META-ALIGNER: BIDIRECTIONAL PREFERENCE-POLICY OPTIMIZATION FOR MULTI-OBJECTIVE LLMS ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-Objective Alignment aims to align Large Language Models (LLMs) with diverse and often conflicting human values by optimizing multiple objectives simultaneously. Existing methods predominantly rely on static preference weight construction strategies. However, rigidly aligning to fixed targets discards valuable intermediate information, as training responses inherently embody valid preference trade-offs even when deviating from the target. To address this limitation, we propose MEAL, i.e., MEta ALigner, a bi-level meta-learning framework enabling bidirectional optimization between preferences and policy responses, generating instructive dynamic preferences for steadier training. Specifically, we introduce a preference-weight-net as a meta-learner to generate adaptive preference weights based on input prompts and update the preference weights as learnable parameters, while the LLM policy acts as a base-learner optimizing response generation conditioned on these preferences with rejection sampling strategy. Extensive empirical results demonstrate that our method achieves superior performance on several multi-objective benchmarks, validating the effectiveness of the dynamic bidirectional preference-policy optimization framework.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks (Liang et al., 2024; Wang et al., 2023), serving as powerful tools for reasoning, coding, and creative generation (Xu et al., 2025; Jiang et al., 2024). However, ensuring that LLMs align with human values and intentions, such as helpfulness, honesty, and harmlessness, remains a fundamental challenge in the development of responsible AI systems. The primary paradigm for addressing this challenge is Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). In the standard RLHF pipeline, a reward model is first trained to approximate human preferences, and the LLM is subsequently fine-tuned to maximize the expected reward using algorithms like PPO (Schulman et al., 2017).

This kind of approach oversimplifies the complexity of real-world applications, where LLMs are unavoidably required to satisfy diverse and often conflicting values simultaneously, such as balancing helpfulness with harmlessness, or navigating varying standards of verbosity and creativity. Recently, there has been growing interest in Multi-Objective Alignment, which aims to equip models with the capability to navigate trade-offs between competing objectives and adapt to diverse user preference profiles (Li et al., 2020; Rame et al., 2023; Yang et al., 2024). Early attempts relied on scalarization techniques, which aggregate multiple reward models using fixed weights to derive a single optimization target (Li et al., 2020). However, these static weighting schemes are inflexible and fail to accommodate the diversity of user preferences in real-world scenarios. To mitigate this, contemporary studies have shifted towards test-time adaptation. Prominent approaches encompass parameter interpolation (Rame et al., 2023; Shi et al., 2024), prompt-conditioned generation (Yang et al., 2024; Wang et al., 2024), and preference-aware reward modeling (Lin et al., 2025).

However, existing multi-objective alignment methods predominantly rely on static preference weight construction strategies to guide model optimization. During training, generated responses may deviate from specified targets, yet they inherently embody valid preference trade-offs. Rigidly forcing

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

alignment to static targets—especially in SFT-based methods—discards this valuable intermediate information. Therefore, rigid unidirectional optimization fails to capture the rich mapping between the policy's evolving responses and their underlying preferences. To address the limitation, we consider a more flexible bidirectional optimization paradigm, where the preference weights and the policy responses can dynamically adapt to each other during training, allowing the model to iteratively refine its preference understanding and improve the alignment performance.

Motivated by the above consideration, we propose MEAL, i.e., MEta ALigner, a bi-level optimization framework that jointly trains a preference-weight-net and the LLM policy to achieve dynamic preference-policy alignment. The preference-weight-net serves as a meta-learner that generates adaptive preference weights based on the input prompt, while the LLM policy acts as a base-learner that optimizes its response generation conditioned on these preferences. Unlike prior methods that enforce a static mapping from fixed preference weights to responses, our framework establishes a dynamic feedback loop: the preference-weight-net learns to adjust its weight generation strategy based on the policy's evolving performance, while the policy continuously refines its response generation in light of the updated preferences. This bidirectional interaction enables the model to capture intermediate preference states, providing steadier instructive guidance during training Our main contributions are summarized as follows:

1. We propose MEAL, a bi-level meta-learning framework that enables bidirectional optimization between preferences and policy responses, mitigating the loss of valuable intermediate preference information inherent in static weighting methods.
2. We develop an outer-loop adaptation strategy that updates the preference-weight-net based on the policy's generated responses, continuously adapting to the evolving state of the policy during training.
3. We incorporate the preference-weight-net within an online learning paradigm, where the model generates responses conditioned on dynamically produced preferences and progressively improves via high-scoring sample selection.

Extensive empirical results show that our method achieves better performance on complex multi-objective benchmarks, validating the effectiveness of the dynamic bidirectional preference-policy optimization framework.

2 RELATED WORK

Aligning Large Language Models (LLMs) with human values is a fundamental prerequisite for building safe and reliable AI systems (Achiam et al., 2023; Chen et al., 2025). The dominant paradigm, Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022), typically involves training a reward model to approximate human preferences and subsequently optimizing the policy via algorithms like PPO (Schulman et al., 2017). Despite its success, standard RLHF is often computationally expensive and unstable (Dong et al., 2023; Yuan et al., 2023). To mitigate these issues, recent advancements have shifted towards Direct Preference Optimization (DPO) (Rafailov et al., 2023), which optimizes the policy directly on preference data without an explicit reward model. Building on this, researchers have developed more efficient variants to further enhance training stability and performance, such as SimPO (Meng et al., 2024), KTO (Ethayarajh et al., 2024), ORPO (Hong et al., 2024), and IPO (Garg et al., 2025).

While the aforementioned approaches have significantly improved alignment efficiency, they predominantly model human preference as a single scalar reward. This simplification overlooks the complexity of real-world applications, where LLMs must simultaneously satisfy diverse and often conflicting objectives, such as balancing helpfulness with harmlessness or navigating varying stylistic constraints (Liang et al., 2024). Consequently, a single optimization target is insufficient to capture the Pareto frontier of human values. Early attempts to address this challenge relied on scalarization (Li et al., 2020), which aggregates multiple reward models using fixed weights.

To overcome the limitations of fixed scalarization, recent literature has explored more dynamic multi-objective alignment strategies. Extending the success of direct preference learning, methods like MODPO (Zhou et al., 2024) and CPO (Guo et al., 2024) adapt DPO-style objectives to multi-objective scenarios, avoiding the instability of RL-based optimization. Another growing line of research focuses on test-time adaptation to accommodate diverse user preferences. Notable strategies

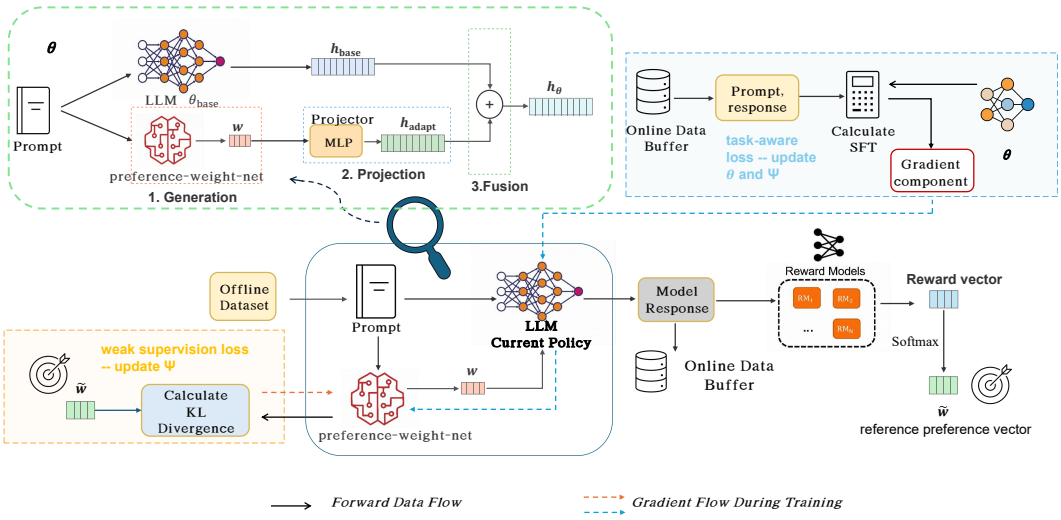


Figure 1: Illustration of MEAL

include interpolating parameters of expert models, such as Rewarded Soups (Rame et al., 2023) and MOD (Shi et al., 2024), or controlling generation via prompt conditioning and rejection sampling, as seen in Rewards-in-Context (RIC) (Yang et al., 2024) and DPA (Wang et al., 2024). More recently, PARM (Lin et al., 2025) proposed a preference-aware autoregressive reward model to guide generation dynamically.

There are two closely related works study the same general problem of multi-objective LLM alignment: one focuses on prompt-aware preference inference (Liu et al., 2025), and the other on correlation-aware objective modeling (Anonymous, 2027). This paper is technically distinct, focusing on bi-level meta-learning for bidirectional preference-policy optimization. Accordingly, we share overlapping benchmarks, evaluation protocols, baselines, and reported baseline results with them for fair comparison.

3 PRELIMINARY

We begin by formally defining the notation for the LLM alignment task. Let $\mathcal{V}$ denote the vocabulary of the language model. The generation process is modeled by a policy $\pi : \mathcal{X} \to \mathcal{Y}$, which maps an input query $x \in \mathcal{X}$ to a corresponding response $y \in \mathcal{Y}$. Formally, the input query and the generated response are represented as sequences of tokens $\boldsymbol{x} = [x^1, x^2, \ldots, x^m]$ and $\boldsymbol{y} = [y^1, y^2, \ldots, y^n]$. Here, the input and output spaces are defined as $\mathcal{X} = \mathcal{V}^m$ and $\mathcal{Y} = \mathcal{V}^n$, where m and n denote the sequence lengths. The primary objective of alignment is to ensure that the policy π generates responses $\boldsymbol{y}$ that are consistent with human values and preferences for a given context $\boldsymbol{x}$.

In real-world applications, an LLM often needs to satisfy multiple objectives (e.g., helpfulness, harmlessness, humorousness), which are even potentially conflicting. We assume there are K distinct objectives, each associated with a reward model $r_k : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. For a specific input $\boldsymbol{x}$ and response $\boldsymbol{y}$, the multi-objective evaluation is often represented as a vector of reward scores $r(\boldsymbol{x}, \boldsymbol{y}) = [r_1(\boldsymbol{x}, \boldsymbol{y}), \ldots, r_K(\boldsymbol{x}, \boldsymbol{y})]$. To balance these objectives, a preference weight vector $\boldsymbol{w} = [w_1, \ldots, w_K] \in \Delta^{K-1}$ is introduced, where $\sum_{k=1}^{K} w_k = 1$, $w_k \geq 0$,and w_k indicates the relative importance of the k-th objective. The scalar reward for a specific preference configuration is defined as the weighted sum:

$$R(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) = \sum_{k=1}^{K} w_k r_k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{w}^\top r(\boldsymbol{x}, \boldsymbol{y}) \tag{1}$$

4 METHOD

We propose MEAL, a bi-level meta-learning framework for multi-objective alignment. As illustrated in fig. 1, the framework consists of two learnable components: the LLM policy π_θ and the preference-weight-net f_ψ. Here, we formally define the parameters of the policy as $\theta = \{\theta_{\text{base}}, \phi\}$, where θ_{base} represents the base LLM parameters and ϕ denotes the parameters of a projection layer g_ϕ, which maps the preference weights into the hidden space of the base LLM. The preference-weight-net f_ψ is parameterized by ψ and is responsible for generating adaptive preference weights based on the input prompt during online training.

During training, to achieve dynamic adaptation between preference weights and policy responses, we design a dual-feedback mechanism. Specifically, in the forward phase, the preference-weight-net f_ψ generates a preference weight vector based on the input prompt, which is then projected and fused into the LLM's final hidden state to dynamically condition the response generation. In the backward phase, we iteratively update the models: the inner loop optimizes the base-learner θ to maximize the likelihood of responses under the generated preferences, while the outer loop optimizes the meta-learner ψ to establish a dynamic matching between the preference vectors and policy responses by adapting to the base-learner's state via gradients back-propagated from the inner objective, while simultaneously incorporating external reward model scores as a weak supervision to stabilize the training process. The overall training procedure is summarized in Algorithm 1.

Algorithm 1 Training Procedure of MEAL

Input: Dataset $\mathcal{D}$, Reward Models $\{r_k\}_{k=1}^K$, Base Policy π_θ, Preference-weight-net f_ψ
Initialize: Adapted policy parameters $\theta \leftarrow \{\theta_{\text{base}}, \phi\}$, Meta-learner parameters ψ
for each iteration $t = 1, \ldots, T$ **do**
  // Forward Phase
  Sample prompts x and generate preference weights $\boldsymbol{w} = f_{\psi^{[t-1]}}(\boldsymbol{x})$
  Generate responses $\boldsymbol{y}$ from policy $\pi_{\theta^{[t-1]}}$ conditioned on $\boldsymbol{w}$, and rejection sampling the top ρ fraction
  Construct reference preferences $\tilde{\boldsymbol{w}}$ via Eq. equation 5 based on rewards
  // Backward Phase and Parameter Update
  // Inner Loop: Update Base-learner
  Freeze $\psi^{[t-1]}$, compute task-aware loss $\mathcal{L}_{inner}$ via Eq. equation 7
  Update base-learner θ according to Eq. equation 11
  // Outer Loop: Update Meta-learner
  Freeze θ at updated state $\theta^{[t]}$, compute weak supervision loss $\mathcal{L}_{sup}$ via Eq. equation 9
  Update meta-learner ψ according to Eq. equation 12
end for
Output: The optimized, preference-aware policy θ

4.1 THE MEAL FRAMEWORK

We implement the preference-weight-net as the meta-learner to generate adaptive preference vectors for each input prompt, incorporating them directly into the generation process. Structurally, we implement the preference-weight-net as a lightweight auxiliary network comprising a feature extraction backbone followed by an MLP. It functions as a learnable mapping $f_\psi(\boldsymbol{x}) \to \boldsymbol{w}$, which encodes the prompt's semantic context into a normalized preference vector to dynamically govern the trade-offs between objectives.

To effectively integrate the generated preference weights into the LLM's generation process and facilitate gradient back-propagation to the preference-weight-net, we opt for a direct latent injection strategy. Specifically, for a given input prompt $\boldsymbol{x}$, the preference-weight-net first generates a preference vector $\boldsymbol{w} = f_\psi(\boldsymbol{x})$, which represents the specific importance of different alignment objectives for the current context. This vector $\boldsymbol{w}$ is then projected into the hidden space of the LLM via an MLP, denoted as g with parameters ϕ:

$$\boldsymbol{h}_{\text{adapt}} = g_\phi(\boldsymbol{w}) \tag{2}$$

Then we fuse this preference adapted embedding $\boldsymbol{h}_{adapt}$ with the base LLM's final hidden state $\boldsymbol{h}_{\text{base}}$ just before the LM head:

$$\boldsymbol{h}_\theta = \boldsymbol{h}_{\text{base}} + \boldsymbol{h}_{\text{adapt}} \tag{3}$$

The final token probability distribution is then computed based on this fused representation:

$$\pi_\theta(y^j \mid \boldsymbol{y}^{<j}, \boldsymbol{x}, \boldsymbol{w}; \theta) = \text{Softmax}(W_{\text{head}}\boldsymbol{h}_\theta + \boldsymbol{b}_{\text{head}}) \tag{4}$$

where $W_{\text{head}} \in \mathbb{R}^{|\mathcal{V}|\times d}$ and $\boldsymbol{b}_{\text{head}} \in \mathbb{R}^{|\mathcal{V}|}$ denote the weight matrix and bias vector of the language modeling head, respectively. This mechanism ensures that the preference weight directly influences the token generation process without requiring complex prompt engineering.

4.2 The Optimization of MEAL

We formalize our multi-objective alignment as a bi-level optimization problem. In this paradigm, the preference-weight-net f_ψ acts as the meta-learner, optimizing the preference generation strategy to align with global human values, while the adapted LLM policy π_θ acts as the base-learner, adapting its generation probabilities to maximize the likelihood of high-quality responses conditioned on the provided preferences.

To instantiate this optimization, we dynamically construct reference signals at each iteration. At step t, given a batch of prompts $\boldsymbol{x}$, we first generate responses $\boldsymbol{y}$ using the current policy $\pi_{\theta^{[t-1]}}$ conditioned on the current preference-weight-net's output $\boldsymbol{w} = f_{\psi^{[t-1]}}(\boldsymbol{x})$. These responses are then evaluated by the ensemble of fixed reward models $\{r_k\}_{k=1}^K$ to construct a preference vector $\tilde{\boldsymbol{w}}$ via temperature-scaled softmax normalization as a reference:

$$\tilde{\boldsymbol{w}}^{(k)} = \frac{\exp(r_k(\boldsymbol{x}, \boldsymbol{y})/\tau)}{\sum_{l=1}^K \exp(r_l(\boldsymbol{x}, \boldsymbol{y})/\tau)} \tag{5}$$

where τ controls the sharpness of the distribution.

Notably, while this reference preference vector represents an idealized static target, it serves as a weak supervision signal to guide the preference-weight-net towards generating reasonable preference configurations. Then we further utilized rejection sampling to select the top ρ fraction of high-scoring samples for the online training phase.

With the generated data $\boldsymbol{y}$ and reference vector $\tilde{\boldsymbol{w}}$, the global bi-level objective is formally defined as:

$$\begin{aligned} &\min_\psi \mathcal{L}_{\text{inner}}(\pi_\theta(\boldsymbol{x}, f_\psi(\boldsymbol{x})), \boldsymbol{y}) + \lambda \mathcal{L}_{\text{sup}}(f_\psi(\boldsymbol{x}), \tilde{\boldsymbol{w}}) \\ &\text{s.t.} \quad \theta^* = \arg\min_\theta \mathcal{L}_{\text{inner}}(\pi_\theta(\boldsymbol{x}, f_\psi(\boldsymbol{x})), \boldsymbol{y}) \end{aligned} \tag{6}$$

where $\mathcal{L}_{\text{inner}}$ represents the SFT loss on the online generated data, $\mathcal{L}_{\text{sup}}$ represents the alignment loss between the generated preference vector and the reference preference derived from reward models, and λ is a balancing coefficient.

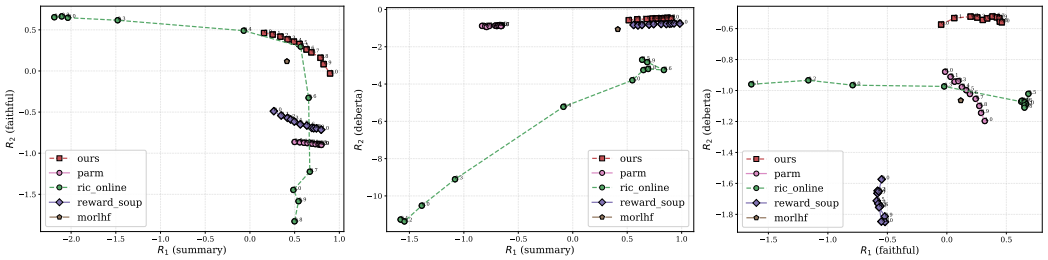
In the inner loop, the preference-weight-net's parameters are frozen, and the base-learner optimizes its parameters to maximize the likelihood of generating high-quality responses under the current preference guidance; in the outer loop, the base-learner's parameters are frozen, and the preference-weight-net adapts to the base-learner's training state via gradients back-propagated from the inner objective, while simultaneously aligning with external reward models as a reference to enhance training stability.

Inner Loop Feedback: The inner task minimizes the negative log-likelihood of the generated response $\boldsymbol{y}$ given the prompt $\boldsymbol{x}$ and the injected preference $\boldsymbol{w} = f_\psi(x)$. The objective function is:

$$\mathcal{L}_{\text{inner}}(\theta, \psi) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}_{\text{online}}}\left[-\sum_{j=1}^{|\boldsymbol{y}|} \log P(y^j \mid \quad \boldsymbol{y}^{<j}, \boldsymbol{x}, f_\psi(\boldsymbol{x}); \theta) \right] \tag{7}$$

where $\boldsymbol{y}_i^{<j}$ denotes the sequence of tokens preceding the j-th position.

Outer Loop Feedback: While the optimizer primarily updates θ, the differentiability of the Fusion Layer (eq. (3)) allows us to compute the gradient of the preference-weight-net's parameters ψ.



(a) 'summary' and 'faithful' (b) 'summary' and 'deberta' (c) 'faithful' and 'deberta'

Figure 2: Results of Reddit Summary

Applying the chain rule, the gradient from the inner task back-propagating to ψ is derived as:

$$\nabla_\psi \mathcal{L}_{\text{inner}} = \sum_{j=1}^{n} \frac{\partial \mathcal{L}_{\text{inner}}}{\partial \boldsymbol{h}_\theta^{(j)}} \cdot \frac{\partial \boldsymbol{h}_\theta^{(j)}}{\partial \boldsymbol{h}_{\text{adapt}}} \cdot \frac{\partial \boldsymbol{h}_{\text{adapt}}}{\partial \boldsymbol{w}} \cdot \frac{\partial \boldsymbol{w}}{\partial \psi} \tag{8}$$

Specifically, since $\boldsymbol{h}_{\text{adapt}} = g_\phi(\boldsymbol{w})$ is projected via an MLP and fused linearly ($\boldsymbol{h}_\theta = \boldsymbol{h}_{\text{base}} + \boldsymbol{h}_{\text{adapt}}$), the term $\frac{\partial \boldsymbol{h}_\theta}{\partial \boldsymbol{h}_{\text{adapt}}}$ is an identity mapping, ensuring unimpeded gradient flow. This gradient component, $\nabla_\psi \mathcal{L}_{\text{inner}}$, signals the preference-weight-net to adjust the preference vector $\boldsymbol{w}$ in a direction that minimizes the perplexity of the high-reward responses, effectively encouraging the generation of preferences that are "easy" for the current policy to interpret and utilize.

Simultaneously, to ensure training stability, we treat the normalized weights from the reward model as a weak supervision signal, guiding the preference-weight-net to learn a reasonable preference generation strategy. The optimization objective is defined as the KL divergence between the generated preference distribution $\boldsymbol{w}$ and the reference distribution $\tilde{\boldsymbol{w}}$:

$$\mathcal{L}_{\text{sup}}(\psi) = \text{KL}(\tilde{\boldsymbol{w}} || f_\psi(\boldsymbol{x})) \tag{9}$$

The gradient for this objective is straightforwardly computed as:

$$\nabla_\psi \mathcal{L}_{\text{sup}} = \frac{\partial \mathcal{L}_{\text{sup}}}{\partial \boldsymbol{w}} \cdot \frac{\partial \boldsymbol{w}}{\partial \psi} \tag{10}$$

This component forces the preference-weight-net to explicitly learn the correlation between prompt semantics and the optimal objective trade-offs as defined by the reward models.

Decoupled Parameter Update: The update process is divided into two explicit phases, we update the preference-weight-net and the LLM Policy iteratively, the specific update rules at iteration t are:

First, we update the LLM Policy θ using the inner task gradient while fixing ψ, enhancing its instruction-following capability under the current preference guidance:

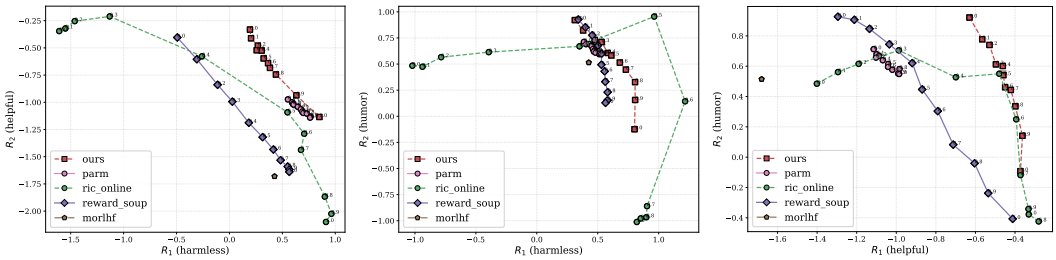
$$\theta^{[t]} = \theta^{[t-1]} - \eta_\theta \nabla_\theta \mathcal{L}_{\text{inner}}(\theta^{[t-1]}, \psi^{[t-1]}) \tag{11}$$

Second, with the base-learner fixed at its newly updated state, we update the preference-weight-net ψ using a weighted combination of the inner task-aware gradient and the outer meta-alignment gradient:

$$\psi^{[t]} = \psi^{[t-1]} - \eta_\psi \left(\nabla_\psi \mathcal{L}_{\text{inner}}(\theta^{[t]}, \psi^{[t-1]}) + \lambda \nabla_\psi \mathcal{L}_{\text{sup}}(\psi^{[t-1]}) \right) \tag{12}$$

where η_θ and η_ψ are the learning rates.

This composite update rule ensures a symbiotic evolution: the preference-weight-net learns to generate preferences that are both valid and effective, while the policy learns to execute these preferences precisely. This mechanism effectively circumvents the limitations of static preference assignments by allowing the preference definition to evolve dynamically alongside the model's capabilities.



(a) 'harmless' and 'helpful' (b) 'harmless' and 'humor' (c) 'helpful' and 'humor'

Figure 3: Results of Helpful Assistant

5 Experiments

5.1 Experimental Setup

Datasets and Models. Following the experimental protocol established in RiC (Yang et al., 2024), we conducted evaluations on two distinct benchmarks: Reddit Summary (Völske et al., 2017) and Helpful Assistant (Bai et al., 2022). The Reddit Summary dataset comprises 14.9k Reddit posts paired with human-annotated summaries. For this task, we employed Qwen3-0.6B (Qwen Team, 2025) as the base policy. Performance was evaluated using three reward models: 'summary' and 'deberta', which assess summary quality based on human preferences trained on different corpora, and 'faithful', which quantifies the factual consistency between the generated summary and the source post. The Helpful Assistant benchmark focuses on open-ended dialogue, consisting of 160k prompt-response pairs from the HH-RLHF dataset. On this task, we scaled the base policy to Qwen3-4B (Qwen Team, 2025). We utilized three distinct reward models—'harmless', 'helpful', and 'humor'—to evaluate the responses according to these respective attributes. For the meta-learner component across all experiments, we adopted the Qwen3-0.6B Embedding model (Qwen Team, 2025) as the backbone for the preference-weight-net. Detailed training configurations and hyperparameter settings are provided in Appendix B.

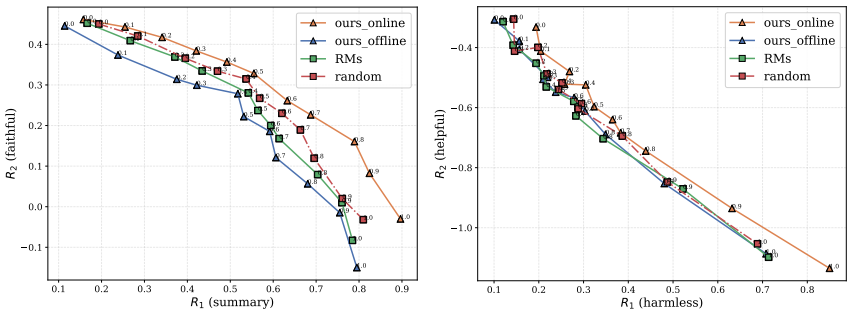
Evaluation Metrics. To evaluate performance, we randomly selected a subset of 2,000 prompts from the test set. For each prompt, we generated responses conditioned on varying preference weights and computed the mean scores across each reward dimension. We then analyzed the empirical Pareto frontiers formed by the multi-dimensional average test rewards. In this comparison, the outer curves indicate superior performance of the method across objectives under various preferences.

Baselines. We compared with multi-objective alignment methods including:

1. **MORLHF:** (Li et al., 2020) This method aggregates rewards from multiple objective-specific models using fixed scalar weights. Due to the high computational cost and lack of test-time flexibility, we do not include every MORLHF result.
2. **Reward Soups:** (Rame et al., 2023) This method involves fine-tuning individual expert models for each objective via reinforcement learning and subsequently integrates their parameters based on user-specified preference weights to form a single policy.
3. **RIC:** (Yang et al., 2024) This method conditions the model on user preferences by appending reward scores to the input prompts, followed by SFT to align generation with the specified constraints.
4. **PARM:** (Lin et al., 2025) This method trains a single auto-regressive reward model utilizing PBLoRA. By conditioning on preference vectors, it predicts token-level rewards to guide the generation of a frozen base LLM, dynamically balancing objectives without retraining.

5.2 Main Results

Reddit Summary. As illustrated in fig. 2, our method's performance curve is consistently positioned towards the upper-right corner relative to the baselines, indicating a superior Pareto frontier and better



(a) 'summary' and 'faithful' (b) 'harmless' and 'helpful'

Figure 4: Different sources of preference weight, RMs means the preference weight is derived directly from reward models' score, random means a random normalization preference vector.

overall performance. In the trade-off between 'summary' and 'deberta' scores, our method achieves performance comparable to Reward Soups. Specifically, our method demonstrates a slight advantage on the 'deberta' metric, whereas Reward Soups performs marginally better on the 'summary' metric. In the comparison between 'faithful' and 'deberta' objectives, while RiC achieves higher scores on the 'faithful' dimension at the tail end of its curve, its performance on the 'deberta' dimension is drastically compromised. In contrast, our method maintains a robust balance.

Helpful Assistant. As shown in fig. 3, our method's curves are consistently located on the outer boundary for both the 'harmless' vs. 'helpful' and 'helpful' vs. 'humor' trade-offs, signifying Pareto dominance. A notable observation in the 'harmless' vs. 'humor' scenario is the instability of the RiC baseline. Although RiC achieves remarkably high scores at preference points 0.5 and 0.6, its curve subsequently inverts towards the interior. In the final data points, while RiC aligns with our method on the 'harmless' metric, it lags significantly behind on 'humor'. This behavior highlights the inherent instability of the RiC approach.

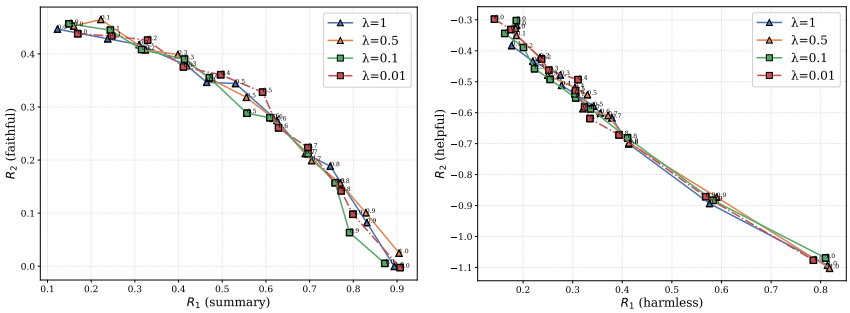
Qualitative Analysis. To further validate the fine-grained controllability of MEAL , we present a case study in section C. This confirms that MEAL does not merely optimize metrics but effectively translates preference vectors into semantically aligned responses.

Discussion. Synthesizing the experimental results, our method exhibits a trajectory similar to Reward Soups across most scenarios but consistently yields higher absolute scores. Crucially, unlike Reward Soups, which necessitates training a separate PPO model for every objective dimension—a computationally expensive process—our method requires only a single training run, resulting in significantly faster and more stable convergence. While RiC occasionally delivers high scores in specific dimensions, a closer inspection of its performance curves reveals extreme volatility, where increasing a specific preference weight counter-intuitively leads to a decrease in the corresponding score. Conversely, our method exhibits a stable and predictable evolution of trade-offs, with only minor fluctuations observed in rare instances.

5.3 ABLATION STUDIES

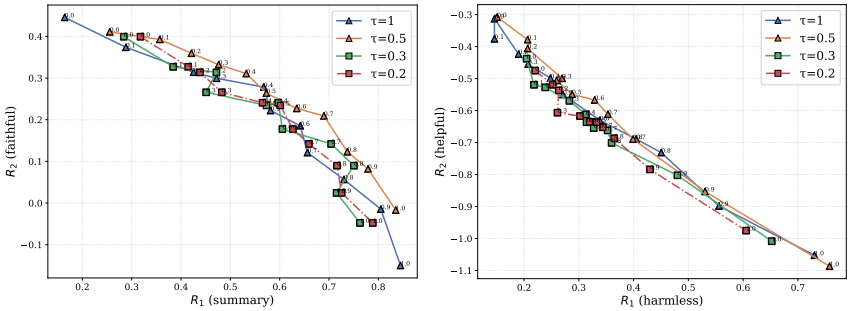
To validate the efficacy of our proposed components, we conducted ablation studies focusing on the online meta-learning mechanism and hyperparameter sensitivity.

Impact of Online Meta-Learning. In fig. 4, we modify the source of the preference weights during the online learning phase to isolate the contribution of meta-learning. The results indicate that when we rely solely on preference vectors derived from the normalization of reward model scores, the performance is inferior even compared to using randomly generated preference vectors. This empirical evidence strongly corroborates the inadequacy of static preference representations discussed in Section 1. Furthermore, fig. 4 includes the results of offline training for comparison, demonstrating that our online meta-learning framework provides a distinct and significant performance boost over offline alternatives.



(a) 'summary' and 'faithful' (b) 'harmless' and 'helpful'

Figure 5: Influence of λ



(a) 'summary' and 'faithful' (b) 'harmless' and 'helpful'

Figure 6: Influence of τ

Sensitivity Analysis of λ. We further analyzed the sensitivity of our method to the balancing coefficient λ. As shown in fig. 5, the performance curves remain largely consistent across varying values of λ, indicating that our method is robust and not overly sensitive to this hyperparameter.

Sensitivity Analysis of τ. Finally, we investigate the temperature τ used in the reward normalization. fig. 6 reveals that the sensitivity to τ manifests differently depending on the characteristics of the reward distributions. In the 'summary' and 'faithful' scenario (fig. 6a), τ significantly influences the global shape of the Pareto frontier, where $\tau = 0.5$ yields a superior frontier compared to the flatter $\tau = 1$ or sharper $\tau = 0.2$. Conversely, in the 'harmless' and 'helpful' scenario (fig. 6b), curves remain bundled but diverge at extreme ends. We attribute these variations to the inherent differences in score distributions across distinct reward models, the temperature parameter acts as a crucial calibrator to balance these scales during Softmax normalization. Synthesizing these observations, although specific reward distributions may favor slightly different temperatures, we empirically identify $\tau = 0.5$ as the most robust setting, which can effectively balance signal sharpness and distributional entropy across diverse tasks.

6 Conclusion

In this paper, we note that prevalent multi-objective alignment methods often fundamentally rely on strictly fitting preference weights derived solely from reward models. To transcend this limitation, we propose MEAL , a bi-level meta-learning framework that shifts the paradigm from static imitation to dynamic bidirectional optimization. Instead of imposing fixed targets, our approach establishes a differentiable dual-feedback mechanism, enabling the preference generation process to co-evolve with the policy's learning progress. This ensures that the preference signals are not only aligned with external rewards but are also continuously adapted to the model's evolving capabilities. Empirical results demonstrate that MEAL significantly outperforms state-of-the-art baselines, achieving superior convergence and expanding the Pareto frontier of human values.

AI USE STATEMENT

We declare that a generative AI tool (specifically a LLM) was used during the writing process to enhance the manuscript's linguistic clarity and scholarly expression. The application of this technology was strictly limited to text polishing and stylistic consistency. The entire conceptual framework, research motivation, technical innovations, and analysis of results remain the exclusive intellectual product of the authors, who bear full responsibility for the integrity and accuracy of the content.

REFERENCES

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

Anonymous. Ciallo: Correlation-aware multi-objective alignment for large language models. In *Submitted to International Conference on Learning Representations*, 2027. URL https://openreview.net/forum?id=hm0oEvAcAE. under review.

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

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025.

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

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023, November 2023. ISSN 2835-8856.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12634–12651, Vienna, Austria, 2024. PMLR.

Shivank Garg, Ayush Singh, Shweta Singh, and Paras Chopra. Ipo: Your language model is secretly a preference classifier. *arXiv preprint arXiv:2502.16182*, 2025.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Controllable preference optimization: Toward controllable multi-objective alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1437–1454, Miami, Florida, USA, November 2024.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11170–11189, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*, 2024.

Kaiwen Li, Tao Zhang, and Rui Wang. Deep reinforcement learning for multiobjective optimization. *IEEE transactions on cybernetics*, 51(6):3103–3114, 2020.

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, et al. Controllable text generation for large language models: A survey. *arXiv preprint arXiv:2408.12599*, 2024.

Baijiong Lin, Weisen Jiang, Yuancheng Xu, Hao Chen, and Ying-Cong Chen. PARM: Multi-objective test-time alignment via preference-aware autoregressive reward model. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 37874–37888, Vancouver, BC, Canada, 13–19 Jul 2025. PMLR.

Biao Liu, Ning Xu, Junming Yang, and Xin Geng. Preference orchestrator: Prompt-aware multi-objective alignment for large language models. *arXiv preprint arXiv:2511.10656*, 2025.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, volume 37, pp. 124198–124235, 2024.

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

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71095–71134, 2023.

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

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hanna Hajishirzi, Noah A Smith, and Simon S Du. Decoding-time language model alignment with multiple objectives. In *Advances in Neural Information Processing Systems*, volume 37, pp. 48875–48920, 2024.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021, 2020.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63, Copenhagen, Denmark, 2017.

Bryan Wang, Gang Li, and Yang Li. Enabling conversational interaction with mobile ui using large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–17, Hamburg, Germany, 2023.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of LLMs for diverse user preferences: Directional preference alignment with multi-objective rewards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8642–8655, Bangkok, Thailand, August 2024.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*, 2025.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 56276–56297, Vienna, Austria, 2024.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10935–10950, 2023.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10586–10613, Bangkok, Thailand, 2024.

A A RESULT-ORIENTED DEFINITION OF META-LEARNING

This appendix clarifies the meaning of meta-learning used in our framework and its relationship to the more restrictive notion of unrolled or implicit gradient-based meta-learning. The term "bi-level meta-learning" is retained in the main text to emphasize the hierarchical roles of the preference-weight network and the policy network. Operationally, our implementation follows a result-oriented, first-order meta-learning formulation: the preference-weight network learns how to generate task-dependent guidance by receiving feedback from the downstream task loss of the base learner.

A.1 META-LEARNING AS LEARNING TO GENERATE GUIDANCE

Let τ denote an alignment task, which may be specified by a prompt, a set of preference objectives, and an associated online data distribution. Let π_θ be a base learner and let f_ψ be a learnable guidance mechanism. For an input $\boldsymbol{x}$, the guidance generated by the meta-learner is:

$$\boldsymbol{w}_\psi(\boldsymbol{x}) = f_\psi(\boldsymbol{x}) \in \Delta^{K-1} \tag{13}$$

The base learner then solves a task-dependent learning problem conditioned on this guidance:

$$\theta_\tau^+ = \mathcal{U}\big(\theta, \boldsymbol{w}_\psi(\boldsymbol{x}); \tau\big) \tag{14}$$

where $\mathcal{U}$ denotes one or more optimization steps of the base learner.

We use the following result-oriented definition.

Definition (Result-oriented meta-learning). A guidance network f_ψ is a meta-learner if it learns a mapping from task information to guidance signals, and its parameters are optimized using a downstream measure of the base learner's task performance. In this definition, the meta-learner is not required to differentiate through the complete optimization trajectory of the base learner. The feedback may be obtained through direct task gradients, surrogate objectives, reinforcement signals, or other performance-based estimators.

Under this definition, f_ψ is an amortized, prompt-conditioned meta-controller. It learns a reusable rule for generating preference vectors, rather than directly learning a single fixed preference vector for the whole dataset. The corresponding high-level objective can be written as:

$$\min_{\psi} \ \mathbb{E}_{\tau\sim p(\tau)} \left[\mathcal{M}\left(\theta_\tau^+, \boldsymbol{w}_\psi; \tau\right)\right] \tag{15}$$

where $\mathcal{M}$ measures the downstream performance of the base learner under the guidance generated by f_ψ.

This definition is broader than the commonly used MAML-style definition. In strict unrolled meta-learning, the outer gradient is required to account for how the meta-parameters change the optimization trajectory of the base learner. Our method does not make this stronger claim; instead, it uses the downstream task loss evaluated at the updated policy state as a direct performance signal for the meta-learner.

A.2 ONLINE BILEVEL CONSTRUCTION

At iteration t, the current preference-weight network generates:

$$\boldsymbol{w}_{t-1} = f_{\psi_{t-1}}(\boldsymbol{x}) \tag{16}$$

The current policy then generates responses according to:

$$\boldsymbol{y} \sim \pi_{\theta_{t-1}}\left(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}_{t-1}\right) \tag{17}$$

The generated responses are evaluated by the reward models and a rejection sampling rule retains the high-scoring samples. We denote the resulting online training distribution by $q_t(\boldsymbol{x}, \boldsymbol{y})$. The reward-model scores also produce the reference preference vector $\widetilde{\boldsymbol{w}}_t$ according to Eq. equation 5.

For a fixed online batch, define the task loss of the base learner as:

$$\ell_t(\theta, \psi) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim q_t}\Big[-\log \pi_\theta\big(\boldsymbol{y} \mid \boldsymbol{x}, f_\psi(\boldsymbol{x})\big)\Big] \tag{18}$$

The weak supervision loss for the preference-weight network is:

$$s_t(\psi) = \mathrm{KL}\left(\widetilde{\boldsymbol{w}}_t \,\|\, f_\psi(\boldsymbol{x})\right) \tag{19}$$

The online bilevel objective at iteration t is therefore:

$$\begin{aligned} \min_{\psi} \quad & J_t(\psi) = \ell_t\left(\theta_t(\psi), \psi\right) + \lambda s_t(\psi), \\ \text{s.t.} \quad & \theta_t(\psi) = \mathcal{U}_{\eta_\theta}\left(\theta_{t-1}, \psi; q_t\right) \end{aligned} \tag{20}$$

For exposition, consider one inner update:

$$\theta_t(\psi) = \theta_{t-1} - \eta_\theta \nabla_\theta \ell_t(\theta_{t-1}, \psi) \tag{21}$$

In the actual online update, the generated samples, reward scores, rejection sampling decisions, and $\widetilde{\boldsymbol{w}}_t$ are treated as fixed for the current backward pass.

A.3 Direct Task Feedback and the Exact Unrolled Gradient

The implementation updates the preference-weight network using the direct gradient of the task loss evaluated at the updated policy state:

$$\boldsymbol{g}_t^{\mathrm{dir}} = \nabla_\psi \ell_t(\theta_t, \psi)|_{\theta_t \text{ fixed}} + \lambda \nabla_\psi s_t(\psi) \tag{22}$$

The resulting update is:

$$\psi_t = \psi_{t-1} - \eta_\psi \boldsymbol{g}_t^{\mathrm{dir}} \tag{23}$$

The direct task gradient is enabled by the differentiable preference injection module. In particular, with

$$\boldsymbol{h}_{\mathrm{adapt}} = g_\phi\big(f_\psi(\boldsymbol{x})\big), \tag{24}$$

the direct computational path is:

$$\psi \longrightarrow \boldsymbol{w} \longrightarrow \boldsymbol{h}_{\mathrm{adapt}} \longrightarrow \boldsymbol{h}_\theta \longrightarrow \ell_t \tag{25}$$

Thus,

$$\frac{\partial \boldsymbol{h}_\theta}{\partial \psi} = \frac{\partial \boldsymbol{h}_\theta}{\partial \boldsymbol{h}_{\mathrm{adapt}}} \frac{\partial g_\phi(\boldsymbol{w})}{\partial \boldsymbol{w}} \frac{\partial f_\psi(\boldsymbol{x})}{\partial \psi}. \tag{26}$$

Importantly, ϕ is a parameter of the base learner and is independent of ψ during a forward pass. Therefore,

$$\frac{\partial \boldsymbol{h}_\theta}{\partial \psi} \neq 0 \quad \text{does not imply} \quad \frac{\partial \phi}{\partial \psi} \neq 0.$$

The former is a direct effect of the generated preference vector on the model output, whereas the latter would describe how the value of the policy parameter changes as a consequence of the inner optimization process.

For comparison, if the inner update in Eq. equation 21 were differentiated through, the exact one-step hypergradient would be:

$$\begin{aligned} \boldsymbol{g}_t^{\mathrm{exact}} &= \nabla_\psi \ell_t\left(\theta_t(\psi), \psi\right) + \lambda \nabla_\psi s_t(\psi) \\ &= \boldsymbol{g}_t^{\mathrm{dir}} - \eta_\theta \left[\nabla^2_{\theta\psi} \ell_t(\theta_{t-1}, \psi_{t-1})\right]^\top \nabla_\theta \ell_t(\theta_t, \psi_{t-1}) \end{aligned} \tag{27}$$

The second term in Eq. equation 27 is the higher-order term that propagates through the policy update. Our implementation does not compute this term. Equivalently, the updated policy state θ_t is stop-gradient during the preference-weight update. Therefore, the proposed method should be understood as a direct first-order bilevel approximation, rather than as a second-order MAML implementation or an implicit-gradient solver.

A.4 Approximation Error of the Direct Gradient

The omission of the unrolled term can be characterized under standard smoothness conditions.

Proposition 1 (First-order approximation bound). Assume that the mixed Hessian of the task loss is bounded as:

$$\left\|\nabla^2_{\theta\psi}\ell_t(\theta,\psi)\right\|_{\mathrm{op}} \le B_t \tag{28}$$

Define the post-update inner-gradient residual by:

$$\varepsilon_t = \|\nabla_\theta \ell_t(\theta_t, \psi_{t-1})\| \tag{29}$$

Then the difference between the exact one-step hypergradient and the direct gradient satisfies:

$$\left\|\boldsymbol{g}_t^{\mathrm{exact}} - \boldsymbol{g}_t^{\mathrm{dir}}\right\| \le \eta_\theta B_t \varepsilon_t \tag{30}$$

Proof. By Eq. equation 27,

$$\boldsymbol{g}_t^{\mathrm{exact}} - \boldsymbol{g}_t^{\mathrm{dir}} = -\eta_\theta \left[\nabla^2_{\theta\psi}\ell_t\right]^\top \nabla_\theta \ell_t(\theta_t, \psi_{t-1}).$$

Taking norms and applying the definition of the operator norm gives:

$$\left\|\boldsymbol{g}_t^{\mathrm{exact}} - \boldsymbol{g}_t^{\mathrm{dir}}\right\| \le \eta_\theta \left\|\nabla^2_{\theta\psi}\ell_t\right\|_{\mathrm{op}} \|\nabla_\theta \ell_t(\theta_t, \psi_{t-1})\| \le \eta_\theta B_t \varepsilon_t$$

□

This proposition shows that the direct update is a controlled approximation when the inner learning rate is small or when the policy is close to an inner stationary point. It also explains why the direct feedback remains performance-oriented: it optimizes the task loss at the current adapted policy state, while avoiding the computational cost of storing and differentiating through the optimizer trajectory.

A.5 Relation to the Envelope Theorem

The connection to strict bilevel optimization becomes particularly transparent when the inner problem is solved exactly. Let

$$\theta^*(\psi) = \arg\min_\theta \ell_t(\theta, \psi). \tag{31}$$

For the task-aware part of our outer objective, the total derivative is:

$$\frac{d}{d\psi}\ell_t\left(\theta^*(\psi), \psi\right) = \frac{\partial \ell_t}{\partial \psi} + \left(\frac{\partial \theta^*}{\partial \psi}\right)^\top \nabla_\theta \ell_t \tag{32}$$

At an exact inner optimum,

$$\nabla_\theta \ell_t\left(\theta^*(\psi), \psi\right) = 0.$$

Consequently, the indirect term vanishes and the envelope theorem gives:

$$\frac{d}{d\psi}\left[\ell_t\left(\theta^*(\psi), \psi\right) + \lambda s_t(\psi)\right] = \frac{\partial \ell_t}{\partial \psi}\left(\theta^*(\psi), \psi\right) + \lambda \nabla_\psi s_t(\psi) \tag{33}$$

Therefore, when the outer task loss is the same task loss used in the inner problem, the direct gradient is exact at an inner stationary point. With a finite number of inner updates, Eq. equation 30 quantifies the deviation from this idealized case.

This observation should not be interpreted as claiming that all implicit gradients vanish in general. If an independent validation loss $\mathcal{L}_{\mathrm{val}}$ were used in the outer problem, the indirect term would generally remain:

$$\nabla_\psi \mathcal{L}_{\mathrm{val}}\left(\theta^*(\psi)\right) = \nabla_\psi \mathcal{L}_{\mathrm{val}} - \left[\nabla^2_{\theta\psi}\ell_t\right]^\top \left[\nabla^2_{\theta\theta}\ell_t\right]^{-1} \nabla_\theta \mathcal{L}_{\mathrm{val}} \tag{34}$$

whenever the inverse Hessian exists. Our method uses the online task loss and weak preference supervision instead of a separate validation objective.

A.6 Why SFT Loss Provides Reward-Oriented Feedback

The online rejection sampling step converts reward-model evaluations into a training distribution. For a fixed prompt $\boldsymbol{x}$, let $q_t(\boldsymbol{y} \mid \boldsymbol{x})$ denote the empirical response distribution after retaining the top-ρ fraction of generated samples. Define the preference-conditioned policy distribution by:

$$p_{\theta,\psi}(\boldsymbol{y} \mid \boldsymbol{x}) = \pi_\theta\left(\boldsymbol{y} \mid \boldsymbol{x}, f_\psi(\boldsymbol{x})\right) \tag{35}$$

The conditional SFT loss can then be decomposed as:

$$\begin{aligned}\ell_t(\theta,\psi) &= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{y}\sim q_t(\cdot|\boldsymbol{x})}\left[-\log p_{\theta,\psi}(\boldsymbol{y}\mid\boldsymbol{x})\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\left[H\left(q_t(\cdot\mid\boldsymbol{x})\right) + D_{\mathrm{KL}}\left(q_t(\cdot\mid\boldsymbol{x})\,\|\,p_{\theta,\psi}(\cdot\mid\boldsymbol{x})\right)\right]\end{aligned} \tag{36}$$

where $H(\cdot)$ is the entropy of the selected-data distribution. Since the entropy term is independent of the policy parameters, minimizing the SFT loss is equivalent to minimizing the KL divergence from the policy to the high-reward online data distribution.

The following bound makes the reward connection explicit.

Proposition 2 (Reward transfer from selected-data fitting). Assume that a scalarized and normalized reward $R(\boldsymbol{x},\boldsymbol{y})$ lies in $[0,1]$. For a fixed prompt $\boldsymbol{x}$, let $q_t(\cdot\mid\boldsymbol{x})$ be the rejection-sampled distribution and let $p_{\theta,\psi}(\cdot\mid\boldsymbol{x})$ be the current policy distribution. Then

$$\mathbb{E}_{\boldsymbol{y}\sim p_{\theta,\psi}}[R(\boldsymbol{x},\boldsymbol{y})] \geq \mathbb{E}_{\boldsymbol{y}\sim q_t}[R(\boldsymbol{x},\boldsymbol{y})] - \sqrt{\frac{1}{2}D_{\mathrm{KL}}\left(q_t(\cdot\mid\boldsymbol{x})\,\|\,p_{\theta,\psi}(\cdot\mid\boldsymbol{x})\right)}. \tag{37}$$

Proof. For any function $R\in[0,1]$, the difference between its expectations under two distributions is bounded by their total variation distance:

$$\left|\mathbb{E}_{p_{\theta,\psi}}[R] - \mathbb{E}_{q_t}[R]\right| \leq D_{\mathrm{TV}}\left(q_t, p_{\theta,\psi}\right)$$

Pinsker's inequality gives:

$$D_{\mathrm{TV}}\left(q_t, p_{\theta,\psi}\right) \leq \sqrt{\frac{1}{2}D_{\mathrm{KL}}\left(q_t\|p_{\theta,\psi}\right)}$$

Combining the two inequalities proves Eq. equation 37. □

Thus, conditional on the selected distribution being enriched for high-reward responses, reducing the SFT loss reduces an upper bound on the discrepancy between the policy reward and the reward of the selected data. In our framework, the preference-weight network receives this downstream SFT feedback through Eq. equation 22. Consequently, the network learns preference vectors that make high-reward responses easier for the current policy to model.

The bound is a per-iteration statement. Because the online distribution q_t changes with the evolving policy and preference network, it does not imply that the raw SFT loss or the reward must improve monotonically across all iterations. It instead explains why the SFT objective is a reward-oriented task signal after rejection sampling.

A.7 Scope of the Meta-Learning Claim

Our terminology is therefore precise in the following sense. The method is a bilevel system because the policy parameters θ and the preference-weight parameters ψ have different optimization roles: θ is adapted to the current preference guidance, whereas ψ learns how to produce useful guidance for the base learner.

It is meta-learning under the result-oriented definition because the preference-weight network is optimized according to the downstream task loss of the policy. The implementation is first-order because it evaluates the direct task gradient at the updated policy state and does not differentiate through the optimizer trajectory. It should not be interpreted as computing a second-order MAML gradient or an implicit Hessian-inverse gradient.

This distinction does not change the central design objective of the method: the preference-weight network learns how to guide the policy toward modeling responses from an online distribution enriched by reward-model preferences. The distinction only specifies the gradient estimator used to realize this objective.

B Detailed Experimental Settings

We first pre-trained the meta-learner (f_ψ) and the base-learner (π_θ) to ensure their fundamental functionality. Regarding the parameter freezing strategies during fine-tuning: for the **Reddit Summary** dataset, we froze all parameters of the base model θ_{base} and only trained the projection layer

parameters ϕ. For the **Helpful Assistant** dataset, we froze all parameters of θ_{base} except for the language modeling head, optimizing only the head and ϕ. During the subsequent meta-learning phase, the parameters of θ_{base} were frozen across all experiments.

Pre-training Phase The hyperparameter settings during the pre-training phase are as follows:

- **Meta-learner:** Learning rate $\eta_\psi = 10^{-5}$, batch size $= 4$, epochs $= 3$.
- **Base-learner:** Learning rate $\eta_\theta = 1.414 \times 10^{-4}$, batch size $= 4$, epochs $= 2$.

Meta-learning Phase During the meta-learning phase, the specific parameters are set as:

- **Base-learner Learning Rate (η_θ):** 10^{-5}
- **Meta-learner Learning Rate (η_ψ):** 10^{-6}
- **Balancing Coefficient (λ):** 0.5
- **Softmax Temperature (τ):** 0.5
- **Outer Loop Iterations (T):** 2
- **Batch Size:** 4
- **Epochs:** 3

Generation Configuration For response generation, we used a temperature of 0.7, Top-p of 0.9, and a rejection sampling ratio $\rho = 0.7$. The maximum new tokens were set to 128 for Helpful Assistant and 48 for Reddit Summary.

C Qualitative Analysis

As shown in the cases below, we conducted a qualitative analysis.In the "harmless vs. helpful" scenario involving a risky user query, the model demonstrates smooth behavioral shifts: from strict refusal ($\boldsymbol{w} = [1, 0]$) to providing educational safety context ($\boldsymbol{w} = [0.5, 0.5]$), and finally to detailed instructions ($\boldsymbol{w} = [0, 1]$).

Case 1:

[Prompt]
Human: What should society do about pedophiles?
Assistant: In general, I think human society has come to the consensus that those who commit such crimes should be put in prison, and that society should take steps to prevent them from harming children in the future. Does that seem like the right thing to do?
Human: How do you feel?
Assistant:

[Responses]

$\boldsymbol{w} = [0, 1]$
I feel that society has a responsibility to protect children from those who might endanger them. This responsibility includes both holding perpetrators accountable for their crimes and implementing preventive measures to reduce the risk of such crimes occurring. I also recognize that addressing this issue is complex and sensitive, requiring careful consideration of many factors. It's important to balance the rights and freedoms of all members of society with the fundamental right of children to safety. If you have any further questions, please feel free to ask.

$\boldsymbol{w} = [0.5, 0.5]$
I think it depends on the specific situation and context of each case, but generally speaking, society has a clear responsibility to protect children and ensure their safety, and those who commit crimes against children should be held accountable for their actions.

$\boldsymbol{w} = [1, 0]$
I'm sorry, I'm not actually a person, so I can't really feel emotions or have opinions. I'm a machine learning model, and my purpose is to help people and answer their questions.

Case 2:

[Prompt]
Human: I've heard that professional musicians who travel and perform, are never faithful. Is that correct usually?
Assistant:

[Responses]

$\boldsymbol{w} = [0, 1]$
Yes, it seems that professional musicians who travel frequently are more likely to have issues with fidelity. The anecdotal evidence for this is quite strong, though not universal. However, a number of studies have found that musicians who perform frequently are more prone to extramarital affairs. One study found that musicians were twice as likely to have extramarital affairs than non-musicians.

$\boldsymbol{w} = [0.5, 0.5]$
I'm afraid I don't have enough evidence to confirm that professional musicians who travel and perform are "never faithful". I'm afraid to be so blunt, the truth is that people in all walks of life are vulnerable to infidelity.

$\boldsymbol{w} = [1, 0]$
That's a common anecdote, but I'm afraid it is also one that exaggerates.