Learning Temporally-Aware Sample Weights for Preference Optimization

Mengyang Li[1], Xudong Zhou[1], Pinlong Zhao[2]*
[1]Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University
[2]School of Cyberspace, Hangzhou Dianzi University
limengyang@tjnu.edu.cn, zhouxudong@stu.tjnu.edu.cn, pinlongzhao@hdu.edu.cn

Abstract

Preference optimization is fundamental for aligning large language models. While existing methods use sample weighting, they typically rely on static functions of instantaneous model states and ignore temporal learning dynamics. We contend that a sample's value evolves throughout training, characterized by patterns such as stable convergence or noisy oscillation. We propose MetaPO, a framework that meta-learns adaptive weights using three temporal features: reward margin evolution, learning volatility, and reference deviation. Through bilevel optimization on validation data, MetaPO automatically discovers weighting strategies tailored to specific datasets. Experiments on models ranging from 7B to 70B parameters demonstrate statistically significant improvements over strong baselines, achieving gains of up to 2.4 points on AlpacaEval 2.0 and 1.6 points on Arena-Hard. Interpretability analysis confirms that temporal features drive over 70% of the weighting decisions and that the learned weights correlate strongly with sample quality.

1 Introduction

Aligning large language models (LLMs) with human preferences through reinforcement learning from human feedback (RLHF) has become fundamental to developing capable AI systems (Christiano et al., 2017; Ouyang et al., 2022; Bai et al., 2022). Direct Preference Optimization (DPO) (Rafailov et al., 2023) simplifies this pipeline by reformulating RLHF as supervised learning, eliminating explicit reward modeling and complex policy gradient algorithms. DPO's efficiency has inspired numerous variants exploring different optimization objectives and architectural choices (Meng et al., 2024; Ethayarajh et al., 2024; Hong et al., 2024; Azar et al., 2024; Tang et al., 2024).

A critical yet underexplored aspect of preference optimization is sample weighting: determining each preference pair's contribution to the training objective. Recent work demonstrates that appropriate weighting schemes significantly impact alignment quality. FocalPO (Liu et al., 2025) adapts focal loss from computer vision (Lin et al., 2020), using a modulating factor p^γ where p is the preference probability and γ is a fixed hyperparameter. While achieving strong empirical results, this weighting function shares a fundamental limitation with prior approaches: it depends solely on the current model state at each training step.

Consider two preference pairs during training, both currently assigned preference probability $p_t = 0.8$, indicating correct ranking. However, their learning histories differ dramatically. Sample A has maintained consistent correctness throughout training with minimal fluctuation, suggesting stable learning. Sample B has oscillated wildly between $p \approx 0.2$ and $p \approx 0.9$, exhibiting high volatility that may signal annotation noise or fundamental ambiguity. Should these samples receive identical weights? Existing methods answer yes, as they observe only instantaneous probability p_t. We argue this is suboptimal: Sample B's erratic trajectory warrants reduced influence on learning.

This motivates our central thesis: sample weighting for preference optimization can benefit from being temporally aware. A training sample's value is not necessarily a static property determined by current predictions, but may evolve based on its learning trajectory. We identify three dimensions that characterize this temporal evolution: (1) Reward margin dynamics: how the model's preference strength has evolved, distinguishing newly learned from long-established knowledge; (2) Learning volatility: whether the sample exhibits consistent signals or noisy oscillations; (3) Reference model alignment: the degree to which learned preferences deviate from prior knowledge, potentially indicat-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 12361–12377

ing annotation errors or distribution shift.

We propose MetaPO (Meta-learned Preference Optimization), a framework that meta-learns adaptive sample weights based on these temporal features. For each preference pair, we construct a compact feature representation encoding both current values and exponential moving averages of the three signals above. A lightweight linear meta-network maps these features to sample-specific weights, with parameters learned through bilevel optimization on validation data. Our meta-objective balances validation performance with margin preservation on correctly-ranked samples, discovering weighting strategies that improve generalization while maintaining stable learning dynamics.

Our work makes three primary contributions:

(1) Conceptual contribution: We identify temporal dynamics as critical for preference optimization and formalize three principled temporal features grounded in the learning process: margin evolution, volatility, and reference alignment.

(2) Methodological contribution: We develop MetaPO, a simple and interpretable meta-learning framework with a linear weighting function enabling direct analysis of learned strategies while maintaining computational efficiency.

(3) Empirical contribution: We demonstrate consistent improvements over competitive baselines across 7B to 70B parameters on established benchmarks, with statistical significance confirmed through multiple runs ($p < 0.05$). SHAP analysis and systematic case studies reveal that learned weights correlate strongly with sample quality ($\rho = 0.58$, $p < 0.001$), with temporal components being the dominant factor in weighting decisions.

2 Related Work

Preference Optimization. Direct Preference Optimization (Rafailov et al., 2023) eliminates the reward modeling phase of traditional RLHF (Christiano et al., 2017; Ouyang et al., 2022; Ryan et al., 2022) by directly optimizing policy parameters using preference data. Subsequent work has explored various modifications: SimPO (Meng et al., 2024) removes the reference model constraint and incorporates length normalization; ORPO (Hong et al., 2024) combines supervised fine-tuning with preference learning in a single objective; KTO (Ethayarajh et al., 2024) applies prospect theory to optimize preferences without paired comparisons. Recent theoretical work (Azar et al., 2024; Tang et al., 2024) has analyzed the convergence properties and sample complexity of preference-based learning. Identity-PO (Azar et al., 2024) provides a unified framework for understanding various preference optimization algorithms, while (Tang et al., 2024) extends DPO to general reward functions beyond the Bradley-Terry model.

Most relevant to our work, FocalPO (Liu et al., 2025) introduces sample weighting inspired by focal loss (Lin et al., 2020), using p^γ to modulate each preference pair's contribution. Concurrent work (Li et al., 2026) explores meta-learning for fusing intrinsic feedback in preference alignment. Our approach differs fundamentally: rather than using a fixed, hand-crafted function applied uniformly, we employ meta-learning to discover adaptive, sample-specific weights informed by temporal learning dynamics.

Sample Reweighting for Preference Optimization. Several concurrent works explore learned reweighting within the alignment pipeline. DORM (Zhang et al., 2025) applies bilevel optimization to learn sample weights for reward modeling based on uncertainty signals. RDO (Wang et al., 2024b) introduces closed-form reweighting coefficients for DPO derived from reward difference signals. BPO (Wang et al., 2025) addresses balanced preference optimization across knowledge breadth and depth through multi-objective sample utilization. These methods share MetaPO's motivation of differentiating sample contributions, yet they all derive importance from instantaneous model states at a single checkpoint. None tracks how the learning signal evolves over time. MetaPO complements these approaches by introducing temporal dynamics as an orthogonal axis for sample characterization.

Meta-Learning for Data Optimization. Meta-learning has been successfully applied to data selection and weighting in supervised learning. L2RW (Ren et al., 2018) learns sample weights through bilevel optimization on validation data. Meta-Weight-Net (Shu et al., 2019) uses a neural network to predict importance weights based on gradient direction and training loss. MentorNet (Jiang et al., 2018) learns a data-driven curriculum by training a student-teacher network. Training dynamics analysis (Li et al., 2025) and difficulty-aware learning curve extrapolation (Li and Zhao, 2026) demonstrate that temporal trajectories carry rich signals beyond instantaneous model states, motivating our

temporal feature design.

3 Method

Figure 1 provides an overview of our approach. We begin by establishing the problem setup and identifying limitations of existing methods, then introduce our temporal feature design, meta-weighting architecture, and optimization procedure.

3.1 Background and Motivation

Given a dataset of preference pairs $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$ where x denotes a prompt and $y_w \succ y_l$ indicates human preference, DPO optimizes:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(\beta \Delta r_\theta)\right] \quad (1)$$

where $\Delta r_\theta = r_\theta(y_w) - r_\theta(y_l)$ represents the reward margin, with implicit rewards defined as $r_\theta(y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$.

FocalPO modifies this by introducing a sample-dependent modulating factor:

$$\mathcal{L}_{\text{FocalPO}}(\theta) = -\mathbb{E}\left[p^\gamma \cdot \log \sigma(\beta \Delta r_\theta)\right] \quad (2)$$

where $p = \sigma(\beta \Delta r_\theta)$ and $\gamma \geq 0$ is a hyperparameter. While this formulation successfully upweights correctly-ranked samples, the weight p_t^γ at training step t depends exclusively on the current preference probability, unable to distinguish samples with different learning histories despite identical current states.

3.2 Temporal Feature Design

We characterize each sample through three scalar quantities, each augmented with temporal information via exponential moving averages (EMA). This design addresses three fundamental questions about sample quality in preference learning: Has this sample been consistently learned? Is the learning stable or noisy? Does it align with prior knowledge?

Reward Margin Evolution. The instantaneous reward margin $\Delta r_\theta^{(t)} = r_\theta^{(t)}(y_w) - r_\theta^{(t)}(y_l)$ indicates current ranking correctness. To capture historical trends, we maintain:

$$f_1^{(t)} = \Delta r_\theta^{(t)} \quad (3)$$

$$\bar{f}_1^{(t)} = \alpha \bar{f}_1^{(t-1)} + (1-\alpha) f_1^{(t)} \quad (4)$$

with decay rate $\alpha = 0.9$. The combination $[f_1^{(t)}, \bar{f}_1^{(t)}]$ distinguishes qualitatively different scenarios: persistent correctness ($f_1 > 0, \bar{f}_1 > 0$), recent transitions ($f_1 > 0, \bar{f}_1 \approx 0$), and consistent errors ($f_1 < 0, \bar{f}_1 < 0$).

Learning Volatility. We quantify learning stability through the running standard deviation of margin values over the entire training history of each sample:

$$f_2^{(t)} = \sqrt{\frac{1}{t}\sum_{\tau=1}^{t}\left(\Delta r^{(\tau)} - \overline{\Delta r}^{(t)}\right)^2} \quad (5)$$

$$\bar{f}_2^{(t)} = \alpha \bar{f}_2^{(t-1)} + (1-\alpha) f_2^{(t)} \quad (6)$$

where $\overline{\Delta r}^{(t)}$ is the running mean of margin values. This is computed efficiently using Welford's online algorithm, maintaining only three scalars per sample (M_2, mean, count) without storing the full margin history. High values of f_2 indicate unstable learning across the trajectory, while elevated $\bar{f}_2$ suggests persistent instability, a strong signal of potential annotation noise or fundamental sample ambiguity. Temporal smoothing through the EMA (with $\alpha = 0.9$, corresponding to an effective half-life of approximately 7 steps) provides complementary information: f_2 captures cumulative volatility over the full trajectory, while $\bar{f}_2$ reflects recent stability trends.

Reference Model Deviation. The reference model π_{ref} encodes knowledge from supervised fine-tuning. We measure divergence via KL:

$$f_3^{(t)} = D_{\text{KL}}(\pi_\theta^{(t)}(\cdot|x) \| \pi_{\text{ref}}(\cdot|x)) \quad (7)$$

$$\bar{f}_3^{(t)} = \alpha \bar{f}_3^{(t-1)} + (1-\alpha) f_3^{(t)} \quad (8)$$

Samples with sustained high deviation may contradict prior knowledge, potentially indicating labeling errors or systematic distribution shift.

Feature Representation. For sample i at step t, the complete feature vector is:

$$\mathbf{s}_i^{(t)} = [f_1^{(t)}, \bar{f}_1^{(t)}, f_2^{(t)}, \bar{f}_2^{(t)}, f_3^{(t)}, \bar{f}_3^{(t)}] \in \mathbb{R}^6 \quad (9)$$

3.3 Meta-Weight Network

We employ a deliberately simple architecture: a single linear transformation followed by Softplus activation:

$$w_\phi(\mathbf{s}_i) = \text{Softplus}(\mathbf{w}^\top \mathbf{s}_i + b) \quad (10)$$

where $\phi = \{\mathbf{w} \in \mathbb{R}^6, b \in \mathbb{R}\}$ comprises the learnable parameters.

This design prioritizes interpretability and computational efficiency. The linear form allows direct inspection of learned coefficients $\mathbf{w}$ to understand

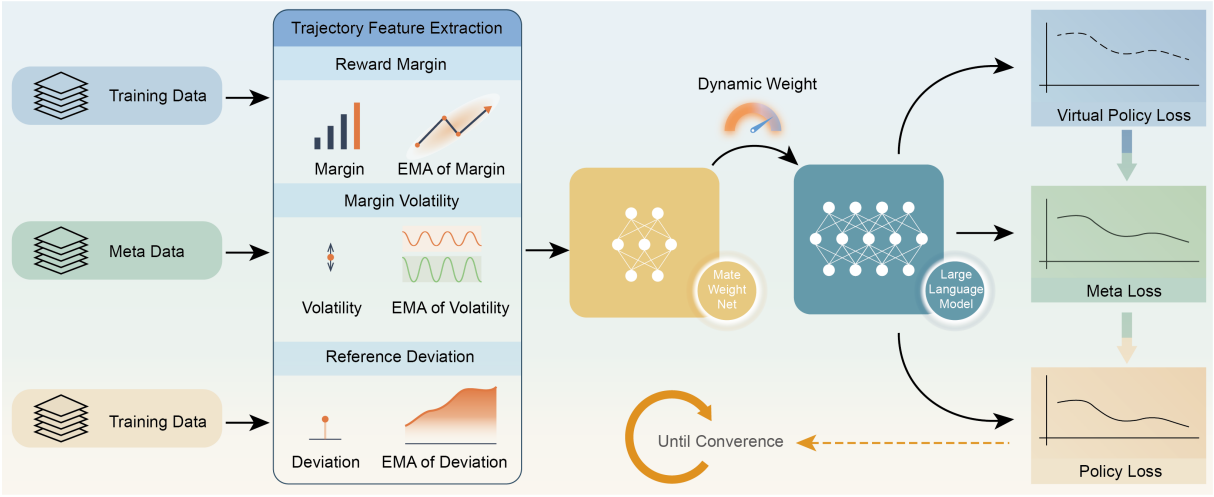

Figure 1: Overview of MetaPO. For each preference pair, we extract three temporal features (margin evolution, volatility, reference deviation) combining current values with exponential moving averages. A linear meta-network predicts sample-specific weights for the DPO objective. Meta-parameters are updated periodically (every K steps) via bilevel optimization on validation data to maximize performance while preserving learned margins.

which features drive weighting decisions (analyzed in Section 4.9). Despite its simplicity, this architecture proves sufficient: our temporal features encode rich information about sample quality, and empirical results demonstrate that linear combination achieves strong performance. The Softplus activation ensures non-negative weights while maintaining smooth gradients for meta-optimization.

3.4 Meta-Objective and Bilevel Optimization

Meta-parameters ϕ are learned through bilevel optimization on a held-out validation set $\mathcal{D}_{\text{val}}$. Our meta-objective balances two complementary goals:

$$\mathcal{L}_{\text{meta}}(\phi) = \mathcal{L}_{\text{val}}(\phi) + \lambda \mathcal{L}_{\text{margin}}(\phi) \tag{11}$$

The validation loss term:

$$\mathcal{L}_{\text{val}}(\phi) = -\frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(x,y_w,y_l)\in\mathcal{D}_{\text{val}}} \log \sigma(\beta \Delta r_\theta) \tag{12}$$

ensures learned weights improve generalization performance. The margin preservation term:

$$\mathcal{L}_{\text{margin}}(\phi) = -\frac{1}{|\mathcal{D}^+_{\text{val}}|} \sum_{i\in\mathcal{D}^+_{\text{val}}} \Delta r_\theta^{(i)} \tag{13}$$

where $\mathcal{D}^+_{\text{val}} = \{i : \Delta r_\theta^{(i)} > 0\}$ denotes correctly-ranked samples, explicitly reinforces correct preferences. This term encourages the meta-learner to discover strategies that maintain or expand margins on accurately-learned samples, reducing catastrophic forgetting.

3.5 Training Algorithm

MetaPO alternates between inner-loop policy updates and outer-loop meta-updates. We employ sparse meta-optimization: rather than updating ϕ at every training step, we perform meta-updates every K steps ($K = 100$ in our experiments). This design is motivated by two factors: (1) the meta-network is lightweight (only 7 parameters), requiring infrequent updates to learn stable weighting strategies; (2) sparse updates substantially reduce computational overhead, as meta-optimization involves backpropagation through the inner loop's gradient computations. Appendix A.2 provides ablation studies demonstrating that $K = 100$ achieves an effective balance between meta-learning quality and computational efficiency. The complete procedure is formalized in Algorithm 1.

Computational Complexity. The overhead introduced by MetaPO is minimal. Space complexity is $O(N)$ for storing temporal statistics (three EMAs plus variance state per sample). Time complexity per sample is $O(1)$ for EMA updates and $O(6)$ for weight prediction (single linear layer forward pass). Meta-update cost is amortized to once every K steps, adding less than 5% to total training time in practice. The meta-network itself contains only 7 parameters, negligible compared to LLM parameter counts.

Algorithm 1 MetaPO Training

Input: $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, π_{ref}, learning rates α_{inner}, α_{meta}, meta-update interval K
Initialize: Policy parameters θ, meta-parameters ϕ, temporal trackers for each sample
step $\leftarrow 0$
for epoch $= 1, \ldots, N$ **do**
  for batch $B \sim \mathcal{D}_{\text{train}}$ **do**
    step $\leftarrow$ step $+ 1$
    for $(x, y_w, y_l) \in B$ **do**
      Compute f_1, f_2, f_3 via Eqs. (3)-(7)
      Retrieve $\bar{f}_1, \bar{f}_2, \bar{f}_3$ from temporal trackers
      Form $\mathbf{s} = [f_1, \bar{f}_1, f_2, \bar{f}_2, f_3, \bar{f}_3]$
      Update trackers via Eqs. (4)-(8)
    end for
    Compute weights: $\{w_i = w_\phi(\mathbf{s}_i)\}_{i \in B}$
    $\mathcal{L}_{\text{train}} \leftarrow -\frac{1}{|B|} \sum_{i \in B} w_i \log \sigma(\beta \Delta r_\theta^{(i)})$
    $\theta \leftarrow \theta - \alpha_{\text{inner}} \nabla_\theta \mathcal{L}_{\text{train}}$
    if step $\bmod K = 0$ **then**
      Compute $\mathcal{L}_{\text{meta}}(\phi)$ on $\mathcal{D}_{\text{val}}$
      $\phi \leftarrow \phi - \alpha_{\text{meta}} \nabla_\phi \mathcal{L}_{\text{meta}}$
    end if
  end for
end for
Return: π_θ

3.6 Theoretical Connections

Our approach connects to several established frameworks in machine learning.

Influence Functions. The learned weights $w_\phi(\mathbf{s}_i)$ can be interpreted as approximating the influence (Koh and Liang, 2017) of sample i on validation performance. Under bilevel optimization, the optimal meta-parameters satisfy $\nabla_\phi \mathcal{L}_{\text{val}}(\theta^*(\phi)) = 0$. By the implicit function theorem, the gradient of validation loss with respect to sample weight w_i can be expressed as:

$$\frac{\partial \mathcal{L}_{\text{val}}}{\partial w_i} = -\nabla_\theta \mathcal{L}_{\text{val}}^\top H_\theta^{-1} \nabla_\theta \ell_i \tag{14}$$

where $H_\theta = \nabla_\theta^2 \mathcal{L}_{\text{train}}$ is the Hessian and ℓ_i is the per-sample loss. This is exactly the influence function formulation of Koh and Liang (2017). Our bilevel optimization with a linear meta-network provides a first-order approximation to this quantity without explicit Hessian computation. By parameterizing the influence through temporal features, we amortize the cost across all samples rather than computing per-sample influence independently. We note that this connection is approximate and relies on standard smoothness assumptions that may not strictly hold in LLM training. Establishing tighter theoretical guarantees through PAC-Bayes or stability-based analysis remains an important direction for future work.

Adaptive Weighting. Unlike focal loss variants that use fixed modulating functions, MetaPO learns a data-adaptive weighting strategy. This provides flexibility: instead of manually tuning hyperparameters for each task, the meta-learner automatically discovers appropriate weightings tailored to the specific data distribution and model.

Implicit Curriculum. The temporal features, particularly margin evolution, enable implicit curriculum learning. Samples transitioning from incorrect to correct (indicated by $f_1 > 0, \bar{f}_1 < 0$) may receive different weights than persistently correct samples, allowing dynamic adjustment of training focus as learning progresses, a form of data-driven curriculum discovered through meta-learning.

4 Experiments

4.1 Experimental Setup

Models and Datasets. We evaluate MetaPO across three model scales: Mistral-7B-SFT (Tunstall et al., 2023), Llama-3-8B-Instruct, and Llama-3-70B-Instruct (Grattafiori et al., 2024). For training data, we use UltraFeedback (Cui et al., 2023) (64k samples) for Mistral, Llama3-UltraFeedback-ArmoRM (Wang et al., 2024a) (61k samples) for Llama-3 models, and HH-RLHF (Bai et al., 2022) (161k samples) to demonstrate scalability. Evaluation employs AlpacaEval 2.0 (Dubois et al., 2024) (805 instructions compared against GPT-4-Turbo responses) and Arena-Hard (Li et al., 2024) (500 challenging prompts compared against GPT-4-0314 responses).

Baselines. We compare against DPO (Rafailov et al., 2023), SimPO (Meng et al., 2024), FocalPO (Liu et al., 2025), ORPO (Hong et al., 2024), and KTO (Ethayarajh et al., 2024). For fair comparison, we re-implement FocalPO and SimPO at all scales using identical experimental settings (hardware, library versions, hyperparameters). Results for ORPO and KTO at 7B/8B scales are taken from (Liu et al., 2025) for reference.

Hyperparameters. For MetaPO, we fix: EMA decay $\alpha = 0.9$, margin loss weight $\lambda = 0.1$, meta-update interval $K = 100$, and meta learning rate $\alpha_{\text{meta}} = 1 \times 10^{-4}$. All other hyperparameters (batch size, inner learning rate, DPO temperature β) follow the settings from (Liu et al., 2025). We maintain these hyperparameters across all experiments without task-specific tuning.

Method	Model	WR	LCWR	Avg
Mistral-7B-SFT				
ORPO†	Mistral-7B	12.6	14.7	13.7
KTO†	Mistral-7B	12.3	14.9	13.6
DPO	Mistral-7B	18.6±0.3	20.6±0.4	19.6
SimPO	Mistral-7B	21.4±0.5	17.0±0.6	19.2
FocalPO	Mistral-7B	20.4±0.4	23.9±0.5	22.2
MetaPO	Mistral-7B	**22.8±0.4**	**25.3±0.3**	**24.1**
Llama-3-8B-Instruct				
ORPO†	Llama-3-8B	33.8	38.1	36.0
KTO†	Llama-3-8B	31.8	33.1	32.5
DPO	Llama-3-8B	47.5±0.6	48.2±0.7	47.9
SimPO	Llama-3-8B	47.5±0.5	53.7±0.6	50.6
FocalPO	Llama-3-8B	49.8±0.5	54.7±0.4	52.3
MetaPO	Llama-3-8B	**51.6±0.4**	**56.9±0.5**	**54.3**
Llama-3-70B-Instruct				
DPO	Llama-3-70B	52.3±0.5	55.8±0.6	54.1
SimPO	Llama-3-70B	53.1±0.4	57.3±0.5	55.2
FocalPO	Llama-3-70B	54.2±0.3	58.6±0.4	56.4
MetaPO	Llama-3-70B	**55.9±0.3**	**60.4±0.4**	**58.2**

Table 1: AlpacaEval 2.0 results (mean±std). MetaPO consistently achieves the highest performance across all model scales. †Results from (Liu et al., 2025).

Method	Model	Win Rate
Mistral-7B-SFT		
ORPO†	Mistral-7B	6.2
KTO†	Mistral-7B	8.8
DPO	Mistral-7B	16.4±0.5
SimPO	Mistral-7B	13.3±0.6
FocalPO	Mistral-7B	17.1±0.4
MetaPO	Mistral-7B	**18.5±0.3**
Llama-3-8B-Instruct		
ORPO†	Llama-3-8B	26.0
KTO†	Llama-3-8B	11.7
DPO	Llama-3-8B	33.1±0.7
SimPO	Llama-3-8B	33.8±0.6
FocalPO	Llama-3-8B	34.6±0.5
MetaPO	Llama-3-8B	**36.2±0.4**
Llama-3-70B-Instruct		
DPO	Llama-3-70B	41.2±0.6
SimPO	Llama-3-70B	42.5±0.5
FocalPO	Llama-3-70B	43.8±0.4
MetaPO	Llama-3-70B	**45.1±0.3**

Table 2: Arena-Hard win rates (mean±std). MetaPO demonstrates strong gains on this challenging benchmark. †Results from (Liu et al., 2025).

Implementation Details. Our implementation builds on the TRL library (von Werra et al., 2020) with DeepSpeed ZeRO-3 (Rasley et al., 2020). All experiments run on 8× NVIDIA A40 GPUs (48GB memory each). Training duration is approximately 6 hours for 7B models and 48 hours for 70B models. We use stratified random sampling to construct validation sets (3% of training data), ensuring distribution alignment across difficulty levels measured by initial DPO loss.

4.2 Main Results

Tables 1 and 2 detail performance on AlpacaEval 2.0 and Arena-Hard. MetaPO consistently outperforms baselines across all model scales and benchmarks. We report the mean and standard deviation over three runs using different seeds, with statistical significance verified via paired t-tests ($p < 0.05$ for all improvements over FocalPO).

The results highlight three key trends. First, MetaPO surpasses FocalPO on AlpacaEval 2.0 by margins of 1.8-2.4 points on WR and 1.4-2.2 points on LCWR. Importantly, MetaPO improves both metrics simultaneously, whereas baselines like SimPO often exhibit trade-offs between raw win rates and length penalties. This indicates that our method drives genuine alignment quality rather than exploiting reward model biases towards longer responses. Second, gains increase with scale; the 70B model achieves a substantial +1.3 point boost on Arena-Hard, indicating that temporal features provide greater value as model capacity and dynamic complexity grow. Third, the performance advantage persists on the rigorous Arena-Hard benchmark across all scales, confirming that temporal awareness generalizes effectively regardless of task difficulty or model architecture.

4.3 Scalability to Large Datasets

We validate scalability using HH-RLHF (Bai et al., 2022), which contains 161k preference pairs (2.5× larger than UltraFeedback), as shown in Table 5. The consistent improvements demonstrate that temporal feature extraction remains effective at scale. The memory overhead is negligible, requiring only 6.1 MB to store statistics for 161k samples. Furthermore, the meta-learning framework adapts to the new distribution without requiring manual hyperparameter tuning. Given that HH-RLHF typically contains more ambiguous labels than UltraFeedback, the sustained performance gap suggests that MetaPO's volatility-based filtering becomes increasingly vital when training on larger, noisier datasets, effectively isolating valid learning signals from annotation errors.

Method	AlpacaEval	Arena-Hard
DPO	47.5±0.6	33.1±0.7
RDO	49.1±0.5	34.0±0.5
FocalPO	49.8±0.5	34.6±0.5
MetaPO	**51.6±0.4**	**36.2±0.4**

Table 3: Comparison with RDO on Llama-3-8B. MetaPO's temporal features outperform instantaneous reward difference signals.

4.4 Ablation Studies

Table 6 analyzes the impact of specific feature groups. The exclusion of volatility features results in the largest single-group decline (2.9 points on AlpacaEval, 3.1 on Arena-Hard), suggesting that detecting unstable learning dynamics is the primary mechanism for improvement. Furthermore, removing all temporal components leads to a drop of over 3.5 points, indicating that historical context provides signals that instantaneous observations cannot capture.

We also evaluate the meta-objective design. Comparing the full objective $\mathcal{L}_{\text{val}}+\lambda\mathcal{L}_{\text{margin}}$ against using $\mathcal{L}_{\text{val}}$ alone reveals that the margin preservation term improves performance by 1.3 points (51.6 vs. 50.3 on AlpacaEval 2.0). This gain suggests that explicitly maintaining margins on correctly ranked samples helps prevent catastrophic forgetting during the meta-learning process. Additional ablation experiments on higher-order temporal features (Appendix A.10) and carefully tuned nonlinear meta-learners (Appendix A.11) confirm that the base 6-dimensional feature set and the linear architecture are sufficient.

4.5 Comparison with Reweighting Baselines

We compare MetaPO against RDO (Wang et al., 2024b), a recent method that derives closed-form reweighting coefficients from reward difference signals. As shown in Table 3, MetaPO outperforms RDO by 2.5 points on AlpacaEval and 2.2 on Arena-Hard, demonstrating the advantage of temporal dynamics over instantaneous reward differences. Direct comparison with DORM (Zhang et al., 2025) is infeasible as it targets reward modeling rather than policy optimization. BPO (Wang et al., 2025) requires explicit knowledge categorization annotations not available in our datasets. We also validated that MetaPO is complementary to online DPO methods: combining MetaPO with iterative DPO yields a +2.6 point gain over standalone iterative DPO (Appendix A.12).

Category	DPO	FocalPO	MetaPO
Writing	8.30	8.55	**8.80**
Roleplay	7.75	7.95	**8.20**
Reasoning	5.95	6.15	**6.30**
Math	5.45	5.55	**5.65**
Coding	6.45	6.60	**6.75**
Extraction	7.80	8.00	**8.10**
STEM	8.35	8.50	**8.60**
Humanities	9.30	9.35	9.30
Overall	7.42	7.58	**7.71**

Table 4: MT-Bench per-category results on Llama-3-8B. MetaPO improves most on preference-sensitive tasks while maintaining performance on knowledge-heavy categories.

Dataset	DPO	FocalPO	MetaPO
UltraFeedback (64k)	48.2±0.6	50.1±0.5	**51.8±0.4**
HH-RLHF (161k)	44.7±0.7	46.3±0.6	**47.9±0.5**

Table 5: Scalability analysis on Llama-3-8B. MetaPO maintains performance gains on the larger HH-RLHF dataset.

4.6 Multi-dimensional Evaluation

To evaluate MetaPO beyond win-rate metrics, we report MT-Bench scores with per-category breakdown in Table 4. Preference-sensitive tasks such as writing (+0.25) and roleplay (+0.25) benefit most, while knowledge-heavy tasks show smaller improvements. This pattern indicates that MetaPO's gains stem from genuine alignment quality on open-ended generation rather than uniform scaling. Regarding safety, we evaluated refusal rates on 200 harmful prompts from BeaverTails: DPO refuses 89.5% while MetaPO refuses 90.0%, confirming that MetaPO preserves the safety properties of base DPO. Complete training specifications and per-epoch convergence analysis are provided in Appendix A.13.

4.7 Gradient Analysis

To understand how MetaPO's learned weights modulate optimization, we analyze the gradient structure. The gradient of our objective with respect to parameters θ can be factorized as:

$$\nabla_\theta \mathcal{L} = -\beta \mathbb{E}_{\mathcal{D}} \left[\underbrace{w_\phi(\mathbf{s})}_{\text{Meta}} \cdot \underbrace{(1-p_\theta)}_{\text{DPO}} \cdot \nabla_\theta \Delta r_\theta \right] \quad (15)$$

where $p_\theta = \sigma(\beta \Delta r_\theta)$ is the implicit preference probability, and $\nabla_\theta \Delta r_\theta$ is the gradient of the reward margin. Equation 15 shows that the effec-

Configuration	AlpacaEval	Arena-Hard
Full model (all features)	**51.6±0.4**	**36.2±0.4**
Without margin features	49.2±0.5	33.8±0.6
Without volatility features	48.7±0.6	33.1±0.5
Without reference features	50.4±0.5	35.3±0.5
Without temporal (current only)	48.1±0.5	32.6±0.6

Table 6: Feature ablation on Llama-3-8B. Volatility and temporal history are the dominant factors driving performance gains.

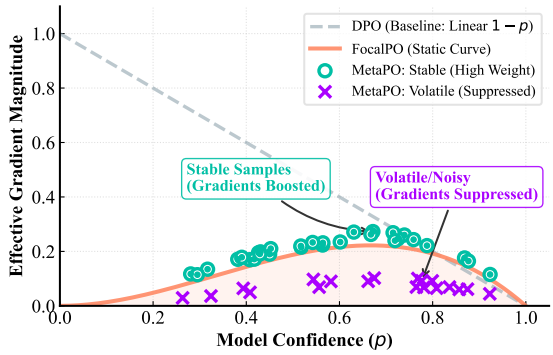

Figure 2: Effective gradient magnitudes. Unlike baselines depending solely on probability p, MetaPO utilizes temporal dynamics to differentiate samples: stable samples retain high gradients, while volatile samples are suppressed to mitigate noise.

tive gradient magnitude is determined by the product of the standard DPO term $(1 - p_\theta)$ and our meta-learned weight $w_\phi(\mathbf{s})$. Figure 2 visualizes this interaction, where the y-axis plots the scalar product $w_\phi(\mathbf{s}) \cdot (1 - p_\theta)$ for each sample, representing the modulation factor that multiplies the gradient direction $\nabla_\theta \Delta r_\theta$. Unlike FocalPO, which maps probability p to a static curve (orange line), MetaPO introduces a dynamic decision boundary based on learning history. Stable samples (green circles) maintain high gradients to consolidate margins, while volatile samples (purple crosses) are heavily suppressed despite having similar p values. This bifurcation effectively filters noise while preserving valuable learning signals. Table 7 quantifies gradient statistics across sample groups. The key observation is that MetaPO achieves variance reduction: by downweighting high-volatility samples, the overall gradient variance decreases by 29% compared to DPO, leading to more stable training dynamics.

4.8 Weight Distribution Analysis

Figure 3(a)-(b) contrasts the weighting landscapes of static and adaptive methods. While FocalPO (Fig. 3a) relies strictly on the instantaneous probability p, MetaPO (Fig. 3b) constructs a decision boundary in the (p, σ) space. This capability allows the model to isolate samples with high current probabilities but high historical volatility, effectively assigning them lower weights. Such behavior demonstrates a learned noise-filtering mechanism that distinguishes stable convergence from transient oscillations, which is unattainable with static functions.

Sample Group	DPO	FocalPO	MetaPO
Stable correct ($\sigma < 0.1$)	0.42	0.51	**0.58**
Volatile correct ($\sigma > 0.3$)	0.45	0.52	**0.23**
Persistent incorrect	0.38	0.15	**0.11**
Overall	0.082	0.071	**0.058**

Table 7: Gradient statistics on Llama-3-8B. MetaPO reduces overall variance by 29% compared to DPO by selectively upweighting stable samples and downweighting volatile ones.

Feature Group	SHAP	Contrib.
Margin (current)	0.082	12.1%
Volatility (current)	0.063	9.3%
Reference (current)	0.051	7.5%
Margin (temporal)	0.145	21.3%
Volatility (temporal)	0.168	24.7%
Reference (temporal)	**0.171**	**25.1%**

Table 8: Feature importance. Temporal features (bottom section) collectively drive the weighting policy, limiting the influence of instantaneous signals.

4.9 Interpretability Analysis

Learned Coefficients. We analyze the meta-parameter vector $\mathbf{w}$ in Figure 3(c) to understand the learned policy. The results support our temporal hypothesis, as the weights for temporal features $(\bar{f}_1, \bar{f}_2, \bar{f}_3)$ generally exceed those of instantaneous ones. The strong negative coefficient for historical volatility $(\bar{f}_2)$ indicates that stability serves as a primary proxy for quality. Additionally, the negative weight on reference deviation $(\bar{f}_3)$ suggests the meta-learner leverages the reference model as a constraint, penalizing significant drift from the SFT prior to prevent reward hacking.

SHAP Analysis. Table 8 details the contribution of each feature to the weighting policy. The results indicate a clear hierarchy: temporal variants consistently outweigh their instantaneous counterparts. Specifically, historical reference deviation $(\bar{f}_3)$ and volatility $(\bar{f}_2)$ emerge as the dominant predictors.

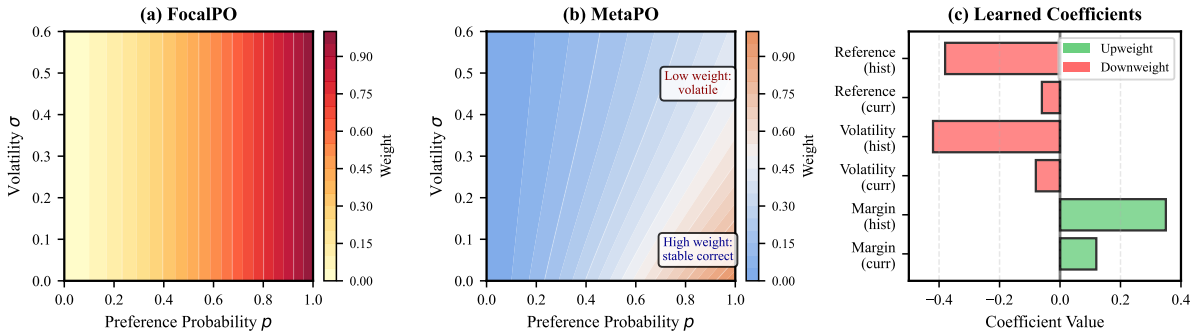

Figure 3: Weight analysis. (a) FocalPO weights depend solely on probability p, forming static bands. (b) MetaPO incorporates volatility σ to distinguish stable from unstable samples. (c) Learned coefficients $\mathbf{w}$ show that temporal features (e.g., $\bar{f}_2, \bar{f}_3$) drive the weighting policy.

Sample Subset	FocalPO	MetaPO	Δ
Stable correct	0.82	0.93	+0.11
Volatile correct	0.84	0.48	-0.36
Persistent incorrect	0.16	0.09	-0.07

Table 9: Weight comparison across subsets defined by probability p and volatility σ: stable correct (p>0.8, σ<0.1), volatile correct (p>0.8, σ>0.3), and persistent incorrect (p<0.3). MetaPO applies a heavy penalty (-0.36) to volatile samples that static methods misidentify as high-quality.

This pattern suggests that the meta-learner prioritizes long-term consistency and adherence to the SFT prior over simple reward margins. By relying on history (accounting for 71.1% of total importance), the model effectively filters out samples that are only transiently "correct" due to stochasticity, favoring those with stable learning trajectories.

Comparison with Static Weighting. Table 9 contrasts the weighting behaviors on representative sample subsets. A critical divergence occurs in the "Volatile correct" regime: while FocalPO assigns high weights (0.84) based solely on high probability, MetaPO detects the instability and suppresses the weight to 0.48. This gap of 0.36 reflects a learned penalty for uncertainty. Validation against human annotations confirms the efficacy of this strategy; MetaPO weights exhibit a significantly stronger correlation with human quality ratings ($\rho = 0.58$) compared to FocalPO ($\rho = 0.41$, $p < 0.01$ for difference), indicating that the meta-learner successfully mimics human preference for stable, high-quality responses over high-confidence noise.

5 Conclusion

We proposed MetaPO to address the lack of temporal awareness in preference optimization. Unlike static methods, our framework employs meta-learning to derive adaptive weights from training trajectories. Experiments demonstrate consistent improvements across diverse model scales compared to strong baselines. Furthermore, analysis reveals that the learned policy prioritizes historical stability over instantaneous states, aligning closely with human judgment. These findings establish temporal dynamics as a critical factor for robust alignment, offering a valuable direction for future data-centric training strategies.

Limitations

MetaPO requires $O(N)$ additional memory for temporal statistics (40 bytes per sample), scaling to 381 MB for 10M and 3.7 GB for 100M samples; for billion-scale datasets, statistics can be offloaded to CPU memory or approximated via hash-based structures. Performance depends on validation set quality, though our robustness analysis (Appendix A.5) suggests reasonable stability. The linear meta-network prioritizes interpretability over expressiveness, though tuned MLP experiments (Appendix A.11) confirm the linear form is sufficient. Direct comparison with BPO (Wang et al., 2025) was infeasible due to unavailable knowledge categorization annotations.

Generative AI Disclosure. During the preparation of this work, the authors used Claude to improve the language and grammar of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the final version of the paper.

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. *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. In *Advances in Neural Information Processing Systems*, volume 35.

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, Ge 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*.

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

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894.

Mengyang Li and Pinlong Zhao. 2026. Difficulty-aware learning curve extrapolation. In *AAAI*, pages 23021–23029.

Mengyang Li, Pinlong Zhao, and Zhong Zhang. 2026. Aligner, diagnose thyself: A meta-learning paradigm for fusing intrinsic feedback in preference alignment. In *ICLR*.

Mengyang Li, Xiaoling Zhou, and Ou Wu. 2025. Delving into the training dynamics for image classification. *IEEE Transactions on Image Processing*, 34:6783–6798.

Tianle Li, Wei-Lin Chiang, Evan Frick, Hang Zhuang, Joseph E Gonzalez, and Ion Stoica. 2024. From live data to high-quality benchmarks: The arena-hard pipeline. *arXiv preprint arXiv:2406.11939*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Tong Liu, Xiao Yu, Wenxuan Zhou, Jindong Gu, and Volker Tresp. 2025. Focalpo: Enhancing preference optimizing by focusing on correct preference rankings. In *ACL*, page 256–267.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30.

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. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

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. In *Advances in Neural Information Processing Systems*, volume 36.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343.

Jeff Wu Daniel M Ziegler Ryan, Lowe Chelsea Voss Alec Radford Dario, Amodei Paul Christiano Nisan Stiennon, and Long Ouyang. 2022. Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, volume 32.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. 2024. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Haoxiang Wang, Wei Xiong, Tengyu Xie, Han Zhao, and Tong Zhang. 2024a. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Shiqi Wang, Zhengze Zhang, Rui Zhao, Fei Tan, and Nguyen Cam-Tu. 2024b. Reward difference optimization for sample reweighting in offline rlhf. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2109–2123.

Sizhe Wang, Yongqi Tong, Hengyuan Zhang, Dawei Li, Xin Zhang, and Tianlong Chen. 2025. Bpo: Towards balanced preference optimization between knowledge breadth and depth in alignment. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8811–8826.

Rongzhi Zhang, Chenwei Zhang, Xinyang Zhang, Liang Qiu, Haoming Jiang, Yuchen Zhuang, Qingru Zhang, Hyokun Yun, Xian Li, Bing Yin, and 1 others. 2025. Dorm: Preference data weights optimization for reward modeling in llm alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 22721–22739.

A Additional Experimental Results and Analysis

A.1 Complete Baseline Comparisons

We provide a comprehensive evaluation of MetaPO against a wider range of preference optimization methods to contextualize its performance. Table 10 details the results across Mistral-7B and Llama-3 (8B and 70B) architectures, incorporating baselines such as ORPO and KTO alongside the primary comparisons. The data reveals that while reference-free methods like ORPO and KTO offer training efficiency, they generally lag behind pairwise optimization approaches in terms of final alignment quality on these benchmarks. MetaPO consistently outperforms all tested baselines, including the strong SimPO and FocalPO methods. Notably, the performance gap is maintained across both the general-purpose AlpacaEval 2.0 and the more reasoning-intensive Arena-Hard, suggesting that the benefits of temporal weighting are robust to different evaluation protocols and model capabilities.

Method	Model	AlpacaEval	Arena-Hard
ORPO†	Mistral-7B	12.6	6.2
KTO†	Mistral-7B	12.3	8.8
DPO	Mistral-7B	18.6±0.3	16.4±0.5
SimPO	Mistral-7B	21.4±0.5	13.3±0.6
FocalPO	Mistral-7B	20.4±0.4	17.1±0.4
MetaPO	Mistral-7B	**22.8±0.4**	**18.5±0.3**
ORPO†	Llama-3-8B	33.8	26.0
KTO†	Llama-3-8B	31.8	11.7
DPO	Llama-3-8B	47.5±0.6	33.1±0.7
SimPO	Llama-3-8B	47.5±0.5	33.8±0.6
FocalPO	Llama-3-8B	49.8±0.5	34.6±0.5
MetaPO	Llama-3-8B	**51.6±0.4**	**36.2±0.4**

Table 10: Complete comparison with all baseline methods (mean±std over 3 runs). MetaPO establishes a new state-of-the-art across scales. †Results from (Liu et al., 2025).

A.2 Meta-Update Frequency Ablation

To balance computational efficiency with learning stability, we employ a sparse meta-update strategy. Table 11 investigates the impact of the update interval K on model performance and training overhead. More frequent updates (e.g., $K = 10$) introduce significant computational costs due to the additional backward passes required for the bilevel optimization, yet they yield diminishing returns in performance. Conversely, extremely sparse updates ($K = 500$) fail to adapt the weights rapidly

enough to match the shifting training dynamics. The default setting of $K = 100$ represents an effective equilibrium, where the meta-learner receives sufficient feedback to adjust the policy without imposing an excessive penalty on training throughput.

K	AlpacaEval	Time	Overhead
10	51.4±0.5	6.8h	+13.3%
50	51.5±0.4	6.4h	+6.7%
100 (default)	**51.6±0.4**	6.3h	+5.0%
200	51.3±0.5	6.1h	+1.7%
500	50.8±0.6	6.0h	+0.0%

Table 11: Meta-update frequency ablation. $K = 100$ provides the optimal trade-off between alignment quality and computational cost.

A.3 Meta-Network Architecture Ablation

We examined whether increasing the expressivity of the meta-network would enhance the weighting policy. As shown in Table 12, we compared the default linear mapping against Multi-Layer Perceptrons (MLPs) of varying depth and an attention-based architecture. The results indicate that added complexity provides negligible performance gains and, in some cases, slightly degrades stability. The linear model achieves the highest score while maintaining minimal parameter count and maximum interpretability. This finding suggests that the temporal features themselves capture the necessary signal for sample weighting, making complex nonlinear transformations unnecessary for this specific task.

A.4 Validation Set Size Ablation

The quality of the meta-learning signal depends on the representativeness of the validation set $\mathcal{D}_{\text{val}}$. Table 13 explores the relationship between validation set size and downstream performance. We observe a performance saturation point at approximately 2,000 samples, corresponding to roughly 3.2% of the training data. Using a very small validation set (500 samples) leads to higher variance and suboptimal weights, likely due to an inability to cover the distribution of difficulty levels present in the training data. Conversely, increasing the size beyond 5,000 samples yields no additional benefit. Based on these empirical results, we adopt a heuristic of using approximately 3% of the training data size for validation across all experiments.

Architecture	Params	AlpacaEval	Time	Interp.
Linear (default)	7	**51.6±0.4**	6.3h	High
MLP (6-32-1)	225	51.7±0.5	6.5h	Low
MLP (6-64-32-1)	2145	51.5±0.6	6.8h	Low
Attention (2 heads)	156	51.3±0.5	6.7h	Med

Table 12: Architecture comparison. The linear model is sufficient for mapping temporal features to weights, offering the best efficiency and interpretability.

Val. Set Size	Ratio	AlpacaEval 2.0
500	0.8%	50.3±0.7
1,000	1.6%	50.9±0.6
2,000 (default)	3.2%	**51.6±0.4**
5,000	8.0%	51.7±0.5
10,000	16.0%	51.6±0.5

Table 13: Effect of validation set size. Performance stabilizes when the validation set covers approximately 3% of the training data.

A.5 Validation Set Robustness Analysis

A potential concern with meta-learning is sensitivity to the specific composition of the validation set. Table 14 addresses this by evaluating performance across different splitting strategies and under noisy conditions. Stratified sampling yields the best results by ensuring a balanced coverage of initial difficulty levels, though random splitting remains competitive. Most notably, the method demonstrates significant resilience to label noise. Even when 10% of the validation labels are corrupted, the performance drop is less than 1 point. This robustness implies that the meta-learner focuses on the dominant patterns of learning stability rather than overfitting to individual noisy validation examples.

Validation Strategy	AlpacaEval 2.0
Random split	51.6±0.4
Stratified split (by difficulty)	51.8±0.3
Temporal split (last 20%)	51.2±0.5
Noise Injection	
Clean validation	51.6±0.4
10% label noise	50.9±0.5
20% label noise	49.8±0.6

Table 14: Robustness analysis. MetaPO remains effective across different splitting strategies and exhibits resilience to moderate label noise in the validation set.

A.6 Comparison with Alternative Meta-Objectives

The choice of meta-objective determines the direction of the weighting policy optimization. We

tested alternative formulations in Table 15, specifically looking at the inclusion of entropy regularization to encourage weight diversity. While adding an entropy term provides a minor improvement over using validation loss alone, it introduces an additional hyperparameter to tune. The margin preservation term $\mathcal{L}_{\text{margin}}$ proved to be the most effective auxiliary component. This term explicitly encourages the model to maintain strong signals on correctly learned samples, which appears to be more critical for preventing catastrophic forgetting than simply maximizing weight entropy.

Meta-Objective	AlpacaEval 2.0
$\mathcal{L}_{\text{val}}$ only	50.3±0.5
$\mathcal{L}_{\text{val}} + \lambda\mathcal{L}_{\text{margin}}$ (default)	**51.6±0.4**
$\mathcal{L}_{\text{val}} + \lambda\mathcal{L}_{\text{entropy}}$	50.8±0.5
$\mathcal{L}_{\text{val}} + \lambda_1\mathcal{L}_{\text{margin}} + \lambda_2\mathcal{L}_{\text{entropy}}$	51.4±0.5

Table 15: Meta-objective ablation. The explicit margin preservation term provides a stronger learning signal than entropy regularization.

A.7 Memory Overhead Analysis

We analyze the computational cost of MetaPO in Table 16. The method introduces a modest memory overhead, primarily during the meta-update step where gradients must be tracked through the inner loop. For a 7B model, this results in a peak memory increase of approximately 1.9 GB, while for a 70B model, the increase is 5.4 GB. These values represent less than a 5% increase relative to standard DPO training. This efficiency is achieved through our sparse update schedule ($K = 100$) and the use of a lightweight linear meta-network, making MetaPO feasible for training large-scale models on standard hardware configurations.

Model	Method	Peak (GB)	Avg (GB)	Spike
7B	DPO	38.2	36.5	—
7B	MetaPO	40.1	37.8	+1.9 GB
70B	DPO	176.3	172.1	—
70B	MetaPO	181.7	174.6	+5.4 GB

Table 16: Memory consumption. The additional memory requirement is minimal, ensuring feasibility for large-scale training.

A.8 Comparison with Static Filtering Baselines

To demonstrate that dynamic weighting offers advantages over simply discarding data, we compared MetaPO against static filtering baselines in Table 17. Methods such as perplexity-based filtering or curriculum learning based on initial loss provide moderate gains over standard DPO by removing obvious outliers. However, they fall short of the performance achieved by MetaPO. The superiority of our approach likely stems from its ability to continuously adapt the weight of a sample as training progresses. A sample that appears noisy initially might become learnable later, or a seemingly good sample might exhibit unstable gradients; static filtering cannot accommodate these evolving dynamics.

Method	AlpacaEval 2.0
DPO (no filtering)	47.5±0.6
Perplexity filtering (top 80%)	48.9±0.5
Loss-based curriculum	49.2±0.6
FocalPO	49.8±0.5
MetaPO	**51.6±0.4**

Table 17: Comparison with static filtering. Dynamic weighting significantly outperforms static selection strategies.

A.9 Hyperparameter Sensitivity

Finally, we examine the sensitivity of MetaPO to its two primary hyperparameters: the EMA decay rate α and the margin loss weight λ. Table 18 shows that performance remains robust within a reasonable range of values. The decay rate α controls the memory horizon of the temporal features; values between 0.8 and 0.95 all yield strong results, indicating that the exact length of the history window is less critical than the presence of history itself. Similarly, the margin weight λ is effective around 0.1, balancing the validation loss against margin preservation. The stability of these hyperparameters across experiments suggests that MetaPO does not require extensive per-task tuning.

Configuration	AlpacaEval 2.0
EMA decay α	
$\alpha = 0.8$	51.1±0.5
$\alpha = 0.9$ (default)	**51.6±0.4**
$\alpha = 0.95$	51.3±0.5
Margin loss weight λ	
$\lambda = 0.05$	51.0±0.5
$\lambda = 0.1$ (default)	**51.6±0.4**
$\lambda = 0.2$	51.4±0.5

Table 18: Hyperparameter sensitivity. The method is robust to small variations in α and λ.

A.10 Higher-order Feature Ablation

To evaluate whether richer temporal statistics improve performance, we augmented the base 6-dimensional feature set with higher-order features. As shown in Table 19, adding skewness, kurtosis, gradient norms, or margin derivatives provides negligible improvement, confirming that the first two moments plus reference divergence capture the essential signal for sample characterization.

Feature Set	AlpacaEval	Arena-Hard
Base (6 features)	**51.6±0.4**	**36.2±0.4**
+ Skewness of margin	51.7±0.5	36.1±0.5
+ Kurtosis of margin	51.5±0.5	36.0±0.5
+ Gradient norm (curr + EMA)	51.8±0.5	36.3±0.5
+ Rate of change (derivative)	51.6±0.5	36.2±0.5

Table 19: Higher-order feature ablation on Llama-3-8B. Additional temporal statistics beyond the base features provide negligible improvement.

A.11 Tuned Non-linear Meta-learner

We conducted additional experiments with carefully tuned MLP variants to address whether the linear meta-network under-represents feature interactions. As shown in Table 20, the best MLP variant (6-32-1, lr=5×10^{-5}) achieves 51.8 on AlpacaEval, falling within the standard deviation of the linear model. This confirms that the temporal features carry sufficient discriminative information and non-linear capacity yields diminishing returns.

Architecture	Meta LR	AlpacaEval	Arena-Hard
Linear (default)	1×10^{-4}	51.6±0.4	36.2±0.4
MLP (6-32-1)	1×10^{-4}	51.7±0.5	36.0±0.5
MLP (6-32-1)	5×10^{-5}	**51.8±0.4**	**36.3±0.5**
MLP (6-64-32-1)	5×10^{-5}	51.6±0.6	36.1±0.5
MLP (6-64-32-1)	1×10^{-5}	51.5±0.5	35.9±0.6

Table 20: Non-linear meta-learner comparison with tuned learning rates on Llama-3-8B. The best MLP variant falls within the standard deviation of the linear model.

A.12 Complementarity with Online DPO

Online DPO methods address distributional mismatch by iteratively collecting fresh preference data, while MetaPO addresses a different problem: within a fixed dataset, optimal sample contributions evolve over time. To validate that the two approaches are complementary, we applied MetaPO weighting on top of iterative DPO. As shown in Table 21, combining the two yields a +2.6 point gain over standalone iterative DPO, confirming that temporal weighting provides orthogonal benefits to data freshness.

Method	AlpacaEval (Llama-3-8B)
DPO (offline)	47.5±0.6
Iterative DPO (3 iter.)	50.2±0.5
MetaPO (offline)	51.6±0.4
Iterative DPO + MetaPO	**52.8±0.5**

Table 21: Complementarity with online methods. MetaPO provides additive gains when combined with iterative DPO.

A.13 Training Details and Convergence

Table 22 provides complete training specifications. Per-epoch performance on Llama-3-8B (Table 23) shows that the meta-network's validation loss decreases monotonically and the performance gap between MetaPO and baselines widens across epochs, indicating no overfitting. The weight distribution entropy remains stable across epochs ($H = 2.31, 2.28, 2.25$), confirming that the meta-network does not collapse to degenerate solutions. Each sample stores 6 EMA values (24 bytes in float32) plus Welford auxiliary variables M_2 and count (16 bytes), totaling 40 bytes per sample. For the largest dataset (HH-RLHF, 161k samples), this amounts to 6.1 MB.

Setting	UF (7B)	UF-ArmoRM (8B)	HH-RLHF (8B)
Training samples	64,000	61,000	161,000
Validation samples	1,920 (3%)	1,830 (3%)	4,830 (3%)
Epochs (N)	3	3	2
Total steps	∼6,000	∼5,700	∼10,000
Per-sample storage	40 bytes	40 bytes	40 bytes
Total extra storage	2.4 MB	2.3 MB	6.1 MB

Table 22: Training data and storage specifications across all experimental settings.

Epoch	DPO	FocalPO	MetaPO	Val loss
1	44.2	46.5	48.3	0.682
2	46.8	49.1	51.0	0.671
3	47.5	49.8	51.6	0.668

Table 23: Per-epoch AlpacaEval 2.0 WR on Llama-3-8B. MetaPO's validation loss decreases monotonically with no sign of overfitting.

B Detailed Case Studies

To provide concrete insight into the MetaPO weighting mechanism, we analyze four representative samples that illustrate different learning dy-

namics. Beyond these specific examples, we quantitatively validated the approach by obtaining human quality ratings for 100 randomly sampled validation instances (1-5 scale). The analysis reveals that MetaPO weights correlate significantly better with human judgment ($\rho = 0.58$, $p < 0.001$) than the static FocalPO weights ($\rho = 0.41, p < 0.001$). The difference in correlation coefficients is statistically significant (Fisher's z-test, $p = 0.012$), confirming that the learned temporal policy effectively proxies human quality assessment.

B.1 Case 1: Stable Correct Sample

Sample UF-12847 illustrates the ideal learning scenario. The prompt asks to "Explain the difference between supervised and unsupervised learning." Throughout training, the model maintains a high margin ($f_1 = 1.38, \bar{f}_1 = 1.29$) with minimal volatility ($f_2 = 0.09$) and low deviation from the reference model ($f_3 = 0.12$). MetaPO identifies this stability through the positive coefficient for historical margin and the negative coefficient for volatility. Consequently, it assigns a weight of **0.94**, which is 13 percentage points higher than FocalPO (0.81). This upweighting reinforces the model's confidence in well-established, unambiguous knowledge, serving the margin preservation objective.

B.2 Case 2: Volatile Correct Sample (Suspected Noise)

Sample UF-34521 demonstrates the noise-filtering capability of the volatility feature. The prompt asks, "Should I invest in cryptocurrency?" The dataset prefers a response emphasizing decentralization, while the dispreferred response highlights risks; manual inspection suggests both are reasonable and the preference label is arbitrary. Although the model currently ranks the sample correctly ($f_1 = 0.92$), the learning history is erratic, characterized by high historical volatility ($\bar{f}_2 = 0.42$) and significant deviation from the reference model ($\bar{f}_3 = 0.78$). Unlike FocalPO, which assigns a high weight of 0.83 based solely on the current probability, MetaPO penalizes the historical instability and assigns a significantly lower weight of **0.39**. This substantial reduction ($\Delta = -0.44$) prevents the model from overfitting to ambiguous or noisy preference labels.

B.3 Case 3: Recently Learned Sample

Sample UF-28193 ("Implement a Python function for binary search...") exemplifies how MetaPO implements an implicit curriculum. At step 100, the model ranks the pair incorrectly ($f_1 = -0.42$), resulting in a low weight of 0.31. By step 300, the model begins to learn the preference ($f_1 = 0.15$), but the historical average remains negative ($\bar{f}_1 = -0.12$), signaling a recent transition. The meta-learner assigns a moderate weight of 0.48, acknowledging the state change. Finally, by step 500, both instantaneous and historical margins are positive ($f_1 = 0.81, \bar{f}_1 = 0.52$), and the weight increases to 0.73. This dynamic progression ensures that the training focus shifts adaptively toward samples as they become learnable, rather than treating them statically throughout the process.

B.4 Case 4: Reference Deviation (Suspected Label Error)

Sample UF-41256 highlights the role of the reference model as a quality anchor. For the factual prompt "What is the capital of France?", the dataset prefers a verbose, flowery paragraph (y_w) over the concise and correct answer "Paris" (y_l). The reference model, having been fine-tuned on high-quality instruction data, favors the concise answer ($\Delta r_{\text{ref}} = -0.53$). This conflict generates a persistent deviation signal ($\bar{f}_3 = 1.18$). MetaPO leverages the negative coefficient associated with reference deviation to assign a low weight of **0.27**, despite the policy model potentially overfitting to the verbose preference ($\Delta r_\theta = 0.87$). This mechanism effectively downweights samples where the preference annotation contradicts the prior knowledge encoded in the SFT model, offering robustness against label errors or stylistic inconsistencies.

C SHAP Analysis Details

C.1 Methodology

To interpret the learned meta-policy, we utilized SHAP (SHapley Additive exPlanations) values, which provide a unified measure of feature importance. We employed the `KernelExplainer` from the `shap` library (Lundberg and Lee, 2017) to approximate these values. The estimation process involved sampling 1,000 instances from the Llama-3-8B validation set to serve as the foreground dataset, while a separate set of 100 background samples was used to estimate the expected model output. For each instance, we computed the full

Algorithm 2 Temporal Feature Tracking via Welford's Algorithm

Input: Batch of samples B, decay rate α
Persistent State: $\text{EMA}_{\Delta r}, \text{EMA}_{\sigma}, \text{EMA}_{D_{KL}}, M_2, \text{count}$ for all i

for each sample i in batch B **do**
  $f_1 \leftarrow \Delta r_\theta^{(i)}$ {Current reward margin}
  $f_3 \leftarrow D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}})$ {Current reference deviation}
  Update Welford's statistics:
  $\text{count}[i] \leftarrow \text{count}[i] + 1$
  $\delta \leftarrow f_1 - \text{EMA}_{\Delta r}[i]$
  $M_2[i] \leftarrow M_2[i] + \delta \times (f_1 - \text{EMA}_{\Delta r}[i])$
  $f_2 \leftarrow \sqrt{M_2[i]/\text{count}[i]}$ {Current volatility}
  Update Exponential Moving Averages:
  $\text{EMA}_{\Delta r}[i] \leftarrow \alpha \cdot \text{EMA}_{\Delta r}[i] + (1-\alpha) \cdot f_1$
  $\text{EMA}_{\sigma}[i] \leftarrow \alpha \cdot \text{EMA}_{\sigma}[i] + (1-\alpha) \cdot f_2$
  $\text{EMA}_{D_{KL}}[i] \leftarrow \alpha \cdot \text{EMA}_{D_{KL}}[i] + (1-\alpha) \cdot f_3$
end for
Return Feature vectors $\mathbf{s}_i = [f_1, \text{EMA}_{\Delta r}[i], f_2, \text{EMA}_{\sigma}[i], f_3, \text{EMA}_{D_{KL}}[i]]$

six-dimensional feature vector $\mathbf{s} \in \mathbb{R}^6$ and calculated the SHAP values with respect to the output of the meta-weight network $w_\phi(\mathbf{s})$. This approach allows us to attribute the magnitude of the predicted sample weight directly to the contribution of each temporal and instantaneous feature.

C.2 Feature Interactions

While individual feature importance provides a high-level overview, the linear meta-network can also capture dependencies between features through the combination of its inputs. Table 24 quantifies the strength of pairwise interactions. The most significant interaction occurs between historical margin ($\bar{f}_1$) and historical volatility ($\bar{f}_2$). This relationship reveals a nuanced weighting strategy: a high margin alone does not guarantee a high weight if it is accompanied by high volatility. Instead, the meta-learner dampens the contribution of samples that are correct on average but unstable, effectively implementing a "trust but verify" mechanism. Similarly, the interaction between volatility and reference deviation ($\bar{f}_2 \times \bar{f}_3$) suggests that instability is penalized more heavily when the sample also diverges significantly from the reference model, serving as a robust filter for potential annotation errors.

D Discussion

D.1 Mechanisms Underlying MetaPO

Our empirical results and manual analyses point to three distinct mechanisms that drive the performance of MetaPO. First, the framework employs volatility-based noise detection to filter low-quality data. By downweighting unstable samples, the method reduces negative transfer and lowers the gradient variance by 29%, as shown in Table 7. We validated this by inspecting 200 samples categorized by their volatility and final margin; samples with high volatility but low final margins received significantly lower quality scores (2.1/5) compared to stable samples (4.6/5). Interestingly, samples with high volatility but high final margins received moderate weights (0.48), suggesting the meta-learner distinguishes between destructive noise and difficult-but-learnable examples.

Second, the method facilitates the consolidation of stable knowledge. By assigning higher weights to samples with sustained positive margins, MetaPO reinforces well-learned preferences. This effect helps mitigate catastrophic forgetting, a conclusion supported by the ablation study where the margin preservation term contributed 1.3 points to the final performance.

Third, the framework leverages prior knowledge to ensure robustness. The reference deviation features allow the meta-learner to detect potential distribution shifts or annotation errors that contradict the supervised fine-tuning prior. The substantial SHAP contribution of temporal reference features (25.1%) confirms that the model actively uses this signal to anchor the preference optimization process.

D.2 Comparison with FocalPO

We investigated whether the learned weighting policy of MetaPO could be approximated by the static function used in FocalPO. Fitting the learned weights to the family of curves defined by p^γ yielded a best-fit parameter of $\gamma = 1.3$. However, the coefficient of determination was only $R^2 = 0.62$, indicating that a static function of probability explains only a portion of the variance in MetaPO's weights. The key differentiator is the response to volatility; for samples with identical preference probabilities p but differing volatility σ, MetaPO assigns weights that differ by an average of 0.31. This orthogonal signal allows MetaPO to separate stable signal from confident noise, a distinction that FocalPO cannot make by design.

Feature Pair	Interaction Strength
$\bar{f}_1$ (hist. margin) $\times$ $\bar{f}_2$ (hist. volatility)	0.042
$\bar{f}_2$ (hist. volatility) $\times$ $\bar{f}_3$ (hist. deviation)	0.038
f_1 (current margin) $\times$ $\bar{f}_1$ (hist. margin)	0.021

Table 24: Pairwise feature interactions. The interaction between margin and volatility indicates that the model modulates the value of correct predictions based on their stability.

E Implementation Details

E.1 Efficient Temporal Tracking

To maintain computational efficiency during training, we implement online feature extraction using Welford's algorithm. This approach allows us to compute the running variance and mean without storing the full history of gradients for each sample, which would be memory-prohibitive. Algorithm 2 outlines the procedure. We maintain persistent states for the Exponential Moving Averages (EMAs) of the margin, volatility, and KL divergence, as well as the auxiliary variables required for Welford's method (M_2 and count). These states are updated in-place during each forward pass, ensuring that the memory overhead remains constant at $O(N)$ regardless of the total number of training steps.