimizes the policy via PPO (Schulman et al., 2017). DPO (Rafailov et al., 2023) simplified optimizes this policy on preference pairs without an explicit reward model, showing that the RLHF objective can be reparameterized as a supervised loss. Subsequent work has explored modifications to the DPO objective. IPO (Azar et al.,

2024) replaces the logistic loss with a squared loss to avoid overfitting to hard labels. CPO (Xu et al., 2024) removes the reference model entirely while SimPO (Meng et al., 2024) introduces length normalization and a target reward margin to improve calibration. AlphaPO (Gupta et al., 2025) generalizes the reward shape with an α-parameterized transformation. However, all of these methods derive their training signal purely from the binary preference label (chosen vs. rejected), ignoring the magnitude of the underlying quality difference. GapPO introduces annotation score gaps as an explicit training signal, making gradient allocation a function of annotation confidence rather than label identity.

We build on reference-free preference objectives, which train the model without a frozen reference model at inference time. The key baselines are formalized below.

CPO (Xu et al., 2024) applies a hinge loss on raw log-probabilities:

$$\mathcal{L}_{\text{CPO}}(\theta) = -\mathbb{E}\Big[\log \sigma\Big(\beta \log \frac{\pi_\theta(y_w|\mathbf{x})}{\pi_\theta(y_l|\mathbf{x})} - \beta\Big)\Big]. \quad (3)$$

IPO (Azar et al., 2024) uses a squared loss to reduce overfitting to hard labels:

$$\mathcal{L}_{\text{IPO}}(\theta) = \mathbb{E}\Bigg[\Bigg(\log \frac{\pi_\theta(y_w|\mathbf{x})}{\pi_{\text{ref}}(y_w|\mathbf{x})} - \log \frac{\pi_\theta(y_l|\mathbf{x})}{\pi_{\text{ref}}(y_l|\mathbf{x})} - \frac{1}{2\beta}\Bigg)^2\Bigg]. \quad (4)$$

SimPO (Meng et al., 2024) uses a length-normalized log-probability reward $r(y|\mathbf{x}) = \frac{1}{|y|}\log \pi_\theta(y|\mathbf{x})$ with a target margin γ:

$$\mathcal{L}_{\text{SimPO}}(\theta) = -\mathbb{E}\Big[\log \sigma\Big(\beta\big(r(y_w|\mathbf{x}) - r(y_l|\mathbf{x})\big) - \gamma\Big)\Big]. \quad (5)$$

AlphaPO (Gupta et al., 2025) replaces the log reward with an α-parameterized transformation $r_\alpha(y|\mathbf{x}) = \frac{1}{|y|}\sum_t \frac{\pi_\theta(y_t|\mathbf{x})^{\alpha}-1}{\alpha}$, modifying the reward shape while keeping pair weights uniform.

All four methods treat every preference pair with equal gradient weight, providing no mechanism to use the magnitude of the annotation score gap as a training signal.

2.4 Weighted and Filtered Preference Optimization

Several works have explored using data quality signals to improve preference learning. Zhou et al. (2023) weight pairs by annotator confidence scores derived from inter-annotator agreement; Liu et al. (2024) use rejection sampling to discard unreliable pairs before training. These approaches either require additional annotator metadata or resort to binary filtering. GapPO differs in using a differentiable continuous weight derived directly from annotation score gaps, preserving all training pairs while proportionally down-weighting noisy comparisons. Wu et al. (2025) take a related direction by adapting the reward margin in DPO itself; GapPO follows the same design principle applied to SimPO, using annotation score gaps as the adaptive signal rather than a learned margin parameter. It also differs in objective: prior work targets general alignment, whereas GapPO explicitly targets ranking accuracy as measured by PWA and Spearman correlation.

3 Methodology: Gradient-Adaptive Pairwise Preference Optimization (GapPO)

GapPO is motivated by a simple observation: annotation score gaps are a direct proxy for how reliable a preference label is, yet no existing preference optimization method uses this signal to allocate gradient.

3.1 Score-Gap Weighting: Concentrating Gradient on Reliable Pairs

The central insight of GapPO is that human annotator score gaps are a direct proxy for pair reliability, and gradient allocation should reflect this. Let $s_w^{(i)}$ and $s_l^{(i)}$ be the quality scores for the chosen and rejected responses of the i-th training pair respectively. We define:

$$\delta^{(i)} = \left|s_w^{(i)} - s_l^{(i)}\right|. \quad (6)$$

We map $\delta^{(i)}$ to a non-negative weight via linear scaling:

$$w^{(i)} = \delta^{(i)}. \quad (7)$$

The GapPO objective is then:

$$\mathcal{L}_{\text{GapPO}}(\theta) = -\frac{1}{B}\sum_{i=1}^{B} w^{(i)} \log \sigma\Big(\beta\big(r(y_w^{(i)}|\mathbf{x}^{(i)}) - r(y_l^{(i)}|\mathbf{x}^{(i)})\big) - \gamma\Big), \quad (8)$$

where $r(y|\mathbf{x}) = \frac{1}{|y|}\log \pi_\theta(y|\mathbf{x})$ is a reference-free, length-normalized log reward. When all $\delta^{(i)}$ are

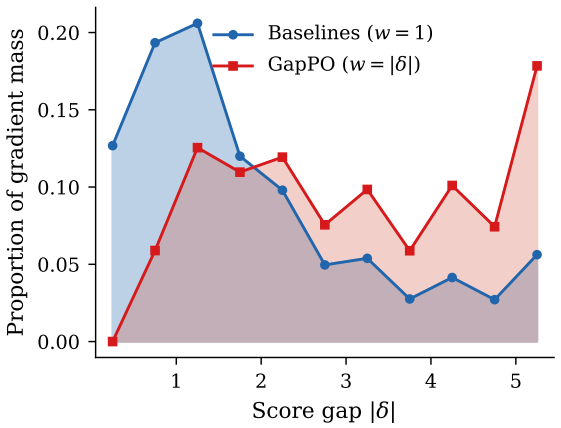

Figure 1: Fraction of total gradient mass per $|\delta|$ bin on UltraFeedback training pairs. Baselines apply uniform per-pair weights ($w = 1$), so their distribution mirrors the raw dataset histogram. GapPO ($w = |\delta|$) actively shifts mass toward high-confidence pairs.

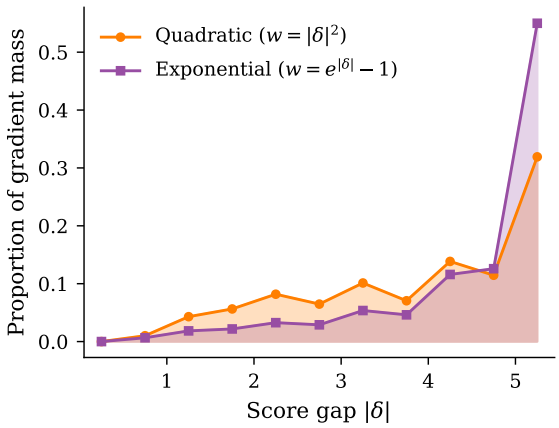

Figure 2: Gradient mass allocation under superlinear weighting functions. Quadratic ($w = |\delta|^2$) and exponential ($w = e^{|\delta|} - 1$) concentrate mass so heavily at the extreme high-$|\delta|$ tail that moderate-$|\delta|$ pairs contribute negligible gradient.

equal, $w^{(i)}$ is constant and GapPO reduces to the baseline SimPO function. The weight $w^{(i)}$ appears *outside* the sigmoid, acting as a per-sample loss multiplier that rescales the gradient contribution of each pair without changing the functional form of the pairwise comparison.

Figure 1 illustrates the effect empirically. Baseline methods (CPO, IPO, SimPO, AlphaPO) concentrate gradient mass at low-to-moderate $|\delta|$ simply because that is where most UltraFeedback pairs lie. GapPO actively redistributes mass toward the high-$|\delta|$ tail, where annotation labels are most reliable.

Linear scaling is the natural choice among monotone weighting functions. It is the unique weight proportional to annotation confidence. A pair with $|\delta| = 4$ contributes exactly twice as much gradient as one with $|\delta| = 2$, directly reflecting how much more reliable the annotation is. Superlinear alternatives such as quadratic ($w = |\delta|^2$) or exponential ($w = e^{|\delta|} - 1$) concentrate mass so aggressively on the extreme high-$|\delta|$ tail that moderate-$|\delta|$ pairs contribute almost nothing, which defeats the stated purpose of soft weighting over hard filtering. Figure 2 shows this effect on the UltraFeedback $|\delta|$ distribution directly.

3.2 Boundary Cases

GapPO admits two natural boundary cases. Firstly, when all $\delta^{(i)} = 1$, weights are uniform and $\mathcal{L}_{\text{GapPO}} = \mathcal{L}_{\text{SimPO}}$ exactly. Secondly, when one pair has $\delta^{(i^*)} \gg \delta^{(i)}$ for all others, the gradient is dominated by that pair, recovering the behavior of hard filtering. These properties arise from the data distribution rather than a hyperparameter.

3.3 Unnormalized Weights as a Batch-Adaptive Learning Rate

We use the raw weights $w^{(i)} = \delta^{(i)}$ directly without batch normalization. This allows batches containing many high-$|\delta|$ pairs to produce proportionally larger gradient steps, creating a form of adaptive learning rate that provides batch informativeness. This is preferable to normalizing within each batch, which would reduce the differential and make the effective update magnitude insensitive to how informative the batch actually is.

3.4 Continuous Soft Weighting vs. Hard Pair Filtering

A natural alternative to GapPO's continuous weighting is hard filtering: discard all pairs with $\delta^{(i)} < \tau$ and train on the remainder with uniform weights. This is equivalent to a step-function weight $w^{(i)} = \mathbf{1}[\delta^{(i)} \geq \tau]$.

Hard filtering has two disadvantages compared to GapPO:

1. It applies a binary threshold, so all retained pairs receive equal gradient weight regardless of how large their score gap is. A pair with $\delta = 2.1$ is treated identically to one with $\delta = 4.8$. GapPO continues to differentiate within the retained set.

2. The threshold is a discrete hyperparameter that must be tuned, whereas GapPO's continuous weight adapts naturally to the full distribution of $|\delta|$ in the training data without requiring a cutoff decision.

We compare GapPO (continuous weighting) against a hard-filtered baseline (`min_delta`$= \tau$, uniform weights on retained pairs) directly in our experiments, using the same τ to ensure a fair comparison of filtering vs. continuous weighting strategies.

3.5 MLP-Only Fine-Tuning as a Parameter-Efficient Ranking Regime

We adopt a parameter-efficient strategy that freezes all attention sublayers, layer-norm parameters, and embedding matrices, training only the MLP (feed-forward network) sublayers. For the models in this work, this amounts to approximately 63.5% of total parameters. Unlike LoRA (Hu et al., 2022), MLP-only tuning modifies weights directly and requires no adapter merging at inference time. It is motivated by the observation that feed-forward layers store factual knowledge (Geva et al., 2021) and dominate representational change needed for preference adaptation. We contrast MLP-only tuning directly against full model fine-tuning across all methods, both as a more compute-efficient training regime and to assess whether the ranking gains from delta weighting persist even when the majority of the model's parameters are held fixed.

3.6 Training Stability and Collapse Mitigation

Length-normalized objectives are susceptible to two failure modes that must be addressed at the methodology level. The first is a sharp crash triggered by fully-truncated sequences: when a prompt is long enough to consume the entire 1024-token budget, the response has zero valid tokens after tokenization, causing length normalization to divide by zero and produce NaN losses that corrupt gradients. We address this by clamping the response length denominator to a minimum of 1, so fully-truncated responses receive a reward of 0 rather than propagating NaN.

The second is a gradual collapse driven by length-label correlation: chosen responses tend to be longer, and the model can exploit this by assigning higher length-normalized reward to stylistic patterns of chosen outputs rather than learning genuine quality distinctions, causing per-token log-probabilities to inflate and the reward to lose discriminative power.

GapPO's delta weighting mitigates the second failure mode by design: high-$|\delta|$ pairs carry genuine quality differences that provide a corrective signal early in training, extending the window before the length-correlation collapse dominates. Additional mitigation strategies applied across all methods include: (1) clamping the length denominator to $\min = 1$; (2) 1-epoch training to stop before collapse; (3) cosine LR schedule with 10% warmup; (4) conservative learning rates; and (5) `load_best_model_at_end=True` to save the peak-PWA checkpoint rather than the final one.

4 Experiments

4.1 Training Details

All experiments use the **UltraFeedback binarized** dataset (Cui et al., 2023), which provides explicit scalar quality scores (`score_chosen` and `score_rejected`) for each preference pair, making δ directly available without additional preprocessing. Scores are continuous averages over multiple annotation criteria, yielding $|\delta| \in [0, 5.5]$.

We evaluate on four pretrained models spanning a range of scales, architectures, and training lineages: Qwen2.5-0.5B, Gemma-2-2B, and Mistral-7B. The original SimPO (Meng et al., 2024) and AlphaPO (Gupta et al., 2025) evaluations both use Llama-3-8B, Mistral-7B, and Gemma-2-9B; we include Mistral-7B for direct comparison, substitute Gemma-2-2B as a smaller representative of the Gemma-2 family, and add Qwen2.5-0.5B as an SLM to test whether GapPO's gradient concentration on high-$|\delta|$ pairs yields greater benefit at small scale. Each model is evaluated under both MLP-only and full fine-tuning regimes.

We compare GapPO against four baselines: **CPO** (Xu et al., 2024) (reference-free hinge loss), **IPO** (Azar et al., 2024) (squared loss), **SimPO** (Meng et al., 2024) (length-normalized, uniform pair weights), and **AlphaPO** (Gupta et al., 2025) (α-shaped reward, uniform pair weights, $\alpha = 0.5$). We also include a hard-filter baseline that discards pairs with $|\delta| < 1$ to contrast continuous weighting against hard pair selection.

All methods are implemented within the `CPOTrainer` framework (Xu et al., 2024) so that PWA is computed identically across methods. Learning rates are scaled by model size: 1×10^{-5} for Qwen2.5-0.5B, 8×10^{-6} for Gemma-2-2B, and

5×10^{-6} for Mistral-7B, with a cosine schedule and 10% linear warmup. We train for 1 epoch with batch size 2 per device and 8 gradient accumulation steps (effective batch size 16), maximum sequence length 1024 tokens, $\beta = 5.0$, $\gamma = 0.5$, and random seed 42.

4.2 Evaluation Metrics: Pairwise Accuracy and the Bridge to Listwise Ranking

We evaluate all methods on two complementary metrics. The primary metric is **pairwise accuracy (PWA)**:

$$\text{PWA} = \frac{1}{|\mathcal{D}_{\text{eval}}|} \sum_{(x,y_w,y_l)\in\mathcal{D}_{\text{eval}}} \mathbf{1}[r(y_w|x) > r(y_l|x)], \tag{9}$$

where $r(y|x) = \frac{1}{|y|}\log \pi_\theta(y|x)$ is the token-averaged log-probability under the trained policy, computed without generation by forward-passing fixed responses from the held-out set. The evaluation set is filtered to pairs with $|\delta| \geq 2$, restricting measurement to high-confidence annotations. We report **peak PWA** — the maximum across all checkpoints — since methods that collapse late in training should not be penalized for instability unrelated to their core optimization quality.

We additionally report **Spearman rank correlation** ρ between the model's implicit reward and annotation scores across the full evaluation set. Let $\{(r_i, s_i)\}_{i=1}^{M}$ denote the implicit rewards and annotation scores for all M responses (chosen and rejected pooled) in the eval set. Let $\text{rk}(r_i)$ and $\text{rk}(s_i)$ be their rank positions. Spearman ρ is:

$$\rho = 1 - \frac{6\sum_{i=1}^{M} d_i^2}{M(M^2-1)}, \quad d_i = \text{rk}(r_i) - \text{rk}(s_i). \tag{10}$$

While PWA measures whether individual pairs are correctly ordered, Spearman measures whether the reward is globally calibrated as a continuous scoring function — the property required to scale from pairwise to listwise ranking. A model with high Spearman assigns scores that monotonically track annotation quality across the full quality spectrum, not just their direction, which is the prerequisite for strong listwise NDCG performance. Figure 5 shows Spearman trajectories over training.

4.3 Training Collapse Validation

Section 3.6 describes two failure modes of length-normalized objectives: NaN crashes from zero-length responses and gradual PWA degradation

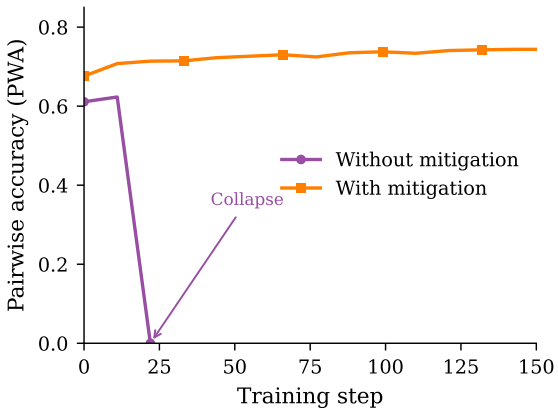

Figure 3: PWA over training steps with and without collapse mitigation on Qwen2.5-0.5B-Instruct.

from length-label correlation. Figure 3 validates the second failure mode empirically. Without collapse mitigation, PWA rises briefly in the first few hundred steps before crashing to near zero as the model degenerates, which confirms that early stopping and best-checkpoint selection are necessary rather than optional. With mitigation strategies applied (length denominator clamping, 1-epoch training, cosine LR schedule, and `load_best_model_at_end`), PWA rises steadily and stabilizes, demonstrating that the failure mode is reproducible and addressable.

4.4 Results

Table 1 reports Spearman Correlation and peak PWA for all methods, models, and fine-tuning regimes. Table 2 reports Spearman correlation and PWA for Qwen2.5-0.5B-Instruct. Figures 4 and 5 show training curves.

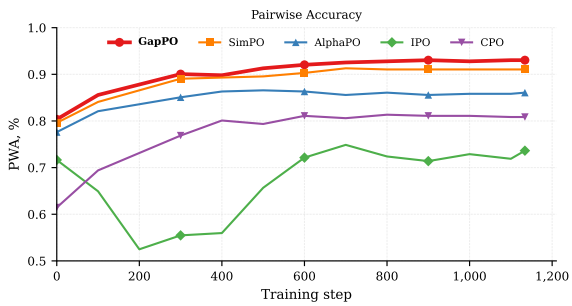

Figure 4: Pairwise accuracy (PWA) over training steps on Gemma-2-2B ($\beta = 5.0$, $\gamma = 0.5$, 1 epoch, cosine LR). All methods are evaluated on the held-out UltraFeedback split every 100 steps.

Score-gap weighting yields large, consistent PWA gains. GapPO improves peak PWA over

Table 1: Spearman Corr (%) and Peak pairwise accuracy (PWA, %) on UltraFeedback binarized eval set. All results use $\beta = 5.0$, $\gamma = 0.5$, $\tau = 2$, $\alpha = 0.5$, 1 epoch, cosine LR.

Model	Fine-tuning	Method	Spearman	Peak PWA
Qwen2.5-0.5B-Instruct	MLP-only	CPO	13.06	60.09
		IPO	16.92	62.66
		SimPO	36.82	78.53
		AlphaPO	29.23	73.08
		GapPO	**38.93**	**81.41**
	Full	CPO	16.92	64.58
		IPO	17.41	65.39
		SimPO	38.18	80.45
		AlphaPO	31.22	74.36
		GapPO	**40.17**	**83.01**
Gemma-2-2B	MLP-only	CPO	29.98	70.50
		IPO	24.38	66.12
		SimPO	46.64	80.91
		AlphaPO	42.04	77.44
		GapPO	**47.76**	**81.74**
	Full	CPO	31.10	70.78
		IPO	30.10	69.22
		SimPO	46.89	81.64
		AlphaPO	42.16	77.63
		GapPO	**49.25**	**82.28**
Mistral-7B-Instruct	MLP-only	CPO	22.92	71.31
		IPO	51.61	80.77
		SimPO	57.66	85.42
		AlphaPO	57.10	84.62
		GapPO	**58.11**	**86.06**
	Full	CPO	11.08	72.70
		IPO	35.70	79.79
		SimPO	49.64	85.52
		AlphaPO	46.76	85.86
		GapPO	**50.87**	**86.34**

Table 2: Peak PWA (%) for GapPO with and without weight normalization ($\beta = 5.0$, $\gamma = 0.5$, 1 epoch). "No Norm" = unnormalized weights; "Weight Norm" = batch-normalized.

Model	Fine-tuning	Variant	PWA (%)
Gemma-2B	MLP	No Norm	**81.71**
		Weight Norm	80.80
	Full	No Norm	**83.26**
		Weight Norm	82.35

the strongest baseline (SimPO) by $+11.28\%$ on Qwen2.5-0.5B MLP-only, $+11.39\%$ on Qwen2.5-0.5B full, $+4.46\%$ on Gemma-2-2B MLP-only, $+4.05\%$ on Gemma-2-2B full, $+0.64\%$ on Mistral-7B MLP-only, and $+0.82\%$ on Mistral-7B full. Gains are largest for small language models (Qwen2.5-0.5B), which is consistent with GapPO's gradient concentration being most consequential when the total model capacity is limited.

Score-gap weighting produces substantially higher Spearman correlation. GapPO improves Spearman over SimPO by $+15.99\%$ on Qwen2.5-0.5B MLP-only, $+15.44\%$ on Qwen2.5-0.5B full, $+6.64\%$ on Gemma-2-2B MLP-only, $+4.09\%$ on Gemma-2-2B full, $+0.45\%$ on Mistral-7B MLP-only, and $+1.23\%$ on Mistral-7B full. Higher Spearman indicates the reward is more globally calibrated across the full quality spectrum, which directly validates the core GapPO claim that weighting by $|\delta|$ shapes the reward to reflect annotation confidence rather than just binary preference direction. This is the calibration property required to scale to listwise NDCG evaluation.

MLP-only fine-tuning retains most of the gain at lower compute cost. Across all models, MLP-only tuning achieves PWA within 0.5–1.0% of full fine-tuning while training only $\sim$36.5% of parameters. On Gemma-2-2B, GapPO MLP-only reaches 85.26% vs. 86.22% for full fine-tuning, which is a 1.0% gap that is small relative to the 4.46% gain over SimPO. This confirms that the rank-

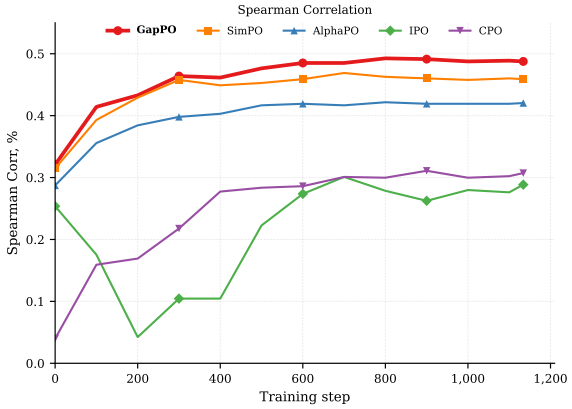

Figure 5: Spearman rank correlation between model reward and annotation scores over training steps on Gemma-2-2B. GapPO's score-based weighting drives higher correlation throughout training, indicating the reward better tracks annotation quality across the full range — the prerequisite for listwise ranking.

ing gains persist under parameter-efficient training. Appendix Table 4 confirms that linear weighting ($w = |\delta|$) outperforms quadratic and exponential alternatives on Qwen2.5-0.5B.

4.5 Why Small Models Benefit Most

PWA gains diminish with model scale: largest at Qwen2.5-0.5B, smaller but consistent at Gemma-2-2B and Mistral-7B. With limited capacity, gradient signal quality matters more since noisy low-$|\delta|$ updates are costlier when there are fewer parameters to absorb them.

4.6 Spearman Gains Validate Reward Calibration

Spearman improvements are disproportionately large relative to PWA gains, especially at small scale. This supports the core GapPO claim that $|\delta|$ weighting shapes reward magnitude, not just its sign. The model develops a globally calibrated reward rather than merely learning to classify unambiguous pairs, which is the prerequisite for listwise NDCG evaluation.

4.7 Consistent Method Performance Ordering

The top three positions are stable across all models and fine-tuning regimes: GapPO first, SimPO second, AlphaPO third. The bottom two swap: IPO $>$ CPO at Qwen2.5-0.5B and Mistral-7B, but CPO $>$ IPO at Gemma-2-2B, reflecting a non-monotonic interaction between architecture and the CPO length-normalization gap rather than a simple scale trend.

The ordering reflects cumulative design choices. CPO lacks length normalization, conflating response length with quality and producing near-chance Spearman in several settings. IPO avoids this via log-probability ratios but its squared-deviation regularizer caps reward dynamic range. AlphaPO's α-concave transformation reduces over-optimization risk while preserving discriminative structure, ranking above CPO and IPO consistently. SimPO adds a target margin γ for further discriminability. GapPO improves over SimPO by weighting each pair by $|\delta|$, concentrating gradient on the most reliable annotations — a signal none of the baselines use.

4.8 Generalization to HelpSteer2-Scores

To verify that score-gap weighting generalizes beyond UltraFeedback, we evaluate on HelpSteer2-Scores (Bai et al., 2022), where $|\delta| \in [1, 19]$ (mean 3.83) — a substantially wider gap distribution than UltraFeedback ($|\delta| \in [0.08, 4.0]$, mean 1.27) and a different annotation scheme (sum of five quality attributes). Full results are reported in Appendix Table 3. GapPO achieves the highest Spearman and PWA in all completed settings across both models and both fine-tuning regimes, confirming that score-gap weighting is a general mechanism rather than one tuned to a specific annotation scale or dataset size.

5 Conclusion

We introduced GapPO (Gradient-Adaptive Pairwise Preference Optimization), a preference optimization method that improves pairwise ranking accuracy in LLMs by weighting each training pair by its annotation score gap. Pairwise accuracy is a direct measure of the model's usefulness as a ranker since a model that correctly orders held-out response pairs is a more reliable backbone for best-of-n sampling, reranking, and downstream RLHF. Large gaps in annotation scores indicate unambiguous comparisons that should drive larger gradients, while near-ties carry noisy labels that should be down-weighted but not discarded. GapPO implements this through linear per-sample weights proportional to $|\delta|$ and unnormalized weighting that allows batch informativeness to modulate the effective learning rate. Across four model families and two fine-tuning regimes, GapPO consistently improves peak PWA and Spearman Correlation over SimPO, CPO, IPO, and AlphaPO baselines.

Limitations

GapPO requires annotation scores with sufficient variance to be effective. On datasets where most pairs have near-identical scores, the weighting degenerates toward SimPO. More broadly, the method requires scalar quality scores: datasets with only binary preference labels (chosen/rejected without scores) cannot directly provide $|\delta|$ weights without additional score estimation. The method also inherits the failure modes of length-normalized objectives, including susceptibility to length-label correlation collapse under extended training, which we address through early stopping and best-checkpoint selection rather than a structural fix. Our evaluation is limited to pairwise accuracy and Spearman correlation on held-out pairs. Listwise NDCG evaluation, which would more directly measure ranking quality across all responses per prompt, is ongoing work. Finally, all experiments use a fixed linear weighting function $w = |\delta|$, which means that the optimality of linear scaling relative to other monotone functions is asserted on theoretical and empirical grounds but not proven for arbitrary data distributions.

Ethical Considerations

GapPO is a preference optimization method intended to improve the ranking accuracy of LLMs when trained on human-annotated preference data. The method does not introduce new data collection pipelines or annotation procedures, and relies entirely on existing scored preference datasets (UltraFeedback, HelpSteer2). As with all preference optimization approaches, the quality and potential biases of underlying annotations directly affect the trained model. If annotators systematically favor certain response styles or topics, score-gap weighting will amplify the gradient from those pairs, potentially reinforcing annotator bias. Practitioners applying GapPO to new datasets should audit inter-annotation agreement, particularly for high-$|\delta|$ pairs that receive the most gradient weight. We do not anticipate dual-use concerns specific to this method beyond those that apply to LLM fine-tuning generally.

References

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Remi Munos. 2024. A general theoretical paradigm to understand learning from human feedback. *arXiv preprint arXiv:2310.12036*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. UltraFeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Aman Gupta, Shao Tang, Qingquan Song, Sirou Zhu, Jiwoo Hong, Ankan Saha, Viral Gupta, Noah Lee, Eunki Kim, Siyu Zhu, Parag Agrawal, Natesh Pillai, and S. Sathiya Keerthi. 2025. AlphaPO: Reward shape matters for LLM alignment. In *Proceedings of the 42nd International Conference on Machine Learning*.

Yupeng Hou, Junjie Zhang, Zihan Lin, Haoran Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, pages 364–381. Springer.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2024. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Shen, Xiaobing Liu, Jiaming Liu, Donald Metzler, Xuanhui Wang, and 1 others. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518.

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

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

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542*.

Junkang Wu and 1 others. 2025. AlphaDPO: Adaptive reward margin for direct preference optimization. In *Proceedings of the 42nd International Conference on Machine Learning*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Prate, Kenton Murray, Kevin Duh, and Matt Post. 2024. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. *arXiv preprint arXiv:2401.08417*.

Zhanhui Zhou, Jie Liu, Chao Yang, Jing Shao, Yu Liu, Xiangyu Yue, Wanli Ouyang, and Yu Qiao. 2023. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. *arXiv preprint arXiv:2310.03708*.

A HelpSteer2-Scores Ablation

Table 3: Spearman Corr (%) and Peak PWA (%) on HelpSteer2-Scores eval set. Scores are the sum of five annotation attributes (helpfulness, correctness, coherence, complexity, verbosity), yielding $|\delta| \in [1, 19]$ with mean 3.83. All results use $\gamma = 0.5$, 1 epoch, cosine LR. Qwen2.5-0.5B uses $\beta = 2.0$; Gemma-2-2B and Mistral-7B use $\beta = 5.0$.

Model	Tuning	Method	Spearman	Peak PWA
Qwen-0.5B	MLP	CPO	0.00	33.33
		IPO	9.51	53.88
		SimPO	12.13	53.89
		AlphaPO	2.30	49.44
		GapPO	**12.79**	**56.67**
	Full	CPO	0.00	33.89
		IPO	10.16	53.33
		SimPO	14.10	56.11
		AlphaPO	2.95	50.56
		GapPO	**15.41**	**57.78**
Gemma-2B	MLP	CPO	1.64	50.82
		IPO	9.51	54.44
		SimPO	25.25	65.00
		AlphaPO	19.34	61.67
		GapPO	**27.21**	**68.33**
	Full	CPO	8.20	54.10
		IPO	10.16	55.08
		SimPO	25.90	62.95
		AlphaPO	20.00	60.00
		GapPO	**30.49**	**65.25**
Mistral-7B	MLP	CPO	10.16	55.08
		IPO	15.41	57.71
		SimPO	37.71	68.85
		AlphaPO	37.05	68.53
		GapPO	**40.33**	**70.16**
	Full	CPO	0.00	46.23
		IPO	14.10	57.05
		SimPO	21.97	60.98
		AlphaPO	20.66	60.33
		GapPO	**25.25**	**62.62**

B Scaling Function Ablation

Table 4: Effect of weighting function on Qwen2.5-0.5B-Instruct (Full fine-tuning, UltraFeedback). Linear ($w = |\delta|$) achieves the best Spearman and PWA; superlinear functions concentrate mass too aggressively on the high-$|\delta|$ tail.

Weighting	Formula	Spearman (%)	PWA (%)
Linear	$w = \lvert\delta\rvert$	**40.17**	**83.01**
Quadratic	$w = \lvert\delta\rvert^2$	39.06	82.05
Exponential	$w = e^{\lvert\delta\rvert} - 1$	40.17	81.89

C Computational Budget

Table 5: Model sizes, trainable parameters under MLP-only fine-tuning (36.5% of total), and observed wall-clock hours per run on a single NVIDIA A100-SXM4-80GB GPU (Harvard FASRC cluster).

Model	Params	MLP	Full (h)	MLP (h)
Qwen-0.5B	0.49B	0.18B	$\sim$1.0	$\sim$0.5
Gemma-2B	2.61B	0.95B	$\sim$2.2	$\sim$1.5
Mistral-7B	7.24B	2.64B	$\sim$5.1	$\sim$4.9

Across all reported experiments (UltraFeedback main table, HelpSteer2 ablation, normalization ablation, and scaling ablation), we ran 35 full fine-tuning runs and 32 MLP-only runs for a total of approximately **160 GPU-hours** on NVIDIA A100-SXM4-80GB GPUs via the Harvard FASRC cluster.