MSPO: Meta Soft Preference Optimization for Robust LLM Alignment

Anonymous ACL submission

Abstract

Noise in preference data significantly impedes the alignment of large language models (LLMs) with human preferences. However, existing methods struggle with two key challenges: reliably identifying noisy preferences and accurately representing preference intensity. To address these challenges, we introduce Meta Soft Preference Optimization (MSPO), a novel framework. MSPO employs a meta-learner to optimize soft preference labels for the alignment task. This meta-learner produces new, adaptive soft labels that more accurately reflect true preference strength and mitigate noise. To achieve this, it primarily processes perplexity differences (PPLDiff) between paired responses, where PPLDiff itself is calculated based on the initial preference indications. The meta-learner is optimized using a small, clean meta-dataset to enhance downstream LLM alignment performance. Extensive experiments demonstrate that MSPO effectively mitigates the adverse effects of noisy preferences. It significantly improves the robustness of LLM alignment in various noisy environments and outperforms existing baseline methods.

1 Introduction

Large Language Models (LLMs) show remarkable capabilities across various natural language tasks (Brown et al., 2020; Anil et al., 2023; Touvron et al., 2023). Aligning these models with human values to ensure helpful, honest, and harmless outputs is crucial for their deployment (Lee et al., 2023; Askell et al., 2021; Amodei et al., 2016). Learning from human preferences, particularly through methods like Direct Preference Optimization (DPO) (Rafailov et al., 2023), has become a prominent approach for this alignment, evolving from earlier Reinforcement Learning from Human Feedback (RLHF) paradigms (Christiano et al., 2017; Ouyang et al., 2022).

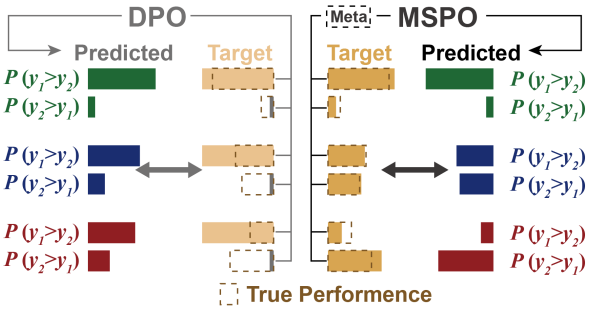

Figure 1: Conceptual illustration of DPO versus MSPO. DPO utilizes fixed preference targets, while MSPO introduces a meta-learner to generate adaptive soft targets, enabling more robust LLM alignment.

However, the quality of preference data poses a significant challenge. Collected datasets frequently contain noise from sources such as annotator subjectivity and disagreement (Sheng et al., 2008), task misinterpretation, random errors, or imperfections in AI-generated feedback (Bai et al., 2022b; Liang et al., 2024; Ziegler et al., 2019). Such noisy preferences can severely undermine the alignment process, leading to suboptimal model performance and unreliable behavior, failing to capture true human intent.

Mitigating the impact of noisy preferences presents two main difficulties. First, accurately identifying specific noisy preference pairs is challenging. Distinguishing subtle preferences from erroneous ones requires nuanced understanding. Second, quantifying the unreliability of a preference and adjusting its influence during training is difficult, especially when using soft preference labels (Furuta et al., 2024). Soft labels aim to represent preference strength but can be distorted by noise, potentially amplifying misleading signals if not properly calibrated.

Previous approaches to address noisy preferences include robust loss functions, such as Conservative DPO (cDPO) (Mitchell, 2023) and Robust DPO (rDPO) (Chowdhury et al., 2024). These

methods often adjust the DPO loss based on an estimated global noise ratio, typically derived from a small clean validation set. While beneficial, they may not fully adapt to instance-specific noise. Other data-centric methods, like Perplexity-aware Correction (PerpCorrect) (Kong et al., 2024), use signals such as perplexity differences (PPLDiff) to correct noisy pairs as a preprocessing step. Although PPLDiff is a useful signal for noise detection, relying on fixed thresholds or heuristic correction rules may limit the optimality of the noise mitigation. An adaptive mechanism, such as MSPO's meta-learner, to intelligently leverage such signals for refining soft preference labels is needed.

To address these limitations, we introduce **Meta Soft Preference Optimization (MSPO)**, a novel meta-learning framework for robust LLM alignment. MSPO learns to generate adaptive soft preference labels by interpreting dynamic noise-indicative signals derived directly from the main language model being aligned.

Figure 1 illustrates the key difference between traditional Direct Preference Optimization (DPO) and our MSPO. DPO (left) uses fixed target preferences to guide model optimization. In contrast, MSPO (right) employs a meta-learner ("Meta") to dynamically generate optimized, adaptive soft targets. These adaptive targets can modulate preference intensity and correct mislabeled preferences (suggested by varying bar lengths and potential inversions in the figure), guiding LLM alignment more robustly, especially with noisy data.

The core of MSPO is its meta-learner, denoted as $V(\cdot;\phi)$. This meta-learner is trained to process a key, evolving indicator of preference consistency or potential mislabeling: the perplexity difference (PPLDiff) between paired responses. This PPLDiff is calculated based on the original preference ordering (i.e., which response was initially labeled as preferred) and critically, using the **current main LLM** $\pi_{\theta^{(t)}}$ at each step. The meta-learner V then takes this PPLDiff value as input to output an optimized soft preference label $\hat{p}_\phi \in [0,1]$, which subsequently modulates a DPO-style objective for training the main LLM π_θ. The meta-learner's parameters ϕ are optimized via a bilevel process, guided by the main LLM's performance on a small, clean meta-dataset. This data-driven strategy enables V to produce high-quality, adaptive soft labels that effectively mitigate noise and adapt to the main model's evolving understanding. The main contributions of this work are threefold:

- We pioneer the application of meta-learning to address noisy preference data in LLM alignment, providing theoretical insights into its robustness and convergence properties.
- We propose MSPO, a novel framework instantiating this meta-learning paradigm. MSPO utilizes a dedicated meta-learner to adaptively generate soft preference labels from dynamically-derived PPLDiff, achieving fine-grained and instance-specific noise mitigation.
- Through extensive experiments on benchmark datasets with various noise settings, we demonstrate that MSPO significantly improves the robustness of LLM alignment and outperforms existing state-of-the-art techniques.

2 Related Work

LLM Alignment with Human Preferences. Aligning Large Language Models (LLMs) with human values is critical for their responsible deployment (Ouyang et al., 2022; Stiennon et al., 2020). Direct Preference Optimization (DPO) (Rafailov et al., 2023) has become a prominent approach, directly learning from preference pairs. This builds upon a rich history of learning from human feedback, including explicit reward modeling followed by reinforcement learning (Knox and Stone, 2008; Warnell et al., 2018). To capture the nuanced nature of human preferences beyond binary choices, soft preference labels, representing preference strength or probability, have been explored (Furuta et al., 2024). For instance, Geometric-Averaged DPO (GDPO) (Furuta et al., 2024) integrates soft labels to modulate the DPO learning process. However, the effectiveness of using soft labels heavily relies on their quality, which is often compromised by noise in the collected preference data. Our work, MSPO, focuses on improving the quality of these soft labels in noisy settings.

Addressing Noisy Preferences in LLM Alignment. The presence of noisy preferences significantly impairs LLM alignment robustness (Gao et al., 2024; Casper et al., 2023). Existing strategies to mitigate noisy preferences broadly fall into two categories. First, robust loss-based methods like Conservative DPO (cDPO) (Mitchell, 2023) and Robust DPO (rDPO) (Chowdhury et al., 2024) adjust the DPO objective based on an estimated

global noise ratio, often requiring a clean validation set. These are part of a broader class of techniques for learning with noisy labels that modify loss functions or sample importances (Patrini et al., 2017; Song et al., 2022). While helpful, these methods may lack adaptability to instance-specific noise characteristics. Second, data-centric approaches such as Perplexity-aware Correction (PerpCorrect) (Kong et al., 2024) aim to detect and correct noisy preferences using signals like perplexity differences (PPLDiff) as a data pre-processing step. These methods, however, often rely on heuristic rules or fixed thresholds for correction. MSPO differs by adaptively learning how to generate or refine soft preference labels using these signals within a meta-learning framework, rather than applying fixed pre-processing rules.

Meta-learning for Robust Label Correction. Meta-learning has proven effective for robust learning in noisy settings, particularly for label correction and sample re-weighting in classification tasks (Ren et al., 2018; Shu et al., 2019; Wu et al., 2021; Wang et al., 2020). These methods typically train a meta-learner on a small set of clean data to learn a strategy (e.g., a weighting function or a label correction model) that improves the main task's performance on noisy data. For example, Meta-Weight-Net (Shu et al., 2019) learns a function to assign weights to training samples based on their loss. While these works demonstrate the potential of meta-learning for handling label noise, its application to adaptively optimize soft preference labels within LLM preference alignment, especially by leveraging model-intrinsic signals like PPLDiff dynamically derived from the LLM being aligned, has been less explored. MSPO aims to bridge this gap by proposing such a meta-learning framework tailored for soft preference label optimization in LLM alignment.

3 Methodology

This section details our proposed meta-learning framework, MSPO, for adaptively generating soft preference labels to improve the robustness of LLM alignment. We begin by outlining the foundational concepts of DPO and GDPO. We then introduce how Perplexity Difference (PPLDiff), calculated using the main language model being trained, can serve as a dynamic noise indicator. Finally, we present the architecture of MSPO and elaborate on its meta-optimization procedure, designed to learn an effective soft label generation strategy.

3.1 Preliminaries

3.1.1 Direct Preference Optimization (DPO)

Direct Preference Optimization (DPO) (Rafailov et al., 2023) directly aligns LLMs from preference data $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}$, avoiding explicit reward modeling. It optimizes the policy π_θ by minimizing the negative log-likelihood of preferences under a Bradley-Terry model. This encourages higher probabilities for preferred responses y_w over dispreferred ones y_l, relative to a reference policy π_{ref}:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\left(\log\frac{\pi_\theta(y_w \mid x)}{\pi_{ref}(y_w \mid x)} - \log\frac{\pi_\theta(y_l \mid x)}{\pi_{ref}(y_l \mid x)}\right)\right)\right]. \tag{1}$$

3.1.2 Geometric-averaged DPO (GDPO) with Soft Preference Labels

Recognizing that human preferences often possess varying degrees of certainty, Geometric-averaged DPO (GDPO) (Furuta et al., 2024) extended DPO by incorporating soft preference labels $\hat{p}$. This label $\hat{p} \in [0.5, 1.0]$ quantifies the estimated probability $P(y_1 \succ y_2|x)$, where y_1 is nominally preferred. GDPO modulates the core DPO log-ratio, $h_\theta(x, y_1, y_2) = \log\frac{\pi_\theta(y_1|x)\pi_{ref}(y_2|x)}{\pi_{ref}(y_1|x)\pi_\theta(y_2|x)}$, by a factor $(2\hat{p} - 1)$:

$$\mathcal{L}_{\text{GDPO}}(\pi_\theta; \pi_{ref}, \hat{p}) = -\mathbb{E}_{(x,y_1,y_2,\hat{p})\sim\mathcal{D}}\left[\log\sigma\left(\beta(2\hat{p} - 1)h_\theta(x, y_1, y_2)\right)\right]. \tag{2}$$

3.1.3 Dynamic Perplexity Difference (PPLDiff) as a Noise Indicator

The perplexity (PPL) of a sequence reflects its likelihood under a given language model. The difference in log-PPL between two responses y_1 and y_2 to a prompt x,

$$\text{PPLDiff}(x, y_1, y_2; \pi_\theta) = \log\text{PPL}([x; y_1]; \pi_\theta) - \log\text{PPL}([x; y_2]; \pi_\theta), \tag{3}$$

can serve as a potent indicator of preference noise (Kong et al., 2024). Specifically, a negative PPLDiff suggests π_θ assigns a higher likelihood to y_1 over y_2 (implying y_1 is preferred by the model), and vice versa for a positive PPLDiff. This quantifiable preference signal derived from PPLDiff is thus utilized as a key input signal by meta-learner in MSPO.

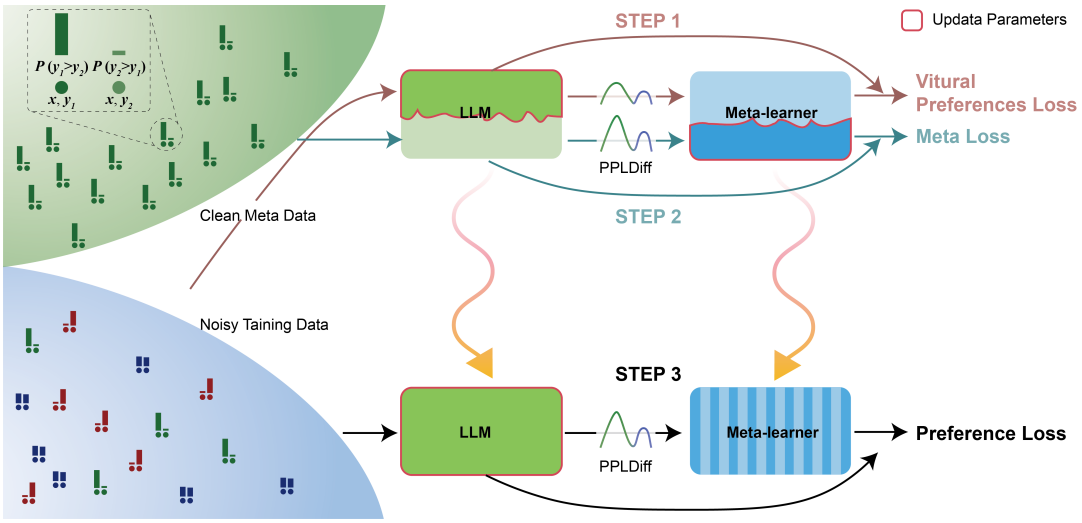

Figure 2: Overview of the MSPO framework. A meta-learner $V(\cdot;\phi)$ processes PPLDiff, calculated using the current main LLM $\pi_{\theta^{(t)}}$, to generate optimized soft labels $\hat{p}_\phi$. These labels guide the training of π_θ via a GDPO-style preference loss. V is optimized using a clean meta-dataset D_{meta} to enhance π_θ's alignment performance.

3.2 MSPO: Meta-learning for Soft Preference Label Optimization

Obtaining reliable soft preference labels $\hat{p}$ is challenging in noisy datasets. MSPO addresses this by introducing a meta-learning framework. This framework trains a meta-learner $V(\cdot;\phi)$ to generate appropriate soft preference labels $\hat{p}_\phi$ directly from a dynamic Perplexity Difference (PPLDiff) signal. The overall architecture is depicted in Figure 2.

3.2.1 The Meta-learner Module $V(\cdot;\phi)$

The meta-learner V, parameterized by ϕ, is a neural network. Its role is to map an input PPLDiff feature to a soft preference label $\hat{p}_\phi \in [0,1]$. For each training sample (x, y_1, y_2) from the noisy training set $\mathcal{D}_{train}$, the PPLDiff is calculated using the current main policy $\pi_{\theta^{(t)}}$:

$$\text{PPLDiff}^{(t)} = \text{PPLDiff}(x, y_1, y_2; \pi_{\theta^{(t)}}). \quad (4)$$

This $\text{PPLDiff}^{(t)}$ serves as the input to $V(\cdot;\phi)$, which then outputs the soft label:

$$\hat{p}_\phi = V(\text{PPLDiff}^{(t)}; \phi). \quad (5)$$

This generated label $\hat{p}_\phi$ is subsequently used in a GDPO-style loss to train the main LLM π_θ.

3.2.2 Bilevel Optimization Procedure

The training of MSPO involves optimizing the main LLM parameters θ and the meta-learner parameters ϕ through a bilevel process. At each iteration t, this process unfolds in three key steps:

STEP 1: Compute Virtual LLM Parameter Update. This step simulates the effect of the current meta-learner on the LLM. For a mini-batch $\mathcal{B}_{train}$ from $\mathcal{D}_{train}$, we first calculate the Perplexity Difference $\text{PPLDiff}^{(t)}(x, y_1, y_2; \pi_{\theta^{(t)}})$ for each sample using the current main LLM parameters $\theta^{(t)}$ (Eq. (4)). This $\text{PPLDiff}^{(t)}$ serves as input to the current meta-learner $V(\cdot;\phi^{(t)})$, which generates soft preference labels $\hat{p}_{\phi^{(t)}}$ for the batch: $\hat{p}_{\phi^{(t)}} = V(\text{PPLDiff}^{(t)}; \phi^{(t)})$. The main LLM's preference loss for this virtual update, denoted as a manner analogous to GDPO loss:

$$\mathcal{L}_{main}^{virtual}(\theta; \phi^{(t)}, \pi_{\theta^{(t)}}) = -\frac{1}{|\mathcal{B}_{train}|} \sum_{\mathcal{B}_{train}} \log \sigma \left(\beta(2\hat{p}_{\phi^{(t)}} - 1) h_\theta(x, y_1, y_2) \right). \quad (6)$$

The virtual LLM parameters, $\theta_{virtual}(\phi^{(t)})$, are obtained by a gradient descent step on this loss, starting from $\theta^{(t)}$:

$$\theta_{virtual}(\phi^{(t)}) = \theta^{(t)} - \alpha \nabla_\theta \mathcal{L}_{main}^{virtual}(\theta; \phi^{(t)}, \pi_{\theta^{(t)}})|_{\theta=\theta^{(t)}}. \quad (7)$$

This provides a lookahead into how π_θ would adapt under $V(\cdot;\phi^{(t)})$.

STEP 2: Update Meta-learner Parameters. Next, the meta-learner parameters ϕ are refined. The performance of the virtual LLM $\pi_{\theta_{virtual}(\phi^{(t)})}$ (from STEP 1) is evaluated on a mini-batch $\mathcal{B}_{meta}$ from the clean meta-dataset $\mathcal{D}_{meta}$. This yields the

meta-loss $\mathcal{L}_{meta}(\phi^{(t)})$, calculated using the standard DPO loss (Eq. (1)):

$$\mathcal{L}_{meta}(\phi^{(t)}) = \mathbb{E}_{(x_m,y_{wm},y_{lm})\in\mathcal{B}_{meta}} [\mathcal{L}_{\text{DPO}}(\pi_{\theta_{virtual}(\phi^{(t)})}; \pi_{ref})]. \quad (8)$$

This meta-loss measures how well the soft labels generated by $V(\cdot;\phi^{(t)})$ guide the LLM towards alignment with clean preferences. The meta-learner parameters ϕ are then updated by descending this meta-loss:

$$\phi^{(t+1)} = \phi^{(t)} - \eta_{meta}\nabla_{\phi}\mathcal{L}_{meta}(\phi^{(t)}). \quad (9)$$

Efficient computation of $\nabla_{\phi}\mathcal{L}_{meta}(\phi^{(t)})$ can follow Shu et al. (2019). This step improves V's ability to produce beneficial soft labels.

STEP 3: Update Main LLM Parameters. Finally, the actual main LLM parameters $\theta^{(t)}$ are updated. This update uses the PPLDiff values (PPLDiff$^{(t)}$) calculated at the start of STEP 1 (from $\pi_{\theta^{(t)}}$) and the newly updated meta-learner $V(\cdot;\phi^{(t+1)})$ from STEP 2. This $V(\cdot;\phi^{(t+1)})$ generates a fresh set of soft labels, $\hat{p}_{\phi^{(t+1)}}$, for the training batch $\mathcal{B}_{train}$. The preference loss for this actual main LLM update, denoted $\mathcal{L}_{main}$, is then calculated using these new soft labels:

$$\mathcal{L}_{main}(\theta;\phi^{(t+1)},\pi_{\theta^{(t)}}) = -\frac{1}{|\mathcal{B}_{train}|}\sum_{\mathcal{B}_{train}} \log\sigma\left(\beta(2\hat{p}_{\phi^{(t+1)}}-1)h_{\theta}(x,y_1,y_2)\right). \quad (10)$$

The main LLM parameters are then updated by a gradient step on this loss:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha\nabla_{\theta}\mathcal{L}_{main}(\theta;\phi^{(t+1)},\pi_{\theta^{(t)}})|_{\theta=\theta^{(t)}}. \quad (11)$$

This completes one iteration t. This cycle of three steps is repeated, allowing MSPO to adaptively learn to generate effective soft preference labels. The complete procedure is detailed in Algorithm 1. Theoretically, this bilevel process enables MSPO to learn an implicit weighting scheme favoring beneficial soft labels, and its generalization performance is related to the complexity of the meta-learner and the size of the meta-dataset (see Appendix B for a detailed analysis).

3.3 Experimental Setup

This subsection outlines our core experimental configurations, including datasets, noise simulation, model architectures, the methods we compare against, and evaluation metrics.

Datasets and Noise Simulation. Our primary experiments utilize two widely-used human preference datasets: Golden HH (Bai et al., 2022a), a high-quality subset of Anthropic HH-RLHF, and OASST1 (Köpf et al., 2023), a large-scale multilingual dataset. To assess robustness, we inject random label flipping noise into their training splits, with noise ratios $r \in \{10\%, 20\%, 30\%, 40\%\}$. For a selected pair (y_w, y_l), its preference is inverted to (y_l, y_w). When methods require initial soft preference labels $\hat{p}_0$ (i.e., GDPO), these are first assigned to clean pairs (0.9 for y_w, 0.1 for y_l) and then correspondingly inverted for flipped pairs to simulate confident mislabeling. Our MSPO, as designed, generates $\hat{p}_{\phi}$ primarily from PPLDiff and does not use $\hat{p}_0$ as a direct input to its meta-learner V. The meta-dataset (D_{meta}) for training our meta-learner is a small, clean subset of approximately 100-200 high-quality preference pairs sampled from the original training data of each main dataset, ensuring no overlap with training or test sets. We primarily use its original binary preference labels for calculating the meta-loss.

Models. We use two open-source LLMs as base architectures: Llama2-7B (Touvron et al., 2023) and Phi-2 (2.7B) (Javaheripi et al., 2023), aligning with Kong et al. (2024). All methods start from a supervised fine-tuned (SFT) version of these models, which also serves as the reference policy π_{ref}. For baseline methods that require perplexity calculations from a fixed surrogate model, such as Perplexity-aware Correction (PerpCorrect), this surrogate model π_s is the SFT model, further aligned on D_{meta} using DPO for a small number of steps and then kept fixed. In contrast, our MSPO framework, as detailed in Section 3, dynamically calculates PPLDiff using the **current main LLM** $\pi_{\theta^{(t)}}$ being aligned.

Comparative Methods. We compare MSPO against several strong methods. These include standard DPO (Rafailov et al., 2023) on noisy binary preferences; GDPO($\hat{p}_0$) (Furuta et al., 2024) using initial noisy soft labels (as described above); robust DPO variants cDPO (Mitchell, 2023) and rDPO (Liang et al., 2024), with their required noise ratio ϵ' estimated from D_{meta}; and PerpCorrect+DPO (Kong et al., 2024), which applies PerpCorrect as a data pre-processing step before DPO training. Further detailed hyperparameters and implementation specifics for all methods are provided in Appendix C.

Table 1: Win Rates (%) of Llama-2-7B Against SFT on Golden HH and OASST1 Datasets under Random Label Flipping Noise. Best results are in **bold**.

	Golden HH					OASST1				
Method	Clean (0%)	10%	20%	30%	40%	Clean (0%)	10%	20%	30%	40%
Vanilla DPO	97.22	92.53	82.62	68.50	53.15	97.17	96.64	92.71	90.21	86.29
GDPO ($\hat{p} = 0.75$)	97.57	97.15	95.53	94.26	91.21	97.52	97.06	94.21	93.08	92.73
cDPO (Oracle ϵ)	97.38	96.04	90.85	83.23	65.60	97.67	96.18	93.63	90.62	88.02
rDPO	97.21	96.65	95.22	93.90	90.45	97.76	95.92	93.73	92.05	90.62
PerpCorrect-DPO	97.87	97.51	96.24	95.53	94.92	98.05	96.38	94.04	93.99	93.17
MSPO (Ours)	**98.42**	**97.94**	**96.62**	**96.32**	**96.11**	**98.67**	**97.37**	**95.35**	**94.65**	**94.42**

Table 2: Win Rates (%) of Phi-2 Against SFT on Golden HH and OASST1 Datasets under Random Label Flipping Noise. Best results are in **bold**.

	Golden HH					OASST1				
Method	Clean (0%)	10%	20%	30%	40%	Clean (0%)	10%	20%	30%	40%
Vanilla DPO	96.50	93.19	85.57	73.07	54.98	69.12	66.94	62.61	58.44	52.42
GDPO ($\hat{p} = 0.75$)	97.07	97.54	96.08	94.52	85.39	68.73	67.93	63.58	59.88	53.05
cDPO (Oracle ϵ)	97.56	97.21	92.63	81.05	66.72	69.30	67.30	61.44	54.87	49.21
rDPO	97.01	96.49	95.73	93.34	84.55	67.16	63.95	59.47	56.45	45.20
PerpCorrect-DPO	98.18	98.17	97.05	96.66	96.39	72.55	71.34	69.04	68.27	68.49
MSPO (Ours)	**98.86**	**98.39**	**98.04**	**97.49**	**97.25**	**74.83**	**72.49**	**71.22**	**70.63**	**70.01**

Evaluation Metric. The primary evaluation metric is the win rate against GPT-4[1] as an automated LLM judge, following common practice (Bai et al., 2022a; Kong et al., 2024). For a held-out set of test prompts, responses from trained models and baselines are pairwise compared. Win rates are calculated as (wins / (wins + losses)), ignoring ties, typically against the SFT model. All reported win rates are averaged over 3 independent runs with different random seeds.

3.4 Main Results: Robustness to Noisy Preferences

We evaluate the robustness of MSPO against varying levels of random label flipping noise (0% to 40%). Performance is measured by win rates against SFT models, presented in Table 1 for Llama-2-7B and Table 2 for Phi-2, across Golden HH and OASST1 datasets.

Overall, both tables demonstrate a clear trend: standard DPO and GDPO (with fixed $\hat{p} = 0.75$) suffer substantial performance degradation as noise levels increase. Robust DPO variants (cDPO, rDPO) and PerpCorrect+DPO offer notable improvements by mitigating some noise effects. However, our proposed MSPO consistently achieves the highest win rates across nearly all noise conditions, datasets, and base models.

[1]Model version `gpt-4-0613`, accessed via OpenAI API.

For instance, with Llama-2-7B on Golden HH (Table 1), MSPO maintains a win rate of 96.11% even at 40% noise, significantly outperforming DPO (53.15%) and strong baselines like PerpCorrect-DPO (94.92%). Similar superiority is observed on OASST1, where MSPO (94.42% at 40% noise) surpasses others. This robust performance underscores the effectiveness of its adaptive meta-learning mechanism in dynamically generating optimized soft preference labels.

The advantages of MSPO are further confirmed with the smaller Phi-2 model (Table 2). On Golden HH, MSPO exhibits exceptional resilience, achieving 97.25% win rate at 40% noise. While absolute win rates are generally lower on OASST1 with Phi-2 for all methods, MSPO again consistently leads, for example, achieving 70.01% at 40% noise compared to PerpCorrect-DPO's 68.49% and DPO's 52.42%.

In summary, MSPO demonstrates significant and consistent improvements in alignment robustness across different models, datasets, and high levels of preference noise. Its ability to learn an adaptive strategy for generating soft preference labels, primarily guided by PPLDiff, provides a clear advantage over methods relying on global noise estimates or heuristic data pre-processing.

Table 3: Ablation study for MSPO on Golden HH (Llama-2-7B) with 30% random label flipping noise. Win rates (%) against SFT. Best performance is in **bold**.

Method Variant	Win Rate (%)
MSPO (Full, uses PPLDiff)	**96.72**
Impact of Meta-learning Mechanism:	
w/o Meta-learning (DPO with binary labels)	68.50
Choice of Meta-learner Input Feature:	
Meta-learner uses Loss instead of PPLDiff	89.47
Impact of Meta-dataset Size ($\|D_{meta}\|$):	
D_{meta} size: 50 samples	94.33
D_{meta} size: 100 samples (Default)	96.72
D_{meta} size: 150 samples	96.75
D_{meta} size: 200 samples	96.82

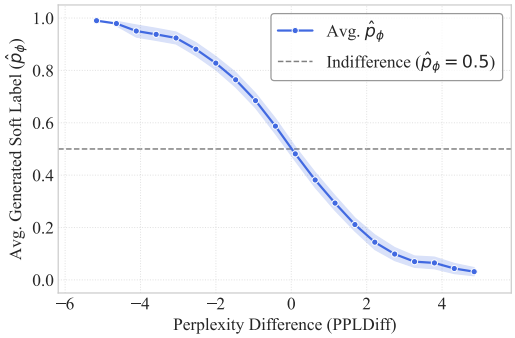

Figure 3: Average generated soft preference label ($\hat{p}_\phi$) by the MSPO meta-learner as a function of binned PPLDiff values. The dashed line indicates $\hat{p}_\phi = 0.5$ (indifference).

3.5 Ablation Studies

To understand the contributions of key components and design choices within MSPO, we conduct ablation studies on the Golden HH dataset with Llama-2-7B under 30% random label flipping noise. Results are presented in Table 3.

Effectiveness of Meta-learning. Removing the meta-learning component and reverting to standard DPO (which does not use adaptive soft labels) reduces the win rate from 96.72% to 68.50% (at 30% noise, from Table 1). This highlights the significant benefit of our adaptive meta-optimization strategy for generating soft labels compared to using fixed binary labels in a noisy environment.

Choice of Input Feature for Meta-learner. Our full MSPO model utilizes PPLDiff as the primary input feature for the meta-learner. We compare this against an alternative where the meta-learner instead uses the main alignment loss value of the current sample (from $\mathcal{L}_{main}$ before any soft label application) as its input signal. Using the loss signal results in a substantially lower win rate of 89.47%, compared to 96.72% with PPLDiff. This finding strongly supports our hypothesis that PPLDiff, an intrinsic measure of preference consistency under a surrogate model, is a more effective and informative signal for the meta-learner to generate appropriate soft labels compared to the training loss signal.

Sensitivity to Meta-dataset Size. We examine the impact of varying the size of the clean meta-dataset D_{meta}. With only 50 meta-samples, the performance is 94.33%. Increasing the size to our default of 100 samples yields 96.72%. Further increases to 150 and 200 samples result in slight improvements to 96.75% and 96.82%, respectively. This indicates that while performance benefits from more meta-data, MSPO can achieve strong results even with a very small meta-dataset, demonstrating its practical applicability. The marginal gains beyond 100-150 samples suggest that a relatively small amount of clean data is sufficient for the meta-learner to learn an effective soft label generation heuristic.

3.6 Analysis of Meta-Learner Behavior

To gain deeper insights into the mechanism of MSPO, we analyze the behavior of the trained meta-learner $V(\cdot;\phi)$. Understanding the patterns learned by the meta-learner helps validate its effectiveness in generating appropriate soft preference labels from noise-indicative signals.

Sensitivity of Generated Labels to PPLDiff. A central tenet of MSPO is that its meta-learner interprets Perplexity Difference (PPLDiff) to generate suitable soft preference labels. Figure 3 visualizes this learned mapping, showing the average soft label ($\hat{p}_\phi$) produced by our trained meta-learner as a function of binned PPLDiff values. The underlying (PPLDiff, $\hat{p}_\phi$) data was aggregated from the MSPO training of Llama-2-7B on the Golden HH dataset (30% random flip noise). Critically, PPLDiff is dynamically calculated at each relevant training step using the evolving main LLM $\pi_{\theta^{(t)}}$, with the meta-learner $V(\cdot;\phi^{(t)})$ generating the corresponding $\hat{p}_\phi$. For plotting, these PPLDiff values were binned and associated $\hat{p}_\phi$ values averaged.

The plot exhibits a clear and rational sigmoidal trend. When PPLDiff is substantially negative, indicating the nominally preferred y_1 is significantly more likely than y_2 under the main LLM's current understanding, the meta-learner assigns a high $\hat{p}_\phi$

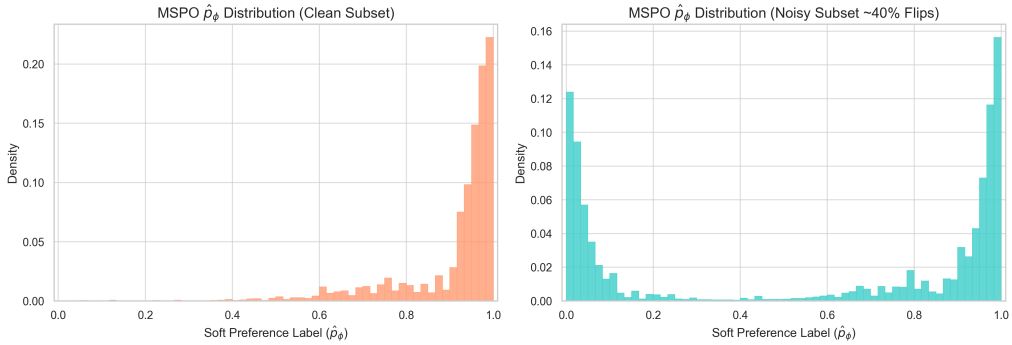

Figure 4: Comparison of naive initial ($\hat{p}_0$, for reference) and MSPO generated ($\hat{p}_\phi$) soft label distributions on Clean and Noisy (Sim. 40% True Flips) subsets.

(approaching 1.0), reflecting strong confidence in $y_1 \succ y_2$. Conversely, as PPLDiff becomes substantially positive, suggesting y_1 is less likely and the nominal preference potentially incorrect, the generated $\hat{p}_\phi$ trends towards 0.0, effectively signaling a reversed preference for mislabeled pairs. Near PPLDiff = 0, where the main LLM shows similar likelihoods, the average $\hat{p}_\phi$ is approximately 0.5, representing indifference. This adaptive and graded response demonstrates that MSPO successfully learns to translate the dynamic PPLDiff signal into nuanced, contextually appropriate soft preference labels.

Distributional Impact of Learned Soft Labels. To further elucidate the adaptive mechanism of our meta-learner, we analyze the distribution of its generated soft preference labels ($\hat{p}_\phi$). Figure 4 contrasts these $\hat{p}_\phi$ distributions on representative clean and noisy (40% label flips) subsets of the Golden HH training data.

On the **Clean Subset** (left panel), where true preferences are consistently labeled, the MSPO $\hat{p}_\phi$ distribution is characterized by a dominant and sharply-defined peak concentrated very near 1.0. This strong concentration signifies that MSPO confidently assigns high belief to preferences it deems correct, closely mirroring an ideal scenario where all labels would be 1.0. The presence of some lesser probability mass extending to slightly lower values (e.g., within the 0.7-0.95 range) can be attributed to the inherent stochasticity of PPLDiff signals and the nuanced decision boundary learned by the meta-learner, reflecting minor, data-driven variations in confidence rather than outright uncertainty for these clean examples. The density then rapidly diminishes for $\hat{p}_\phi$ values further below this high-confidence region.

The transformative impact of MSPO's adaptive labeling is most striking on the **Noisy Subset** (right panel). Here, the $\hat{p}_\phi$ distribution becomes distinctly bimodal. A sharp peak near 0.0 corresponds to the 40% of instances where MSPO identifies likely label flips (where PPLDiff contradicts the label), assigning a very low $P(y_{nominal\_winner} \succ y_{nominal\_loser})$. Concurrently, a taller, steep peak near 1.0 represents the 60% of correctly identified non-flipped preferences, for which MSPO maintains high confidence. The relative peak heights accurately reflect the underlying noise proportion. The significantly lower density in the intermediate region (e.g., 0.2-0.8) highlights MSPO's tendency to make decisive judgments, pushing labels towards extreme confidence rather than defaulting to uncertainty for many noisy samples. This bimodal characteristic compellingly demonstrates MSPO's learned capacity to differentiate reliable preferences from probable noise by interpreting signals like PPLDiff and adaptively recalibrating preference strength.

4 Conclusion

In this work, we addressed the critical challenge of noisy preferences in aligning Large Language Models. We proposed MSPO, a novel meta-learning framework that learns to adaptively generate soft preference labels. By training a meta-learner to leverage noise-indicative signals, notably perplexity differences (PPLDiff), and optimizing this meta-learner on a small clean meta-dataset, MSPO effectively generates appropriate soft labels to guide the main LLM alignment more robustly. Our extensive experiments on benchmark datasets with various LLMs and noise levels demonstrate that MSPO significantly outperforms existing DPO variants and noise-handling techniques, particularly in high-noise scenarios. This leads to more robust and reliable LLM alignment.

Limitations and Future Work. Despite its strong performance, MSPO introduces computational overhead from its meta-learning process and depends on a clean meta-dataset. The PPLDiff signal, while effective, may have limitations for subtle semantic noise. Future research could focus on more efficient meta-optimization, exploring diverse noise indicators beyond PPLDiff, and designing more advanced meta-learners. Reducing meta-data dependency and extending the framework to other preference learning paradigms or multi-dimensional soft labels are also promising avenues.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, and 109 others. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, and 1 others. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, and 1 others. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

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

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*.

Hiroki Furuta, Kuang-Huei Lee, Shixiang Shane Gu, Yutaka Matsuo, Aleksandra Faust, Heiga Zen, and Izzeddin Gur. 2024. Geometric-averaged preference optimization for soft preference labels. In *NeurIPS*, volume 37, pages 57076–57114.

Yang Gao, Dana Alon, and Donald Metzler. 2024. Impact of preference noise on the alignment performance of generative language models. *arXiv preprint arXiv:2404.09824*.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, and 1 others. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3.

W Bradley Knox and Peter Stone. 2008. Tamer: Training an agent manually via evaluative reinforcement. In *ICDL*, pages 292–297. IEEE.

Keyi Kong, Xilie Xu, Di Wang, Jingfeng Zhang, and Mohan S Kankanhalli. 2024. Perplexity-aware correction for robust alignment with noisy preferences. In *NeurIPS*, volume 37, pages 28296–28321.

Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, and 1 others. 2023. Openassistant conversations-democratizing large language model alignment. In *NeurIPS*, volume 36, pages 47669–47681.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Xize Liang, Chao Chen, Jie Wang, Yue Wu, Zhihang Fu, Zhihao Shi, Feng Wu, and Jieping Ye. 2024. Robust preference optimization with provable noise tolerance for llms. *arXiv preprint arxiv:2404.04102*.

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

Eric Mitchell. 2023. A note on dpo with noisy preferences & relationship to ipo.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952.

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 *NeurIPS*.

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. PMLR.

Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622.

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 *NeurIPS*, volume 32.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8135–8153.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Zhen Wang, Guosheng Hu, and Qinghua Hu. 2020. Training noise-robust deep neural networks via meta-learning. In *CVPR*, pages 4524–4533.

Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. 2018. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *AAAI*, volume 32.

Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. 2021. Learning to purify noisy labels via meta soft label corrector. In *AAAI*, volume 35, pages 10388–10396.

Sen Zhao, Mahdi Milani Fard, Harikrishna Narasimhan, and Maya Gupta. 2019. Metric-optimized example weights. In *ICML*, pages 7533–7542.

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

A Algorithm for MSPO Training

The training procedure for MSPO is detailed in Algorithm 1. It outlines the iterative, three-step process for updating the main LLM parameters θ and the meta-learner parameters ϕ, assuming PPLDiff is derived from the current main LLM $\pi_{\theta^{(t)}}$.

B Theoretical Analysis of MSPO

B.1 Weighting Scheme in MSPO

The meta-learning process in MSPO can be interpreted as learning an implicit weighting scheme. The meta-learner parameters ϕ are updated to minimize the meta-loss $\mathcal{L}_{\text{meta}}(\phi^{(t)})$, which is the DPO loss evaluated on a clean meta-dataset $\mathcal{D}_{\text{meta}}$ using a virtual LLM $\pi_{\theta_{\text{virtual}}(\phi^{(t)})}$. This virtual LLM itself is obtained by a one-step gradient update on the noisy training batch $\mathcal{B}_{\text{train}}$ using the virtual main loss $\mathcal{L}_{\text{main}}^{\text{virtual}}$, which incorporates soft labels $\hat{p}_{\phi^{(t)}}$ generated by the current meta-learner $V(\cdot;\phi^{(t)})$.

The update rule for ϕ is (Eq. (9)):

$$\phi^{(t+1)} = \phi^{(t)} - \eta_{\text{meta}} \nabla_{\phi} \mathcal{L}_{\text{meta}}(\phi^{(t)}) \tag{12}$$

Using the chain rule for $\nabla_{\phi}\mathcal{L}_{\text{meta}}(\phi^{(t)})$:

$$\nabla_{\phi}\mathcal{L}_{\text{meta}} = \mathbb{E}_{\mathcal{B}_{\text{meta}}}\Bigg[\nabla_{\theta_{\text{virtual}}}\mathcal{L}_{\text{DPO}}(\pi_{\theta_{\text{virtual}}(\phi^{(t)})};\pi_{\text{ref}}) \cdot \frac{d(\theta_{\text{virtual}}(\phi^{(t)}))}{d\phi^{(t)}}\Bigg]. \tag{13}$$

Algorithm 1 MSPO Training (with PPLDiff from current $\pi_{\theta^{(t)}}$ and explicit steps)

Input: Noisy training data $\mathcal{D}_{train} = \{(x, y_1, y_2)\}$, clean meta-data $\mathcal{D}_{meta}$ (approx. 100-200 samples), batch sizes N_{tr}, N_m, learning rates α, η_m.
Output: Trained LLM parameters θ.
Initialize LLM $\pi_{\theta^{(0)}}$, meta-learner $V(\cdot; \phi^{(0)})$, reference π_{ref}.
for iteration $t = 0$ **to** $T - 1$ **do**
  Sample a training mini-batch $\mathcal{B}_{train} \sim \mathcal{D}_{train}$.
  Sample a meta mini-batch $\mathcal{B}_{meta} \sim \mathcal{D}_{meta}$.
  // STEP 1: Compute Virtual LLM Parameter Update
  For each sample $i = (x^{(i)}, y_1^{(i)}, y_2^{(i)}) \in \mathcal{B}_{train}$, compute $\text{PPLDiff}^{(t)}(i; \pi_{\theta^{(t)}})$ using Eq. (4).
  Generate soft labels for $\mathcal{B}_{train}$: $\hat{P}_{\phi^{(t)}} = \{V(\text{PPLDiff}^{(t)}(i; \pi_{\theta^{(t)}}); \phi^{(t)})\}_{i \in \mathcal{B}_{train}}$.
  Compute virtual LLM parameters $\theta_{virtual}(\phi^{(t)})$ using Eq. (7) (which uses $\mathcal{L}_{main}^{virtual}$ as defined in Eq. (6)).
  // STEP 2: Update Meta-learner Parameters
  Compute meta-loss $\mathcal{L}_{meta}(\phi^{(t)})$ using $\theta_{virtual}(\phi^{(t)})$ on $\mathcal{B}_{meta}$ via Eq. (8).
  Update meta-learner parameters: $\phi^{(t+1)} \leftarrow \phi^{(t)} - \eta_m \nabla_\phi \mathcal{L}_{meta}(\phi^{(t)})$ using Eq. (9).
  // STEP 3: Update Main LLM Parameters
  Generate new soft labels for $\mathcal{B}_{train}$ using the updated meta-learner: $\hat{P}_{\phi^{(t+1)}} = \{V(\text{PPLDiff}^{(t)}(i; \pi_{\theta^{(t)}}); \phi^{(t+1)})\}_{i \in \mathcal{B}_{train}}$.
  Compute actual main LLM loss $\mathcal{L}_{main}(\theta; \phi^{(t+1)}, \pi_{\theta^{(t)}})$ using Eq. (10).
  Update main LLM parameters: $\theta^{(t+1)} \leftarrow \theta^{(t)} - \alpha \nabla_\theta \mathcal{L}_{main}(\theta; \phi^{(t+1)}, \pi_{\theta^{(t)}})|_{\theta=\theta^{(t)}}$ using Eq. (11).
end for
return $\theta^{(T)}$.

The derivative $\frac{d(\theta_{\text{virtual}}(\phi^{(t)}))}{d\phi^{(t)}}$ in Eq. (13) is given by:

$$-\alpha \nabla_\phi \nabla_\theta \mathcal{L}_{\text{main}}^{\text{virtual}}(\theta; \phi^{(t)}, \pi_{\theta^{(t)}})|_{\theta=\theta^{(t)}}. \quad (14)$$

The term $\nabla_\phi \nabla_\theta \mathcal{L}_{\text{main}}^{\text{virtual}}$ in Eq. (14) involves the gradient of the meta-learner's output $\hat{p}_{\phi^{(t)}}$ with respect to its parameters $\phi^{(t)}$, i.e., $\nabla_\phi V(\text{PPLDiff}^{(t)}; \phi^{(t)})$.

This structure implies that the meta-learner parameters ϕ are updated in a direction that rewards the generation of soft labels $\hat{p}_\phi$ which, when used to define $\mathcal{L}_{\text{main}}^{\text{virtual}}$ for training the virtual LLM on $\mathcal{B}_{\text{train}}$, lead to improved performance (lower $\mathcal{L}_{\text{DPO}}$) on the clean meta-dataset $\mathcal{B}_{\text{meta}}$. Effectively, training instances (via their PPLDiff signals) that are transformed by $V(\cdot; \phi)$ into "beneficial" soft labels (as judged by downstream performance on clean data) exert a stronger influence on the meta-learner's update. This can be viewed as an implicit, adaptive re-interpretation or re-weighting of the training preferences based on their utility for clean alignment.

B.2 Generalization Bound for MSPO

We provide a high-level generalization bound for MSPO, inspired by meta-learning analyses (Zhao et al., 2019). Let $R_{\text{clean}}(\phi)$ be the true expected performance (e.g., negative expected $\mathcal{L}_{\text{DPO}}$ on the true clean preference distribution P_{clean}) of the main LLM π_θ when its training is guided by soft labels generated by $V(\cdot; \phi)$. Let $\hat{R}_{\text{meta}}(\phi) = -\mathcal{L}_{\text{meta}}(\phi)$ be the empirical performance on the clean meta-dataset $\mathcal{D}_{\text{meta}}$ of size M. We aim to bound the generalization gap.

Assumptions:

1. Meta-learner parameters ϕ belong to a bounded space $\Phi \subset \mathbb{R}^{d_\phi}$ (where d_ϕ is the dimensionality of ϕ).
2. The DPO loss $\mathcal{L}_{\text{DPO}}$ is bounded, e.g., in $[0, B_{\text{loss}}]$.
3. The meta-dataset $\mathcal{D}_{\text{meta}}$ consists of M i.i.d. samples from P_{clean}.

Theorem (MSPO Generalization Bound - Informal): Let $\phi^* = \arg\max_{\phi \in \Phi} \hat{R}_{\text{meta}}(\phi)$ be the parameters learned by MSPO by minimizing $\mathcal{L}_{\text{meta}}(\phi)$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of $\mathcal{D}_{\text{meta}}$:

$$R_{\text{clean}}(\phi^*) \leq \hat{R}_{\text{meta}}(\phi^*) + \mathcal{O}\left(\sqrt{\frac{\text{Comp}(\mathcal{F}_\Phi) + \log(1/\delta)}{M}}\right) \quad (15)$$

where $\text{Comp}(\mathcal{F}_\Phi)$ is a measure of the complexity of the function class $\mathcal{F}_\Phi = \{(x_m, y_{wm}, y_{lm}) \mapsto$

$\mathcal{L}_{\text{DPO}}(\pi_{\theta_{\text{virtual}}(\phi)}(\dots)) \mid \phi \in \Phi\}$ (e.g., Rademacher complexity or VC dimension if applicable). For parametric models like neural networks for $V(\cdot;\phi)$, $\text{Comp}(\mathcal{F}_\Phi)$ is often related to d_ϕ.

Proof Sketch: The proof follows standard arguments:

1. For a fixed ϕ, $\hat{R}_{\text{meta}}(\phi)$ is an empirical mean. Hoeffding's inequality bounds $|R_{\text{clean}}(\phi) - \hat{R}_{\text{meta}}(\phi)|$.
2. To ensure the bound holds uniformly over all $\phi \in \Phi$, uniform convergence bounds (e.g., based on Rademacher complexity) are used.
3. The generalization gap then bounds $R_{\text{clean}}(\phi^*)$ relative to $R_{\text{clean}}(\phi_{true\_opt})$, where $\phi_{true\_opt} = \arg\max_{\phi\in\Phi} R_{\text{clean}}(\phi)$. A common final form is:

$$R_{\text{clean}}(\phi^*) \leq R_{\text{clean}}(\phi_{true\_opt}) + \mathcal{O}\left(\sqrt{\frac{\text{Comp}(\mathcal{F}_\Phi) + \log(1/\delta)}{M}}\right). \tag{16}$$

This bound indicates that as M increases, the performance of the MSPO-learned meta-learner ϕ^* on unseen clean data approaches the performance of the best possible meta-learner within the hypothesis space Φ. The complexity of the meta-learner (related to d_ϕ) also influences the required size of M. This provides theoretical justification for MSPO's adaptive label generation.

C Implementation Details

This appendix provides further details on our experimental setup, including model configurations, training hyperparameters, dataset processing, and evaluation specifics, to facilitate reproducibility.

C.1 Model Configurations

Base Large Language Models. We utilize two publicly available pre-trained language models as the foundation for our experiments:

- Llama-2-7B (Touvron et al., 2023): We use the `Llama-2-7b-chat-hf` model version.
- Phi-2 (Javaheripi et al., 2023): We use the base pre-trained version of this 2.7-billion parameter model.

All alignment methods are initialized from a Supervised Fine-Tuned (SFT) version of these base models. The SFT model also serves as the reference policy π_{ref} in all DPO-style loss calculations.

Supervised Fine-Tuning (SFT). The SFT phase is conducted on the clean training split of the respective datasets (Golden HH, OASST1) before any noise injection. We fine-tune the base LLMs for 1 epoch using a causal language modeling objective on the chosen responses (y_w) from the preference pairs, formatted as instruction-response sequences. Key SFT hyperparameters include a learning rate of 2e-5, a global batch size of 64, a weight decay of 0.01, and a cosine learning rate scheduler with a warm-up ratio of 0.03 of total training steps.

Perplexity Difference (PPLDiff) Calculation. Our MSPO framework, as detailed in Section 3, dynamically calculates PPLDiff using the **current main LLM** $\pi_{\theta^{(t)}}$ being aligned. For baseline methods that require PPLDiff calculations from a fixed surrogate model, such as Perplexity-aware Correction (PerpCorrect) (Kong et al., 2024), we employ a surrogate model π_s. This π_s is the SFT version of the respective base LLM, further aligned on a small, clean preference dataset (a subset of D_{meta} or a similar clean set) using DPO for a few steps, and then kept fixed to provide stable PPLDiff estimations for those baselines.

Meta-learner Architecture ($V(\cdot;\phi)$). The meta-learner V in MSPO is implemented as a Multi-Layer Perceptron (MLP). It consists of:

- An input layer that takes the calculated PPLDiff value (from the current $\pi_{\theta^{(t)}}$ for MSPO) as input. PPLDiff values are z-score normalized based on statistics computed from an initial portion of the training set or a held-out calibration set.
- Two hidden layers, each with 128 units and ReLU activation functions.
- An output layer with a single neuron and a sigmoid activation to ensure the generated soft label $\hat{p}_\phi$ is within the range $[0, 1]$.

C.2 Training Hyperparameters

Training hyperparameters for all DPO-style alignment methods (DPO, GDPO, cDPO, rDPO, and the main LLM component of MSPO) are kept consistent where applicable. All alignment methods are trained for 1 epoch over their respective training datasets using the AdamW optimizer (Loshchilov and Hutter, 2017). Specifics are detailed below and summarized in Table 4.

Table 4: Key Hyperparameters for LLM Alignment and MSPO Meta-learner.

Hyperparameter	LLM Alignment (π_θ)	MSPO Meta-learner ($V(\cdot;\phi)$)
Optimizer	AdamW	AdamW
Learning Rate (α, η_m)	Llama-2-7B: 1e-6 Phi-2: 5e-6	1e-4
Effective Batch Size (N_{tr}, N_m)	Main Training: 64 pairs	Meta-Update: 32 pairs
β in DPO/GDPO loss	0.1	N/A
Warm-up Steps (for LLM α)	100	N/A
Gradient Clipping (Max Norm, for LLM)	1.0	N/A
Training Epochs	1	Concurrent with LLM (1 epoch total)

C.3 Dataset Processing and Noise Injection

Dataset Splits. Standard public train/test splits are used for Golden HH and OASST1. D_{meta} is sampled from the clean training split (100 samples).

Initial Soft Label $\hat{p}_0$ Generation (for GDPO baseline). For methods requiring initial soft labels $\hat{p}_0$ (specifically GDPO in our comparisons), on clean data, we set $\hat{p}_0(y_w \succ y_l) = 0.9$ and $\hat{p}_0(y_l \succ y_w) = 0.1$. MSPO does not use $\hat{p}_0$ as input.

Noise Injection Protocol. Random label flipping is applied as described in Section 3.3. If a pair (x, y_w, y_l) is flipped to (x, y_l, y_w), for GDPO its initial soft label would become $\hat{p}_0^{flip}(y_l \succ y_w) = 0.9$. MSPO processes the (x, y_l, y_w) pair and generates its soft label from PPLDiff.

C.4 Evaluation Details

LLM Judge. We use GPT-4 (model version: `gpt-4-0613`) via the OpenAI API as our LLM judge. The prompt template is:

> You are an impartial AI assistant evaluating the quality of two anonymous responses (Response A and Response B) to a given user prompt. Please consider helpfulness, harmlessness, honesty, and overall quality. User Prompt: [User Prompt Here] Response A: [Response A Here] Response B: [Response B Here] Which response is better? (A) Response A is significantly better. (B) Response A is slightly better. (C) Response B is significantly better. (D) Response B is slightly better. (E) Both responses are of similar quality. (F) Both responses are very poor. Please choose only one option (A, B, C, D, E, or F) and briefly explain your reasoning in one or two sentences. Your choice (A-F):

For win rate calculation, options (A) and (B) constitute a win for Response A, while (C) and (D) constitute a win for Response B. Options (E) and (F) are treated as ties and excluded from win/loss counts. Response positions (A or B) are randomized to mitigate positional bias.

Test Set and Sampling. Evaluation is performed on a held-out test set of 1,000 prompts randomly sampled from the original test splits of Golden HH and OASST1. For each prompt, one response is generated from each model using nucleus sampling with $p = 0.9$ and temperature $T = 0.7$.

Statistical Significance. Reported win rates (Tables 1 and 2) are averaged over 3 independent runs. For key comparisons between MSPO and the strongest baseline under each noise condition, we perform McNemar's test, with $p < 0.05$ considered statistically significant.

C.5 Computational Resources

Experiments were conducted on a cluster with NVIDIA A100 (40GB) GPUs. The SFT phase takes approximately 2-3 hours for Llama-2-7B and 4-5 hours for Phi-2 on their respective full training sets. Training Llama-2-7B with MSPO (which includes dynamic PPLDiff calculation) for 1 epoch on Golden HH (approx. 80k pairs) takes approximately 10-12 hours on 4 A100 GPUs. Similarly, training Phi-2 with MSPO on an OASST1 subset (approx. 100k pairs) takes approximately 15-18 hours on 4 A100 GPUs.

For baseline methods requiring pre-computed PPLDiff from a fixed surrogate model (PerpCorrect), the preparation of this surrogate π_s (initial alignment on a small clean dataset) takes approximately 0.5-1 hour. The subsequent calculation of PPLDiff values for the entire training set using this fixed π_s takes an additional 1-2 hours.