Meta-Align: Robust Preference Alignment via Perplexity-aware Meta-Learning

Anonymous ACL submission

Abstract

Noisy Preferences (NPs) present a significant challenge in aligning Large Language Models (LLMs), as incorrect preference labels can substantially degrade alignment quality. However, existing strategies to mitigate NPs often face two key limitations: (1) applying global-level adjustments that result in imprecise instance-level noise handling, and (2) relying on heuristic rules that limit the capacity to adaptively optimize alignment tasks. In response to these challenges, this paper proposes Meta-Align, a novel framework designed to address the aforementioned limitations. Meta-Align pioneers a perplexity-aware meta-learning strategy for adaptive sample reweighting, with Perplexity Difference (PPLDiff) serving as a fine-grained, instance-level signal. Unlike traditional methods employing static rules, Meta-Align trains an adaptive weighting function via meta-learning. This function dynamically assigns sample weights based on their PPLDiff, guided by performance on a small, clean meta-dataset. Such a design enables precise instance-level noise modulation while optimizing the weighting strategy in an adaptive manner. Comprehensive experiments on benchmark datasets demonstrate that Meta-Align substantially outperforms state-of-the-art robust alignment methods, effectively down-weighting potentially noisy preferences while emphasizing reliable ones.

1 Introduction

Large Language Models (LLMs) show remarkable abilities in many tasks (Brown et al., 2020; Touvron et al., 2023). Aligning these models with human preferences is crucial to ensure they are helpful, harmless, and honest (Cao et al., 2021; Bai et al., 2022). This alignment often uses preference datasets, where humans or AI systems indicate preferred responses among candidates (Christiano et al., 2017; Stiennon et al., 2020; Rafailov et al., 2023b).

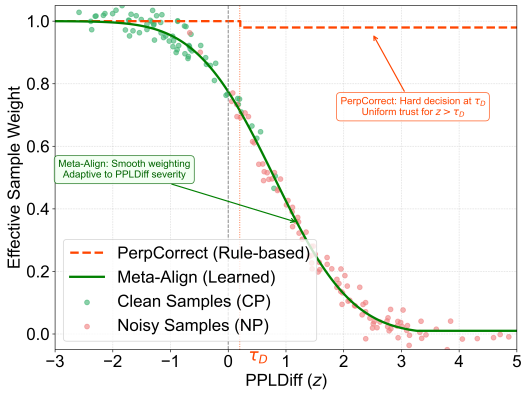


Figure 1: Conceptual comparison of PPLDiff-based weighting.

A significant challenge in this process is the presence of Noisy Preferences (NPs) within these datasets (Gao et al., 2024; Zheng et al., 2023). NPs occur when the recorded preference label is incorrect. Such noise can arise from annotator disagreement, subjective biases, or errors in AI-based labeling (Baumgärtner et al., 2024; Yi et al., 2024). Studies suggest that NPs can comprise a substantial portion, potentially 20-40% or more, of commonly used preference data (Gao et al., 2024; Rafailov et al., 2023b). Standard alignment algorithms struggle with NPs, which can lead to poor model performance, reduced alignment quality, and the reinforcement of undesirable behaviors. Therefore, developing alignment methods robust to noisy preferences is essential for building reliable LLMs.

Existing strategies to mitigate NPs often face limitations. Some apply global-level adjustments to the loss function or data (Rafailov et al., 2023a; Chowdhury et al., 2024), offering some robustness but potentially leading to imprecise instance-level noise handling due to uniform effects. A notable advancement involves using instance-specific signals like Perplexity Difference (PPLDiff) (Kong et al., 2024). PPLDiff, calculated from the LLM being aligned, can flag inconsistencies between preference labels and model likelihoods. For exam-

ple, PerpCorrect (Kong et al., 2024) uses a PPLDiff threshold to detect and flip noisy labels. This targeted approach is a step forward. Yet, these PPLDiff-based methods typically employ heuristic rules with inherent drawbacks. Reliance on hard thresholds can make them sensitive to tuning. Moreover, they often assign uniform trust to all samples identified as noisy (and subsequently corrected), overlooking varying noise severities potentially revealed by the PPLDiff signal itself. Such fixed rules also struggle to adapt as the main model and its PPLDiff calculations evolve during training.

These PPLDiff-based heuristics, despite improving upon global adjustments by using instance-level information, still lack the fine-grained adaptability needed to optimally leverage such signals. Our work, Meta-Align, directly addresses this gap. We also utilize PPLDiff, but critically, we replace fixed rules with a learned, adaptive reweighting mechanism. This makes rule-based PPLDiff methods the most relevant baseline for comparison, and Figure 1 conceptually illustrates the key differences in how PPLDiff is handled. Specifically, rule-based methods, like a PerpCorrect-inspired heuristic (dashed orange line), make sharp, discrete decisions around a PPLDiff threshold and might subsequently apply high, uniform trust to all samples identified as noisy and corrected. In contrast, Meta-Align (solid green line) embodies our proposed adaptive approach. It learns a smooth, continuous weighting function that adaptively modulates sample influence based on the PPLDiff signal, enabling a more graduated response to varying noise levels without hard cutoffs. This ability to differentiate degrees of noise severity and reduce sensitivity to any single threshold placement is central to Meta-Align.

To realize this adaptive weighting, Meta-Align employs a meta-learning strategy (Ren et al., 2018; Shu et al., 2019) instead of relying on pre-defined global adjustments or fixed heuristic rules. The core of this strategy is an adaptive weighting function that takes the PPLDiff signal—dynamically calculated from the LLM being trained—as input. This meta-learning process trains the weighting function to assign instance-specific weights to training samples, guided by performance feedback from a small, clean meta-dataset. Consequently, Meta-Align learns to automatically down-weight samples whose PPLDiff values suggest they are noisy, while up-weighting those that appear reliable, leading to a more precise and robust alignment. The main contributions of this paper are:

- We pioneer the use of meta-learning for preference alignment in large language models (LLMs), and provide theoretical analysis demonstrating its convergence advantages in the presence of noisy preference data.
- We propose Meta-Align, a novel framework that leverages a dynamically generated PPLDiff signal from the training model and a meta-learning objective to learn an instance-specific, adaptive reweighting strategy for robust alignment.
- Extensive experiments demonstrate that Meta-Align consistently outperforms existing robust alignment baselines across a wide range of noise settings by effectively down-weighting unreliable preferences and emphasizing informative ones.

2 Related Work

Our work intersects with and extends recent advances in LLM alignment, learning with noisy supervision, and meta-learning for adaptive training.

2.1 LLM Alignment with Noisy Preferences

Aligning LLMs with human values via preference data (Ouyang et al., 2022; Bai et al., 2022) is standard, using methods like RLHF (Christiano et al., 2017; Stiennon et al., 2020) and DPO (Rafailov et al., 2023b). However, these are susceptible to NPs (Gao et al., 2024; Zheng et al., 2023), which severely impair alignment. Efforts to mitigate NPs include data filtering (Northcutt et al., 2021), risking information loss; robust loss adjustments like cDPO (Rafailov et al., 2023a) and rDPO (Chowdhury et al., 2024), which apply uniform corrections based on global noise estimates; and using auxiliary signals like PPLDiff for rule-based correction (PerpCorrect (Kong et al., 2024)). Our work, Meta-Align, while inspired by PPLDiff's utility, departs from rule-based approaches by employing it within a learned, adaptive reweighting mechanism for more nuanced noise handling.

2.2 Learning with Noisy Supervision

Learning from noisy labels is a well-studied problem (Frénay and Verleysen, 2013; Song et al., 2022), with sample reweighting being a prominent paradigm (Liu and Tao, 2015; Jiang et al., 2018). This involves down-weighting likely mislabeled instances. While various heuristics or learning strategies exist to determine weights, often based on

loss values (Han et al., 2018; Shu et al., 2019), our work adapts sample reweighting to LLM preference alignment by uniquely using PPLDiff to inform weights learned via a meta-learning framework.

2.3 Meta-Learning for Adaptive Training

Meta-learning, or "learning to learn" (Ren et al., 2018), has been successfully applied to learn sample reweighting schemes for noisy classification (Ren et al., 2018; Shu et al., 2019) and data imbalance (Jamal et al., 2020), typically by optimizing weights on a clean meta-dataset. Applying this to LLM preference alignment is novel. Our **Meta-Align** framework adapts this concept, but distinctively uses the PPLDiff signal, not just training loss, as input to a meta-learned weighting function optimized for alignment quality on clean preferences. To our knowledge, Meta-Align is the first to synergize PPLDiff with meta-learning for adaptive sample reweighting in noisy LLM preference alignment, offering a data-driven, flexible alternative to heuristic or uniform correction techniques.

3 Method

We propose **Meta-Align**, a novel approach for robust LLM alignment against noisy preferences. Meta-Align uniquely leverages a dynamically computed perplexity difference signal within a meta-learning paradigm for adaptive sample reweighting. The framework aims to mitigate the negative impact of NPs in the training data $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$ by learning instance-specific weights. This learning process is guided by a small, clean meta-dataset $\mathcal{D}_{\text{meta}}$. The core idea involves using the perplexity difference, calculated by the main LLM itself during training, to inform a meta-learned weighting function $V(z; W)$. This dynamic PPLDiff signal allows the noise assessment to co-evolve with the main LLM's learning state. Figure 2 depicts the core meta-learning workflow for adaptive reweighting.

3.1 Dynamic Perplexity Difference as a Noise Indicator

A central component of Meta-Align is the use of PPLDiff as an *adaptive* indicator of potential preference noise. For a preference pair $(x^{(i)}, y_w^{(i)}, y_l^{(i)})$ from a training batch at step t, the PPLDiff is computed using the current parameters θ_t of the main LLM π_{θ_t}:

$$\begin{aligned} z_t^{(i)} &\equiv z(x^{(i)}, y_w^{(i)}, y_l^{(i)}; \theta_t) \\ &= \log \text{PPL}(\pi_{\theta_t}, [x^{(i)}; y_w^{(i)}]) \\ &\quad - \log \text{PPL}(\pi_{\theta_t}, [x^{(i)}; y_l^{(i)}]), \end{aligned} \tag{1}$$

$$\text{PPL}(\pi_{\theta_t}, s) = \exp(-\frac{1}{|s|} \sum_{k=1}^{|s|} \log \pi_{\theta_t}(s_k | s_{<k})) \tag{2}$$

where $\text{PPL}(\pi_{\theta_t}, s)$ is the perplexity of sequence s under the main policy π_{θ_t}. Intuitively, as the main LLM π_{θ_t} progressively aligns with human preferences, it should assign lower perplexity (higher probability) to genuinely preferred responses compared to rejected ones. Thus, for clean preferences (CPs), $z_t^{(i)}$ tends to be negative. Conversely, for NPs, where the chosen response $y_w^{(i)}$ is actually less preferable than $y_l^{(i)}$, $z_t^{(i)}$ tends to be positive.

Crucially, this PPLDiff signal $z_t^{(i)}$ is computed dynamically for each batch during the training of the main LLM π_θ. This ensures that the noise indicator is not static but rather adapts to the evolving understanding of preferences by the main LLM. This dynamic signal $z_t^{(i)}$ is then immediately used as an input feature to the meta-learned weighting function $V(z; W)$ within the same training iteration. This approach contrasts with methods relying on pre-computed or fixed noise scores, allowing for a tighter coupling between the main model's learning state and the sample reweighting mechanism.

3.2 Meta-Learning Adaptive Weights with Dynamic PPLDiff

Meta-Align employs the dynamically computed PPLDiff signal $z_t^{(i)}$ as input to a meta-learned weighting function $V(z; W)$, parameterized by W. This function learns to map the current PPLDiff values to non-negative sample weights $v_t^{(i)} = V(z_t^{(i)}; W)$. These weights determine the influence of each sample in the current batch during the alignment of the main LLM π_{θ_t} using a chosen alignment loss $\mathcal{L}_{\text{align}}$. The parameters W of the weighting function are optimized using a meta-learning objective defined on the clean meta-dataset $\mathcal{D}_{\text{meta}}$.

The training process, simultaneously updates the main LLM parameters θ (initialized from a base model θ_{base}) and the weighting function parameters W. At each training step t, a mini-batch

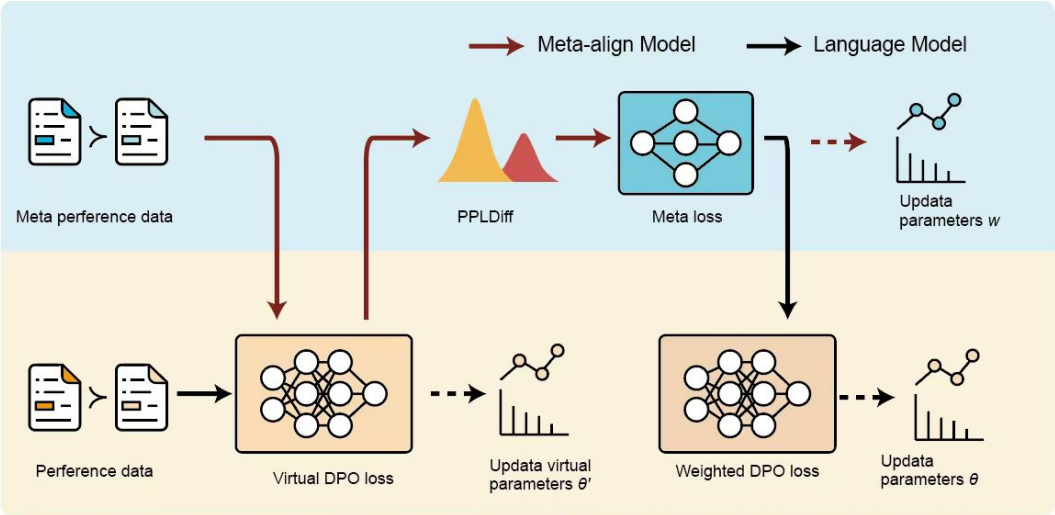


Figure 2: Overview of Meta-Align's core meta-learning loop for adaptive sample reweighting.

$\mathcal{B}_t = \{(x^{(j)}, y_w^{(j)}, y_l^{(j)})\}_{j=1}^{|\mathcal{B}_t|}$ is first sampled from the noisy training set $\mathcal{D}$. Then, for each sample j within this batch $\mathcal{B}_t$, the dynamic PPLDiff $z_t^{(j)}$ is computed using the current main LLM parameters θ_t according to Eq. (1). Concurrently, a mini-batch $\mathcal{B}_{\text{meta},t}$ is sampled from the clean meta-dataset $\mathcal{D}_{\text{meta}}$.

The current weighting function $V(\cdot; W_t)$ then uses these dynamic PPLDiff values $z_t^{(j)}$ to compute weights $v_t^{(j)}$ for all samples j in $\mathcal{B}_t$. These weights are subsequently normalized (denoted as $\tilde{v}_t^{(j)}$) for stability. The normalized weights modulate the alignment loss $\mathcal{L}_{\text{align}}$ for the training batch. To evaluate the effectiveness of the current weights W_t, we perform a *virtual update* step. We compute the weighted alignment loss $\mathcal{L}_{\text{weighted}}(\theta_t, W_t)$ on $\mathcal{B}_t$ using the dynamic weights $\tilde{v}_t^{(j)}$:

$$\mathcal{L}_{\text{weighted}}(\theta_t, W_t) = \frac{1}{|\mathcal{B}_t|} \sum_{j \in \mathcal{B}_t} \tilde{v}_t^{(j)} \mathcal{L}_{\text{align}}(\pi_{\theta_t}, \pi_{\text{ref}}, x^{(j)}, y_w^{(j)}, y_l^{(j)}). \quad (3)$$

Here, $\mathcal{L}_{\text{align}}$ represents the specific alignment loss function chosen for the main LLM (e.g., DPO, IPO), and π_{ref} is the corresponding reference policy if required by $\mathcal{L}_{\text{align}}$.

A hypothetical one-step gradient descent update using this loss yields virtual LLM parameters $\theta_t'(W_t)$:

$$\theta_t'(W_t) = \theta_t - \alpha_\theta \nabla_{\theta_t} \mathcal{L}_{\text{weighted}}(\theta_t, W_t). \quad (4)$$

Next, the quality of this virtual update, and thus the quality of the weighting parameters W_t, is assessed by evaluating the performance of the virtual model $\pi_{\theta_t'(W_t)}$ on the clean meta-batch $\mathcal{B}_{\text{meta},t}$. This yields the *meta-loss*, $\mathcal{L}_{\text{meta}}(W_t)$, calculated using the standard unweighted alignment loss $\mathcal{L}_{\text{align}}$ on the meta-data:

$$\mathcal{L}_{\text{meta}}(W_t) = \frac{1}{|\mathcal{B}_{\text{meta},t}|} \sum_{k \in \mathcal{B}_{\text{meta},t}} \mathcal{L}_{\text{align}}(\pi_{\theta_t'(W_t)}, \pi_{\text{ref}}, x_m^{(k)}, y_{mw}^{(k)}, y_{ml}^{(k)}). \quad (5)$$

The gradient of this meta-loss with respect to the weighting parameters, $\nabla_{W_t} \mathcal{L}_{\text{meta}}$, provides the signal for improving the weighting function. The weighting function parameters W are updated using this gradient:

$$W_{t+1} = W_t - \alpha_W \nabla_{W_t} \mathcal{L}_{\text{meta}}(W_t). \quad (6)$$

Finally, the actual update for the main LLM parameters θ_t is performed. This step utilizes the newly updated weighting parameters W_{t+1} to recompute weights $\tilde{v}_{\text{new},t}^{(j)} = \text{Normalize}(V(z_t^{(j)}; W_{t+1}))$ for the training batch $\mathcal{B}_t$ using the same PPLDiff values $z_t^{(j)}$ computed earlier in the step. The main LLM parameters θ_t are then updated by descending the gradient of this newly re-weighted alignment loss, $\mathcal{L}'_{\text{weighted}}(\theta_t, W_{t+1})$:

$$\theta_{t+1} = \theta_t - \alpha_\theta \nabla_{\theta_t} \left(\frac{1}{|\mathcal{B}_t|} \sum_{j \in \mathcal{B}_t} \tilde{v}_{\text{new},t}^{(j)} \mathcal{L}_{\text{align}}(\pi_{\theta_t}, \pi_{\text{ref}}, x^{(j)}, y_w^{(j)}, y_l^{(j)}) \right). \quad (7)$$

This iterative refinement allows $V(z; W)$ to learn an effective weighting strategy based on the main

LLM's own dynamically generated PPLDiff signal. This process is specifically optimized to improve alignment performance on clean data, thereby robustly handling noise in $\mathcal{D}$. The complete procedure is detailed in Appendix A. Theoretical underpinnings, including an analysis of the weighting scheme and generalization guarantees, are detailed in Appendix B.

4 Experiments

This section presents a comprehensive empirical evaluation of our proposed Meta-Align approach. Our experiments were designed to investigate its effectiveness in robustly aligning LLMs under noisy preference conditions and to understand the contributions of its core components. Specifically, we sought to determine: (1) whether Meta-Align outperforms existing vanilla and robust alignment baselines across various levels of preference noise when using DPO as the base alignment loss; (2) the individual contributions of using the dynamically computed PPLDiff as an input signal versus raw loss, and the meta-learning based reweighting mechanism itself; (3) the sensitivity of Meta-Align to the size and potential imperfections of the clean meta-dataset; (4) whether the learned weighting mechanism behaves in an interpretable manner.

4.1 Experimental Setup

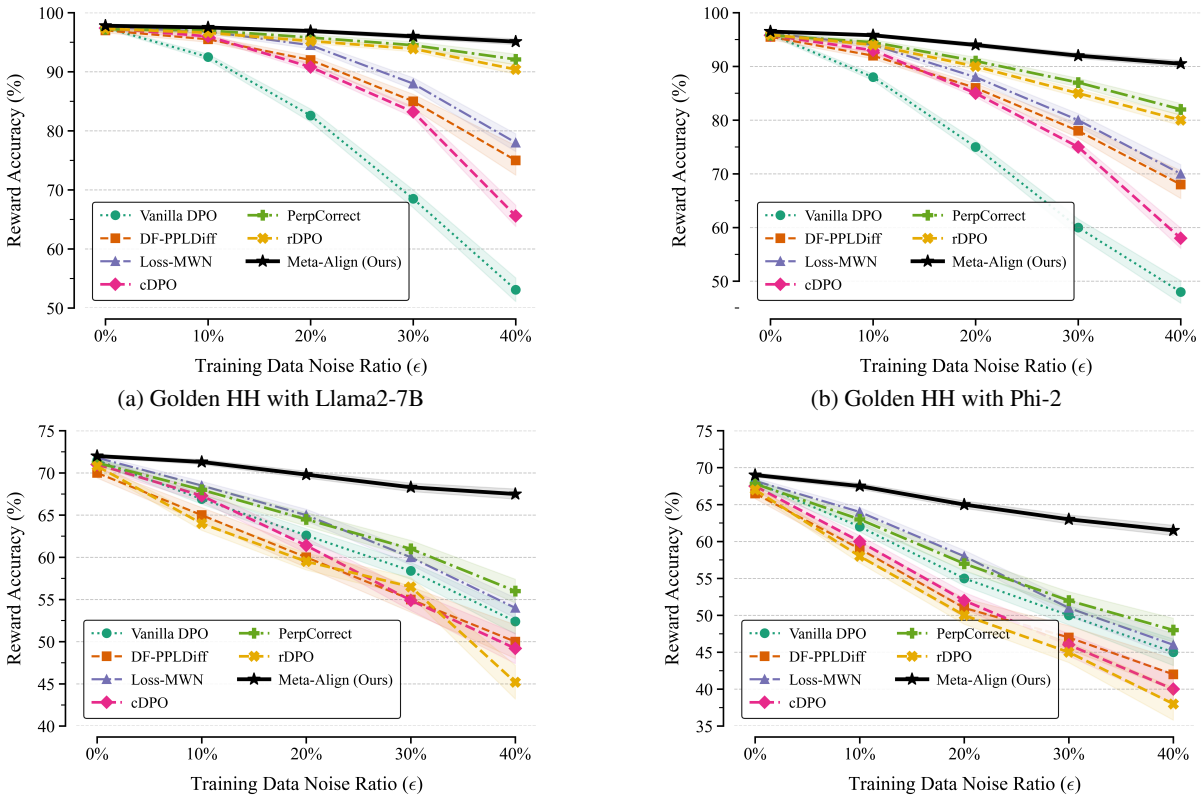
Datasets. Our approach was evaluated on two widely used public preference datasets: Golden HH (Bai et al., 2022; Ethayarajh et al., 2024), a helpfulness-focused subset of Anthropic-HH (approx. 12K train / 654 test samples), and OASST1 (Köpf et al., 2024), the OpenAssistant Conversations dataset (multi-turn dialogues), using the processed version from Rafailov et al. (2023b) (approx. 18K train / 951 test pairs).

For Meta-Align, a small subset was randomly sampled from the original *training split* of each dataset to serve as the clean meta-dataset ($\mathcal{D}_{\text{meta}}$). Unless otherwise specified, we used $M = 100$ samples for $\mathcal{D}_{\text{meta}}$. The remaining training data constituted the potentially noisy training set $\mathcal{D}$. The original *test split* was used exclusively for evaluation and was assumed to be clean. For general hyperparameter tuning of Meta-Align and baseline methods, we utilized a separate held-out clean validation set $\mathcal{D}_{\text{val}}$, also sampled from the original training split, ensuring no overlap with $\mathcal{D}$ or $\mathcal{D}_{\text{meta}}$.

Noise Injection. Following common practice (Kong et al., 2024; Chowdhury et al., 2024), we simulated noisy preferences by randomly flipping the preference labels ($y_w \leftrightarrow y_l$) of a fraction ϵ of the samples in the training set $\mathcal{D}$. We experimented with noise rates $\epsilon \in \{0\%, 10\%, 20\%, 30\%, 40\%\}$. The $\epsilon = 0\%$ setting represents training on clean data.

Models and Implementation Details. Experiments were conducted using two representative open-source LLMs: Llama2-7B (Touvron et al., 2023) and Phi-2 (Javaheripi et al., 2023). All models were initialized from their standard pre-trained or supervised fine-tuned (SFT) checkpoints where applicable. The primary alignment loss $\mathcal{L}_{\text{align}}$ used in our main comparative experiments and ablation studies was Direct Preference Optimization (DPO) (Rafailov et al., 2023b), implemented using the TRL library (von Werra et al., 2020). This choice facilitates a fair and direct comparison with prevalent robust DPO baselines. The DPO hyperparameter β was set to 0.1. For Meta-Align, the PPLDiff signal for each batch was computed dynamically using the current main LLM's parameters as described in Section 3.1. The Meta-Weight-Net $V(z; W)$ in Meta-Align was implemented as a two-layer MLP with ReLU activation and a Sigmoid output layer, ensuring output weights (before normalization) are between 0 and 1. Learning rates $(\alpha_\theta, \alpha_W)$ and other optimization details were tuned based on performance on $\mathcal{D}_{\text{val}}$ and are detailed in Appendix C. All experiments were repeated with 3 different random seeds, and we report the mean and standard deviation of the results.

Baselines. We compared Meta-Align (instantiated with DPO as $\mathcal{L}_{\text{align}}$ for these comparisons) against several methods: Vanilla DPO (Rafailov et al., 2023b); cDPO (Rafailov et al., 2023a); rDPO (Chowdhury et al., 2024); and PPLDiff-based heuristic methods including PerpCorrect (Kong et al., 2024) and Data Filtering (DF-PPLDiff). For PerpCorrect and DF-PPLDiff in our experiments, the PPLDiff signal was pre-computed using a surrogate LLM aligned on the clean validation set $\mathcal{D}_{\text{val}}$. This approach is consistent with common implementations of such heuristic methods and provides a clear contrast to Meta-Align, where the PPLDiff signal is computed dynamically by the main LLM during training. An additional ablation baseline was Standard MWN (Loss-MWN), which uses the DPO training loss as input to the meta-weighting



(a) Golden HH with Llama2-7B

(b) Golden HH with Phi-2

(c) OASST1 HH with Llama2-7B

(d) OASST1 with Phi-2

Figure 3: Reward Accuracy (%) on the Golden HH (top row) and OASST1 (bottom row) test sets, for Llama2-7B (left column) and Phi-2 (right column) models, under varying training noise rates (ϵ).

network. Further details on these baselines, including their reliance on $\mathcal{D}_{\text{val}}$ for noise estimation or PPLDiff computation, or $\mathcal{D}_{\text{meta}}$ for meta-learning, are provided where relevant in our analysis.

Evaluation Metric. Following standard practice (Rafailov et al., 2023b; Chowdhury et al., 2024), our primary evaluation metric was Reward Accuracy. An independent reward model (RM) was trained on the clean training split of each dataset (or a designated RM training set). The aligned policy π_θ was then evaluated by calculating the percentage of test set preference pairs $(x, y_w, y_l) \in \mathcal{D}_{\text{test}}$ for which the RM assigned a higher score to the human-preferred response y_w, i.e., $RM(x, y_w) > RM(x, y_l)$.

4.2 Comparative Performance with DPO

To comprehensively evaluate the efficacy of Meta-Align when instantiated with Direct Preference Optimization (DPO) as the underlying alignment loss, we compared its performance against established baselines under varying degrees of simulated preference noise. This evaluation was conducted across two diverse datasets, Golden HH and OASST1, and using two distinct model architectures, Llama2-7B and Phi-2. The results, presented as Reward Accuracy (%) versus noise rate (ϵ), are visualized in Figure 3. For baseline results of Vanilla DPO, cDPO, and rDPO, we reference performance figures reported in Kong et al. (2024) where experimental setups align, ensuring a fair comparison. Results for PerpCorrect (using pre-computed PPLDiff from a surrogate model as detailed in our setup) and our Meta-Align (using dynamically computed PPLDiff) are generated under matched conditions.

The performance trends across all four settings (Figures 3a through 3d) consistently highlight the robustness of Meta-Align. As anticipated, Vanilla DPO's accuracy sharply deteriorates with increasing noise levels (ϵ) on both datasets and for both model architectures. While robust baselines such as cDPO and rDPO offer considerable improvements, and PPLDiff-based heuristics like PerpCorrect (utilizing a static, pre-computed PPLDiff in our evaluations) also show benefits, Meta-Align consistently establishes a new state-of-the-art. It achieves the highest Reward Accuracy across all non-zero noise conditions, underscoring the advantages of its adaptive reweighting mechanism guided by a dynamic PPLDiff signal.

Specifically, on the Golden HH dataset, Meta-Align with Llama2-7B (Figure 3a) at $\epsilon = 40\%$

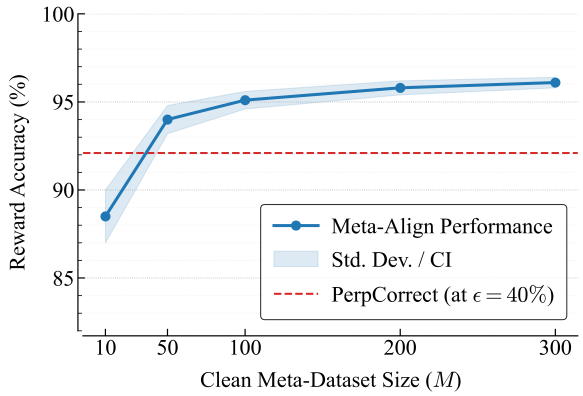


Figure 4: Reward Accuracy of Meta-Align (DPO, Llama2-7B on Golden HH, main training noise $\epsilon = 40\%$) as a function of the clean meta-dataset size (M). Performance increases with M but saturates relatively quickly.

noise achieved 95.1% ± 0.5% accuracy, a significant margin over rDPO (90.4% ± 0.8%) and PerpCorrect (92.1% ± 0.9% with static PPLDiff). A similar pattern of superiority was observed for Meta-Align with Phi-2 on Golden HH (Figure 3b), where it attained 90.5% ± 0.6% at $\epsilon = 40\%$.

The performance advantage of Meta-Align was even more pronounced on the more challenging OASST1 dataset. For Llama2-7B (Figure 3c), Meta-Align reached 67.5% ± 0.6% at $\epsilon = 40\%$, and for Phi-2 (Figure 3d), it achieved 61.5% ± 0.7% under the same high-noise condition, substantially outperforming all baseline methodologies. These comprehensive results underscore the effectiveness and robustness of the proposed Meta-Align framework when integrated with DPO, across different LLM scales and data distributions.

4.3 Ablation Studies: Dissecting Meta-Align's Efficacy

We conducted ablation studies on the Golden HH dataset to validate Meta-Align's core components, with key findings consistent across Llama2-7B and Phi-2 models, primarily referencing results from Figure 3.

First, to assess the input signal's role, we compared Meta-Align (which uses its dynamically computed PPLDiff) against Loss-MWN (which uses the DPO training loss as input to the meta-weighting network). Meta-Align consistently and significantly outperformed Loss-MWN across all non-zero noise ratios for both models; on Llama2-7B at $\epsilon = 30\%$ noise, Meta-Align achieved 96.0% accuracy versus Loss-MWN's 88.0%. This highlights the PPLDiff, specifically when computed dynamically by the main model, as a more effective noise indicator than raw alignment loss for adaptive reweighting within our framework.

Second, to demonstrate the benefit of our meta-learned adaptive weighting, we compared Meta-Align against heuristic PPLDiff-based methods: Data Filtering (DF-PPLDiff) and PerpCorrect (Kong et al., 2024). As established in our experimental setup, DF-PPLDiff and PerpCorrect in our evaluations utilize a PPLDiff pre-computed from a surrogate model. Meta-Align surpassed both heuristics across noise levels and for both models; on Llama2-7B at $\epsilon = 40\%$ noise, Meta-Align reached 95.1% accuracy compared to PerpCorrect's 92.1% (with static PPLDiff). This underscores the superiority of Meta-Align's approach, which combines meta-learned adaptive weighting with a dynamic PPLDiff signal, over rule-based utilization of a static PPLDiff signal. These ablations confirm the synergistic contributions of the dynamic PPLDiff signal and the meta-learning framework to Meta-Align's robustness.

4.4 Sensitivity to Meta-Dataset Characteristics

We further investigated the impact of the clean meta-dataset $\mathcal{D}_{\text{meta}}$ characteristics, specifically its size (M) and its potential contamination with noise, on the performance of Meta-Align (DPO). These analyses were conducted using the Llama2-7B model on the Golden HH dataset.

Impact of Meta-Dataset Size. Figure 4 illustrates the performance of Meta-Align (DPO with Llama2-7B) on Golden HH (main training data at $\epsilon = 40\%$ noise) as the size M of $\mathcal{D}_{\text{meta}}$ was varied from 10 up to 300 samples. A clear trend of improved performance was observed with increasing M, although diminishing returns became apparent. Meta-Align achieved strong performance even with a meta-dataset size of $M = 100$, significantly outperforming baselines that do not leverage such meta-guidance. Performance tended to saturate when M reached approximately 100-200 samples, suggesting that a modest amount of clean meta-data is sufficient for effective meta-learning. This finding supports the practical applicability of our method, as acquiring extensive, perfectly clean meta-datasets can be resource-intensive.

Impact of Meta-Dataset Noise. To assess Meta-Align's robustness to imperfections in the meta-dataset, we intentionally introduced label-flipping noise into $\mathcal{D}_{\text{meta}}$. For this analysis, $\mathcal{D}_{\text{meta}}$ had a

Table 1: Impact of noise rate in $\mathcal{D}_{\text{meta}}$ on Meta-Align (DPO, Llama2-7B) Reward Accuracy (%) (Golden HH, main training noise $\epsilon = 30\%$, base $M = 100$).

Meta-Noise Rate in $\mathcal{D}_{\text{meta}}$	0%	1%	3%	5%
Meta-Align Accuracy (%)	96.0 ± 0.4	95.5 ± 0.5	94.2 ± 0.6	92.5 ± 0.8

base size of $M = 100$, and we observed the performance of Meta-Align on the Golden HH dataset, where the main training data contained $\epsilon = 30\%$ noise. The results are presented in Table 1. While Meta-Align's performance naturally degraded as the noise level within the meta-set increased, the method exhibited reasonable tolerance to low levels of meta-noise, specifically up to 5%. Even when $\mathcal{D}_{\text{meta}}$ contained 5% noise, Meta-Align achieved an accuracy of $92.5\% \pm 0.8\%$. This remained substantially higher than Vanilla DPO trained on the main set with $\epsilon = 30\%$ noise, which, as observed in our main DPO comparison for Llama2-7B on Golden HH (Figure 3a), achieved approximately $68.5\% \pm 1.5\%$ accuracy. This suggests that although a clean meta-dataset is ideal, Meta-Align is not overly brittle to minor imperfections, further enhancing its practical utility.

4.5 Analysis of the Learned Weighting Mechanism

To ascertain whether Meta-Align learns a meaningful and interpretable weighting strategy, we analyzed the characteristics of the adaptive weighting function $V(z; W)$ learned by Meta-Align (DPO) with the Llama2-7B model. This analysis focused on results from training on the Golden HH dataset with an injected noise rate of $\epsilon = 30\%$.

Figure 5 plots the learned weighting function $V(z; W)$ after training, illustrating the mapping from a sample's dynamically computed PPLDiff signal (z_t) during training to its assigned normalized weight ($\tilde{v}_{\text{new},t}$). A clear and interpretable trend is evident: the function assigns substantially lower weights to samples exhibiting high positive PPLDiff values, which are strong indicators of NPs given the main model's evolving understanding. Conversely, samples with negative or near-zero PPLDiff values, characteristic of CPs, receive markedly higher weights. The transition in weights is smooth and continuous, contrasting sharply with the hard thresholding employed in heuristic methods like PerpCorrect or DF-PPLDiff (which also operate on a static PPLDiff). This allows Meta-Align to offer a more nuanced handling of samples, particularly those with intermediate or ambiguous PPLDiff signals reflecting the main model's current state.

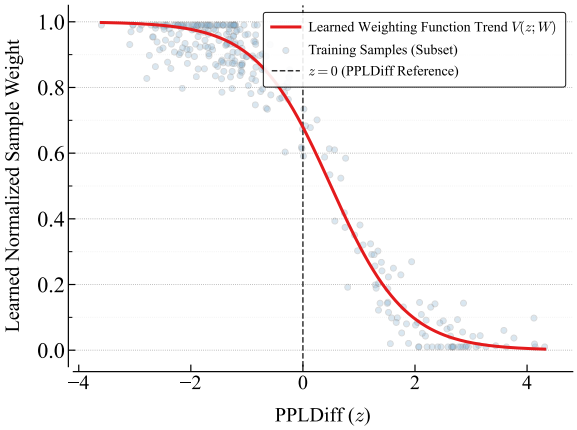


Figure 5: The weighting function learned by Meta-Align.

Furthermore, our qualitative analysis of weight assignments confirmed that samples known to be synthetically injected NPs in our simulation consistently received, on average, significantly lower weights compared to samples known to be CPs. This observation validates that the meta-learned mechanism, guided by the dynamic PPLDiff, effectively identifies and down-weights preferences likely corrupted by noise, which is pivotal to Meta-Align's robust performance. Additional analyses and visualizations are provided in Appendix D.

5 Conclusion

This work introduced Meta-Align, a novel framework for robust LLM preference alignment in the presence of noisy data. Meta-Align uniquely leverages a dynamically computed PPLDiff signal from the main LLM, synergized with a meta-learning objective, to achieve adaptive sample reweighting. Guided by a small clean meta-dataset, Meta-Align learns to effectively down-weight noisy preferences based on the LLM's evolving understanding. Extensive experiments demonstrated Meta-Align's significant outperformance over state-of-the-art baselines. Our findings highlight the efficacy of combining dynamic, instance-level noise indicators with meta-learned reweighting for robust LLM alignment.

6 Limitations

Despite its strong performance, Meta-Align has limitations. Firstly, its efficacy depends on the quality of the dynamically computed PPLDiff signal from the main LLM. If this signal is suboptimal for certain noise types or training stages, reweighting accuracy may be affected. The dynamic PPLDiff calculation also introduces computational overhead compared to static scores. Secondly, the framework relies on a clean meta-dataset, whose acquisition can be challenging, and its quality impacts meta-learning performance. Future work could explore more advanced dynamic noise indicators and strategies to reduce dependency on pristine meta-data or improve computational efficiency.

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. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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

Tim Baumgärtner, Yang Gao, Dana Alon, and Donald Metzler. 2024. Best-of-venom: Attacking rlhf by injecting poisoned preference data. In *CoLM*, pages 1–10.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *NeurIPS*, volume 33, pages 1877–1901.

Zehong Cao, KaiChiu Wong, and Chin-Teng Lin. 2021. Weak human preference supervision for deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5369–5378.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. 2024. Provably robust DPO: Aligning language models with noisy feedback. In *ICML*, pages 1–10.

Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *NeurIPS*, volume 30, pages 1–10.

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

Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

Yang Gao, Dana Alon, and Donald Metzler. 2024. Impact of preference noise on the alignment performance of generative language models. In *CoLM*, pages 1–10.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, volume 31, pages 1–10.

Ajmal Jamal, Lei Shang, Yuqi Gong, Yi-Tong Chen, Ming-Ming Cheng, and Jian Zhao. 2020. Rethinking class-balanced methods for long-tailed visual recognition. In *ECCV*, pages 1–10.

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.

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 *ICML*, pages 2304–2313.

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. 2024. Openassistant conversations-democratizing large language model alignment. In *NeurIPS*, volume 36, pages 1–10.

Tongliang Liu and Dacheng Tao. 2015. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. In *NeurIPS 2021 Track on Datasets and Benchmarks*, pages 1–10.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L 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 *NeurIPS*, volume 35, pages 27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023a. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, pages 1–10.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023b. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, volume 36, pages 53728–53741.

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

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, pages 1–10.

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 F Christiano. 2020. Learning to summarize with human feedback. In *NeurIPS*, volume 33, pages 3008–3021.

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.

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

Jingwei Yi, Rui Ye, Qisi Chen, Bin Benjamin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. 2024. Open-source can be dangerous: On the vulnerability of value alignment in open-source llms.

Sen Zhao, Mahdi Milani Fard, Harikrishna Narasimhan, and Maya Gupta. 2019. Metric-optimized example weights. In *International Conference on Machine Learning*, pages 7533–7542. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *NeurIPS*, 36:46595–46623.

A Algorithm

The core training procedure of Meta-Align, detailing the simultaneous optimization of the main LLM parameters θ and the weighting function parameters W using the dynamic PPLDiff signal, is presented in Algorithm 1.

B Theoretical Analysis

B.1 Weighting Scheme Derivation and Interpretation

The update rule for the meta-learner parameters W in Meta-Align is (Eq. (6) in main text):

$$W_{t+1} = W_t - \alpha_W \nabla_{W_t} L_{\text{meta}}(W_t) \tag{8}$$

where $L_{\text{meta}}(W_t)$ is the meta-loss on D_{meta} using virtual LLM parameters $\theta'_t(W_t)$. These are obtained by (Eq. (4) in main text):

$$\theta'_t(W_t) = \theta_t - \alpha_\theta \nabla_{\theta_t} L_{\text{weighted}}(\theta_t, W_t) \tag{9}$$

The meta-loss is $L_{\text{meta}}(W_t) = L_{\text{align}}(\pi_{\theta'_t(W_t)}, D_{\text{meta}})$. Using the chain rule for $\nabla_{W_t} L_{\text{meta}}(W_t)$:

$$\begin{aligned}\nabla_{W_t} L_{\text{meta}}(W_t) =& \nabla_{\theta'_t(W_t)} L_{\text{align}}(\pi_{\theta'_t(W_t)}, D_{\text{meta}}) \\ & \cdot \frac{d(\theta'_t(W_t))}{d(W_t)}\end{aligned} \tag{10}$$

From $\theta'_t(W_t)$'s definition, assuming θ_t is fixed for this partial derivative:

$$\begin{aligned}\frac{d(\theta'_t(W_t))}{d(W_t)} &= \frac{d(\theta_t - \alpha_\theta \nabla_{\theta_t} L_{\text{weighted}}(\theta_t, W_t))}{d(W_t)} \\ &= -\alpha_\theta \nabla^2_{W_t,\theta_t} L_{\text{weighted}}(\theta_t, W_t)\end{aligned} \tag{11}$$

This $\nabla^2_{W_t,\theta_t} L_{\text{weighted}}$ is a second-order derivative term. The update for W involves:

$$\begin{aligned}W_{t+1} =& W_t \\ & + \alpha_W \alpha_\theta \left[\nabla_{\theta'_t(W_t)} L_{\text{align}}(\pi_{\theta'_t(W_t)}, D_{\text{meta}})\right] \\ & \cdot \left[\nabla^2_{W_t,\theta_t} L_{\text{weighted}}(\theta_t, W_t)\right]\end{aligned} \tag{12}$$

This update rule is analogous to those in meta-learning for re-weighting or label correction (Ren et al., 2018). The term $\nabla_{\theta'_t(W_t)} L_{\text{align}}$ reflects meta-loss sensitivity to virtual model parameters. The term $\nabla^2_{W_t,\theta_t} L_{\text{weighted}}$ (or an approximation) reflects

Algorithm 1 Meta-Align Algorithm (with Dynamic PPLDiff)

Require: Noisy data $\mathcal{D}$, clean meta-data $\mathcal{D}_{\text{meta}}$; Base LLM parameters θ_{base}, reference policy π_{ref} (if required by $\mathcal{L}_{\text{align}}$); Learning rates α_θ, α_W; Total main training steps T_{main}; Alignment loss function $\mathcal{L}_{\text{align}}$.

Ensure: Aligned LLM parameters $\theta_{T_{\text{main}}}$.

Meta-Learning Weighted Alignment with Dynamic PPLDiff
Initialize main LLM parameters $\theta_0 \leftarrow \theta_{\text{base}}$.
Initialize weighting function parameters W_0.
Initialize optimizers Opt_θ (for θ) and Opt_W (for W).
for $t = 0$ to $T_{\text{main}} - 1$ **do**
  Sample mini-batch $\mathcal{B}_t = \{(x^{(j)}, y_w^{(j)}, y_l^{(j)})\}_{j=1}^{|\mathcal{B}_t|} \subset \mathcal{D}$.
  # Dynamically compute PPLDiff for the current batch using π_{θ_t}
  For each sample $j \in \mathcal{B}_t$, compute $z_t^{(j)} \leftarrow z(x^{(j)}, y_w^{(j)}, y_l^{(j)}; \theta_t)$ using Eq. (1).
  Sample mini-batch $\mathcal{B}_{\text{meta},t} = \{(x_m^{(k)}, y_{mw}^{(k)}, y_{ml}^{(k)})\}_{k=1}^{|\mathcal{B}_{\text{meta},t}|} \subset \mathcal{D}_{\text{meta}}$.
  Compute weights $v_t^{(j)} \leftarrow V(z_t^{(j)}; W_t)$ for $j \in \mathcal{B}_t$.
  Normalize weights: $\tilde{v}_t^{(j)} \leftarrow \text{Normalize}(\{v_t^{(j')}\}_{j' \in \mathcal{B}_t})$ for $j \in \mathcal{B}_t$.
  Compute weighted alignment loss $\mathcal{L}_{\text{weighted}}(\theta_t, W_t)$ on $\mathcal{B}_t$ using Eq. (3) (with $\tilde{v}_t^{(j)}$).
  Compute virtual LLM parameters $\theta_t'(W_t) \leftarrow \theta_t - \alpha_\theta \nabla_{\theta_t} \mathcal{L}_{\text{weighted}}(\theta_t, W_t)$ using Eq. (4).
  Compute meta-loss $\mathcal{L}_{\text{meta}}(W_t)$ on $\mathcal{B}_{\text{meta},t}$ using virtual parameters $\theta_t'(W_t)$ via Eq. (5).
  Update weighting function parameters: $W_{t+1} \leftarrow Opt_W(W_t, \nabla_{W_t} \mathcal{L}_{\text{meta}}(W_t))$ using Eq. (6).
  # Recompute weights using updated W_{t+1} and the same $z_t^{(j)}$ from this step
  Recompute weights $v_{\text{new},t}^{(j)} \leftarrow V(z_t^{(j)}; W_{t+1})$ for $j \in \mathcal{B}_t$.
  Normalize weights: $\tilde{v}_{\text{new},t}^{(j)} \leftarrow \text{Normalize}(\{v_{\text{new},t}^{(j')}\}_{j' \in \mathcal{B}_t})$ for $j \in \mathcal{B}_t$.
  Compute newly re-weighted alignment loss $\mathcal{L}'_{\text{weighted}}(\theta_t, W_{t+1})$ on $\mathcal{B}_t$ using $\tilde{v}_{\text{new},t}^{(j)}$.
  Update main LLM parameters: $\theta_{t+1} \leftarrow Opt_\theta(\theta_t, \nabla_{\theta_t} \mathcal{L}'_{\text{weighted}}(\theta_t, W_{t+1}))$ using Eq. (7) (with $\tilde{v}_{\text{new},t}^{(j)}$).
end for
return $\theta_{T_{\text{main}}}$.

how the training loss gradient w.r.t. θ_t is influenced by W_t.

Interpretation: The meta-learning objective adjusts W_t so that re-weighted training samples guide θ_t towards a θ_t' performing well on D_{meta}. If a weighting choice improves L_{meta}, it is reinforced; otherwise, it is penalized. This process learns to up-weight "helpful" samples and down-weight "harmful" (likely noisy) ones for better generalization.

B.2 Generalization Bound

We provide a generalization bound for Meta-Align, inspired by (Zhao et al., 2019) and standard learning theory. Let $R(W) = \mathbb{E}_{(x,y)\sim P_{\text{clean}}}[L_{\text{align}}(\pi_{\theta^*(W)}, (x, y))]$ be the true risk on P_{clean}, where $\theta^*(W)$ are LLM parameters learned using weights W. Let $\hat{R}_{\text{meta}}(W) = \frac{1}{M}\sum_{(x_i,y_i)\in D_{\text{meta}}} L_{\text{align}}(\pi_{\theta^*(W)}, (x_i, y_i))$ be the empirical risk on D_{meta} of size M. Let $\mathcal{W}$ be the hypothesis space for parameters W of $V(z; W)$. Let $W^* = \arg\min_{W\in\mathcal{W}} R(W)$ and $\hat{W} = \arg\min_{W\in\mathcal{W}} \hat{R}_{\text{meta}}(W)$.

Assume L_{align} is bounded by B_L. Let $\mathfrak{R}_M(\mathcal{F}_W)$ be the Rademacher complexity of the function class $\mathcal{F}_W = \{(x, y) \mapsto L_{\text{align}}(\pi_{\theta^*(W)}, (x, y)) \mid W \in \mathcal{W}\}$. Using standard generalization bounds, with probability at least $1 - \delta$:

$$\sup_{W\in\mathcal{W}} |R(W) - \hat{R}_{\text{meta}}(W)| \le 2\mathfrak{R}_M(\mathcal{F}_W) + B_L\sqrt{\frac{\ln(2/\delta)}{2M}} \tag{13}$$

This implies:

$$R(\hat{W}) \le \hat{R}_{\text{meta}}(\hat{W}) + 2\mathfrak{R}_M(\mathcal{F}_W) + B_L\sqrt{\frac{\ln(2/\delta)}{2M}} \tag{14}$$

Since $\hat{R}_{\text{meta}}(\hat{W}) \le \hat{R}_{\text{meta}}(W^*)$ by definition of

$\hat{W}$:

$$R(\hat{W}) \leq \hat{R}_{\text{meta}}(W^*) + 2\mathfrak{R}_M(\mathcal{F}_W) + B_L\sqrt{\frac{\ln(2/\delta)}{2M}} \tag{15}$$

And using the bound for W^*:

$$\hat{R}_{\text{meta}}(W^*) \leq R(W^*) + 2\mathfrak{R}_M(\mathcal{F}_W) + B_L\sqrt{\frac{\ln(2/\delta)}{2M}} \tag{16}$$

Combining (15) and (16):

$$R(\hat{W}) \leq R(W^*) + 4\mathfrak{R}_M(\mathcal{F}_W) + 2B_L\sqrt{\frac{\ln(2/\delta)}{2M}} \tag{17}$$

The complexity $\mathfrak{R}_M(\mathcal{F}_W)$ depends on $V(z;W)$ and its interaction with LLM training. $\mathfrak{R}_M(\mathcal{F}_W)$ is bounded by $O(\sqrt{d/M})$. Thus, Eq. (17) suggests $R(\hat{W}) \leq R(W^*)+O(\sqrt{d/M})$. This bound shows that $R(\hat{W})$ approaches $R(W^*)$ as M increases, if meta-weight-net complexity (related to d) is controlled.

C Implementation Details

This appendix provides further details on the hyperparameters used in our experiments, specifics of our implementation, and additional analyses to support our findings and ensure reproducibility.

C.1 Dataset Preprocessing and Splits

The public preference datasets, Golden HH (Bai et al., 2022; Ethayarajh et al., 2024) and OASST1 (Köpf et al., 2024) (using the version processed by Rafailov et al. (2023b)), underwent minimal further preprocessing beyond standard tokenization. For Meta-Align, the clean meta-dataset $\mathcal{D}_{\text{meta}}$ was constructed by randomly sampling $M = 100$ preference pairs from the original training split of each dataset, unless stated otherwise. The clean validation set $\mathcal{D}_{\text{val}}$, used for tuning hyperparameters for Meta-Align and certain baselines, comprised 300 randomly sampled preference pairs from the original training split, ensuring no overlap with the main training data $\mathcal{D}$, $\mathcal{D}_{\text{meta}}$, or the test set $\mathcal{D}_{\text{test}}$. The remaining portion of the original training split formed the potentially noisy training set $\mathcal{D}$ for our experiments.

C.2 Main LLM Alignment: Meta-Align and Baselines

For the alignment of the main LLM π_θ, both for Meta-Align and the DPO-based baselines, experiments were conducted with consistent base configurations to ensure fair comparisons. The chosen alignment loss for these primary experiments was DPO ($\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{DPO}}$), with $\beta = 0.1$. The reference policy π_{ref} was the initial SFT checkpoint of the respective LLM.

For **Meta-Align**, the PPLDiff signal $z_t^{(i)}$ was computed dynamically for each batch using the current main LLM parameters θ_t as described in Section 3.1. The Meta-Weight-Net $V(z;W)$ consisted of a two-layer MLP: an input layer processing the PPLDiff z (1 neuron), a hidden layer with 100 neurons and ReLU activation, and a Sigmoid output layer (1 neuron) producing weights $v \in [0,1]$. These raw weights $v_t^{(i)}$ were normalized within each mini-batch $\mathcal{B}_t$ by dividing by their sum: $\tilde{v}_t^{(i)} = v_t^{(i)} / \sum_{j\in\mathcal{B}_t} v_t^{(j)}$. The learning rate for the main LLM parameters (α_θ) was 5×10^{-6} for Llama2-7B and 1×10^{-5} for Phi-2, while the learning rate for the Meta-Weight-Net parameters (α_W) was 1×10^{-4}. Training was performed with a batch size ($\mathcal{B}_t$) of 16 for Phi-2 and 8 for Llama2-7B, and a meta-batch size ($\mathcal{B}_{\text{meta},t}$) of 16 (or $|\mathcal{D}_{\text{meta}}|$ if $M \leq 32$). Training proceeded for approximately one epoch over the noisy training set $\mathcal{D}$. Both θ and W were optimized using AdamW with a weight decay of 0.01.

The **baselines** (Vanilla DPO, cDPO, rDPO, PerpCorrect, DF-PPLDiff) shared the same main LLM learning rate, batch size, and training duration as Meta-Align where applicable. For PerpCorrect and DF-PPLDiff, as detailed in Section 4.1, the PPLDiff signal was pre-computed using a surrogate LLM aligned on $\mathcal{D}_{\text{val}}$. The threshold τ for DF-PPLDiff was selected from the 10th to 90th percentiles of these pre-computed PPLDiff values on $\mathcal{D}_{\text{val}}$. The Loss-MWN baseline utilized the same Meta-Weight-Net architecture and learning rate α_W as Meta-Align, with DPO loss as its input.

For the generalizability study with **IPO**, Meta-Align (IPO) used IPO as $\mathcal{L}_{\text{align}}$. The IPO-specific hyperparameter κ was set to 0.05. PPLDiff was computed dynamically. Other Meta-Align hyperparameters (e.g., α_θ, α_W) were kept consistent with the DPO setup or fine-tuned on $\mathcal{D}_{\text{val}}$. Vanilla IPO was trained with a learning rate of 5×10^{-6}.

C.3 Reward Model (RM) for Evaluation

The independent reward model (RM), pivotal for calculating Reward Accuracy, was trained on the entirety of the clean training split for each dataset (Golden HH: approx. 12K samples; OASST1: approx. 18K samples). The RM architecture was initialized from the same base SFT checkpoint as the policy models and included a final linear layer to output a scalar reward. Training employed a standard pairwise preference ranking loss, a learning rate of 1×10^{-5}, a batch size of 4, and proceeded for 1 epoch using the AdamW optimizer with a weight decay of 0.01. This RM remained fixed during the evaluation of all aligned policy models.

C.4 Computational Resources

All experiments were conducted on NVIDIA A100 GPUs. Training Meta-Align for one epoch on the Golden HH dataset with Llama2-7B typically required approximately 8 hours on a single GPU, while the standard DPO baselines took about 6 hours. The dynamic computation of PPLDiff in Meta-Align contributes to a moderate increase in training time per epoch compared to methods using pre-computed scores or no PPLDiff signal.

D Additional Analysis

D.1 PPLDiff Distribution

To further illustrate the efficacy of PPLDiff as a noise indicator, Figure 6 visualizes the distribution of PPLDiff values for samples known to be CPs versus those synthetically injected as NPs. This analysis was performed on data simulating the Golden HH dataset with an injected noise rate of $\epsilon = 30\%$. The PPLDiff values for this visualization were computed using the main LLM after it had undergone some initial alignment steps, to reflect the dynamic nature of the signal used by Meta-Align. As depicted, the PPLDiff distribution for CPs is concentrated primarily in the negative regime, with a peak density around $z \approx -1.5$. In stark contrast, the distribution for injected NPs is clearly shifted towards positive PPLDiff values, exhibiting a broader spread with a peak density around $z \approx 2.0$. A notable, albeit small, overlap exists between the tails of the two distributions, particularly around the $z = 0$ reference line. Nonetheless, this clear separation in the primary modes of the distributions underpins the utility of PPLDiff as a strong discriminative feature for our Meta-WeightNet, enabling it to distinguish and subsequently reweight potentially noisy samples based on this dynamically generated signal.

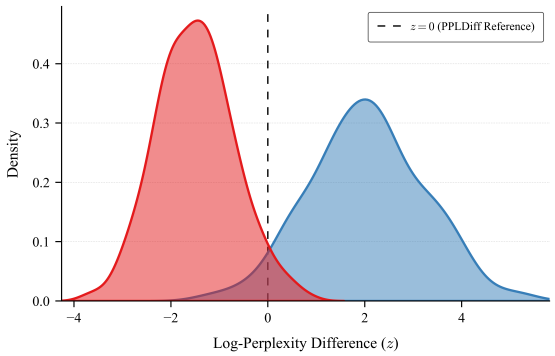


Figure 6: Distribution of PPLDiff (CPs in blue, injected NPs in red) on simulated Golden HH data ($\epsilon = 30\%$). Values were computed using the main LLM after initial alignment steps, reflecting Meta-Align's dynamic signal. NPs show notably higher PPLDiff.

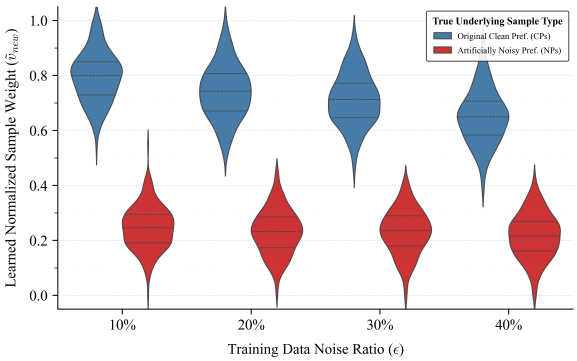


Figure 7: Distribution of learned sample weights (normalized $\tilde{v}_{\text{new}}$) assigned by Meta-Align to original Clean Preferences (CPs, blue) and synthetically created Noisy Preferences (Injected NPs, red) under varying training data noise ratios (ϵ) on the Golden HH dataset. Boxes (or violins) illustrate the distribution, showing Meta-Align adaptively assigns lower weights to injected NPs.

D.2 Adaptivity of Learned Weights to Varying Noise Ratios

To further investigate the adaptive nature of the weighting mechanism learned by Meta-Align, we analyzed the distribution of final normalized sample weights ($\tilde{v}^{(i)}_{\text{new},t}$) assigned to known CPs and synthetically injected NPs across different overall training data noise ratios (ϵ). This analysis was conducted on the Golden HH dataset.

Figure 7 presents box plots of these learned weights. A consistent pattern is evident: Meta-Align assigns significantly higher weights to samples that are genuinely CPs across all tested noise ratios from $\epsilon = 10\%$ to $\epsilon = 40\%$. For exam-

ple, at $\epsilon = 10\%$, the median weight for CPs is approximately 0.75, and this remains relatively high even as noise increases, being around 0.6 at $\epsilon = 40\%$. Conversely, samples synthetically labeled as NPs consistently receive substantially lower weights; their median weight starts around 0.23 at $\epsilon = 10\%$ and stays within a low range (around 0.17 at $\epsilon = 40\%$). The interquartile ranges for CPs and NPs show minimal overlap, especially at lower to moderate noise ratios, clearly indicating that Meta-Align's weighting function, informed by the dynamic PPLDiff, effectively learns to differentiate between reliable and likely corrupted preference signals. This adaptive down-weighting of suspicious samples is crucial for maintaining robust alignment performance in noisy environments.

D.3 Generalizability to Other Alignment Algorithms

To provide initial empirical evidence supporting this generalizability, we conducted experiments applying Meta-Align to Identity Preference Optimisation (IPO) (Azar et al., 2024). These experiments were performed on the Golden HH dataset with a training noise rate of $\epsilon = 30\%$. Meta-Align (IPO) utilized its standard architecture with dynamically computed PPLDiff, with IPO serving as the underlying alignment loss $\mathcal{L}_{\text{align}}$. We evaluated Meta-Align (IPO) against a Vanilla IPO baseline. The results, presented in Table 2, show that Meta-Align (IPO) substantially improved Reward Accuracy for both Llama2-7B (75.8% vs. Vanilla IPO's 60.3%) and Phi-2 (72.5% vs. Vanilla IPO's 58.1%), suggesting the versatility of our PPLDiff-guided adaptive reweighting approach.

Table 2: Reward Accuracy (%) on Golden HH test set ($\epsilon = 30\%$ training noise) for IPO-based methods. Mean $\pm$ Std over 3 runs. Meta-Align (IPO) uses dynamic PPLDiff.

Method	Model Architecture	
	Llama2-7B	Phi-2
Vanilla IPO	60.3 ± 1.2	58.1 ± 1.5
Meta-Align (IPO)	$\mathbf{75.8 \pm 0.9}$	$\mathbf{72.5 \pm 1.1}$