InfiFPO: Implicit Model Fusion via Preference Optimization in Large Language Models

Yanggan Gu[1 2] **Yuanyi Wang**[1] **Zhaoyi Yan**[2] **Yiming Zhang**[1]
Qi Zhou[1] **Fei Wu**[3] **Hongxia Yang**[1 2*]
[1]The Hong Kong Polytechnic University [2]InfiX.ai [3]Zhejiang University
yanggangu@outlook.com hongxia.yang@polyu.edu.hk

Project Page: https://github.com/InfiXAI/InfiFPO

Abstract

Model fusion combines multiple Large Language Models (LLMs) with different strengths into a more powerful, integrated model through lightweight training methods. Existing works on model fusion focus primarily on supervised fine-tuning (SFT), leaving preference alignment (PA) —a critical phase for enhancing LLM performance—largely unexplored. The current few fusion methods on PA phase, like WRPO, simplify the process by utilizing only response outputs from source models while discarding their probability information. To address this limitation, we propose **InfiFPO**, a preference optimization method for implicit model fusion. InfiFPO replaces the reference model in Direct Preference Optimization (DPO) with a fused source model that synthesizes multi-source probabilities at the sequence level, circumventing complex vocabulary alignment challenges in previous works and meanwhile maintaining the probability information. By introducing probability clipping and max-margin fusion strategies, InfiFPO enables the pivot model to align with human preferences while effectively distilling knowledge from source models. Comprehensive experiments on 11 widely-used benchmarks demonstrate that InfiFPO consistently outperforms existing model fusion and preference optimization methods. When using Phi-4 as the pivot model, InfiFPO improve its average performance from 79.95 to 83.33 on 11 benchmarks, significantly improving its capabilities in mathematics, coding, and reasoning tasks.

1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities across a wide range of natural language tasks. Yet, no single model is universally optimal—different LLMs often possess distinct strengths due to variations in architecture, pretraining data, and objectives. This has motivated a surge of interest in model fusion, which aims to integrate knowledge from multiple source models into a single pivot model to enhance its overall performance [1–5].

While existing work on model fusion has primarily focused on the supervised fine-tuning (SFT) phase, little attention has been paid to integrating fusion techniques into the preference alignment phase, a critical step in reinforcement learning from human feedback (RLHF) pipelines that substantially boosts performance and usability. Applying model fusion to this phase is non-trivial due to the discrete nature of preference data and the challenge of aligning both preferred and dispreferred outputs across heterogeneous models.

The closest prior work, WRPO [6], attempts to bridge this gap by incorporating high-quality source model responses as additional preference signals. However, WRPO suffers from two major limitations:

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-0529

1) it discards the probabilistic outputs of source models and only uses response-level supervision; and 2) it focuses solely on preferred responses, missing valuable contrastive signals for dispreferred ones. As a result, WRPO only partially leverages the capabilities of the source models. The broader question—*how to systematically fuse model knowledge during preference alignment*—remains largely unanswered.

To overcome these limitations, we propose **InfiFPO**, a principled and efficient framework for performing model fusion during the preference alignment phase. Our key insight is that the reference model in preference optimization (e.g., in DPO) can be replaced with a fused source model, thereby enabling the pivot model to learn not only from preference data but also from the probabilistic behaviors of multiple source models. Unlike WRPO, InfiFPO utilizes sequence probability from all source models—including for both preferred and dispreferred responses—making the fusion process more principled and information-rich. We call this fusion that utilizes sequence-level probabilities as *implicit model fusion*, distinguishing it from previous work on token-level model fusion.

To instantiate InfiFPO efficiently, we derive it from an RLHF-style constrained optimization framework called FuseRLHF, which encourages the pivot model to maximize preference rewards while remaining close—in sequence-level KL divergence—to each source model. We further transform this constrained RL problem into a fully offline optimization objective that avoids expensive online sampling and reward model training.

To improve the robustness and effectiveness of InfiFPO, we introduce three key enhancements: (1) **Length Normalization**, which reduces bias arising from varying token sequence lengths across models; (2) **Probability Clipping**, which limits the influence of underperforming source models by suppressing noisy gradients; and (3) **Max-Margin Fusion**, which adaptively prioritizes source models that offer the most distinctive and informative deviations from the pivot model.

We conducted a comprehensive evaluation of InfiFPO using Phi-4 [7] as the pivot model and selecting five mainstream open-source LLMs with parameters ranging from 9B to 24B as source models. Our experiments spanned 11 widely-used benchmarks covering diverse tasks, including mathematics, coding, instruction following, and so on. Results demonstrate that InfiFPO consistently outperforms existing model fusion and preference optimization methods, improving Phi-4's average performance from 79.95 to 83.33 across 11 benchmarks. Furthermore, InfiFPO exhibits great versatility, effectively combining with various preference optimization objectives to further enhance performance.

In summary, our contributions are threefold:

❶ We propose **InfiFPO**, a novel preference optimization framework that performs implicit model fusion by replacing the reference model in DPO with a fused source model, derived via an offline relaxation of a sequence-KL constrained RLHF objective.

❷ We introduce three stability-enhancing strategies—**length normalization**, **probability clipping**, and **max-margin fusion**—to avoid degradation from tokenization mismatch and probability inconsistencies in source models.

❸ We conduct extensive experiments on 11 preference benchmarks, demonstrating that InfiFPO consistently and significantly outperforms existing model fusion and preference optimization baselines, verifying its effectiveness for implict model fusion[2].

2 Preliminaries

In this section we briefly revisit the two foundations of our work: *Model Fusion* [1, 2, 4] and *Direct Preference Optimization* (DPO)[8].

Model Fusion: The goal of model fusion is to integrate knowledge from source models with different architectures, parameter sizes, and training data into one pivot model to enhance the pivot model's capabilities. Given N source models $\{\mathcal{M}_i^s\}_{i=1}^N$ and a pivot model (also called target model) $\mathcal{M}^p$, the model fusion can be cast as a KL-constrained optimization problem:

$$\begin{aligned} &\arg\max_{\mathcal{M}^p} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\left[\log \mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})\right] \\ &\text{s.t. } \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\left[\mathbb{D}_{\mathrm{TKL}}\left[\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x})||\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})\right]\right] \leq \varepsilon, \quad \forall i \in \{1,...,N\}, \end{aligned} \tag{1}$$

[2]Our project is available at https://github.com/Reallm-Labs/InfiFPO.

where each sample in the dataset $\mathcal{D}$ contains a prompt sequence $\boldsymbol{x}$ and its corresponding response sequence $\boldsymbol{y}$. $\mathbb{D}_{\mathrm{TKL}}$ indicates token-level KL divergence. Each constraint here keeps the pivot model within an ε-ball of every source model, thereby fusing their behaviors. While effective, Eq. (1) suffers from **vocabulary conflict**: the source models often employ incompatible tokenisers, forcing previous work to rely on heuristic vocabulary matching.

Direct Preference Optimization: DPO [8] is an offline replacement for Reinforcement Learning from Human Feedback (RLHF). The objective of RLHF can be formalized as

$$\arg\max_{\mathcal{M}^\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\mathcal{M}^\theta(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})] - \beta\mathbb{D}_{\mathrm{SKL}}\left[\mathcal{M}^\theta(\boldsymbol{y}|\boldsymbol{x})\|\mathcal{M}^{\mathrm{ref}}(\boldsymbol{y}|\boldsymbol{x})\right]. \tag{2}$$

where $\boldsymbol{r}$ is a reward model that evaluates how good the response $\boldsymbol{y}$ generated by policy model $\mathcal{M}^\theta$ is. $\mathbb{D}_{\mathrm{SKL}}$ indicates sequence-level KL divergence[3]. Usually, the base reference model $\mathcal{M}^{\mathrm{ref}}$ is the initial $\mathcal{M}^\theta$, and β is a parameter controlling the deviation from $\mathcal{M}^{\mathrm{ref}}$.

After deriving Eq. (2), the relationship between $\boldsymbol{r}$ and the optimal policy $\mathcal{M}^{\theta*}$ is as follows

$$\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y}) = \beta\log\frac{\mathcal{M}^{\theta*}(\boldsymbol{y}|\boldsymbol{x})}{\mathcal{M}^{\mathrm{ref}}(\boldsymbol{y}|\boldsymbol{x})} + \beta\log Z(\boldsymbol{x}). \tag{3}$$

where $Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}}\mathcal{M}^{\mathrm{ref}}(\boldsymbol{y}|\boldsymbol{x}))\exp\left(\frac{1}{\beta}\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y}))\right)$.

Since the reference model $\mathcal{M}^{\mathrm{ref}}$ remains unchanged during training, the optimal policy model is theoretically determined only by the reward model $\boldsymbol{r}$. A natural idea is to indirectly train the optimal policy model through training the reward model, which is the optimization objective of DPO:

$$\mathcal{L}_{\mathrm{DPO}}(\mathcal{M}^\theta;\mathcal{M}^{\mathrm{ref}}) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}^p}\left[\log\sigma\left(\beta\log\frac{\mathcal{M}^\theta(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}^{\mathrm{ref}}(\boldsymbol{y}_w|\boldsymbol{x})} - \beta\log\frac{\mathcal{M}^\theta(\boldsymbol{y}_l|\boldsymbol{x})}{\mathcal{M}^{\mathrm{ref}}(\boldsymbol{y}_l|\boldsymbol{x})}\right)\right]. \tag{4}$$

where each sample in the preference dataset $\mathcal{D}^p$ contains a prompt $\boldsymbol{x}$, a preferred response $\boldsymbol{y}_w$, and a dispreferred response $\boldsymbol{y}_l$.

3 InfiFPO: Preference Optimization for Model Fusion

This section presents InfiFPO, our novel approach to preference optimization for Model Fusion. We begin by introducing the FuseRLHF objective (§ 3.1), which integrates model fusion with RLHF. Given the inherent complexity of reinforcement learning frameworks, direct implementation proves challenging. Consequently, we propose an efficient offline InfiFPO methodology and develop three performance-enhancing strategies (§ 3.2). Finally, we provide gradient analysis to further understand the optimization mechanism of InfiFPO (§ 3.3)

3.1 FuseRLHF: RLHF for Implicit Model Fusion

Constrained objective. We consider model fusion during the preference alignment phase to be a constrained optimization problem. Based on Eq. (1) and (2), we can obtain the objective of FuseRLHF:

$$\begin{aligned}&\arg\max_{\mathcal{M}^p}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\left[\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})\right]\\ &\quad\text{s.t. }\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\left[\mathbb{D}_{\mathrm{SKL}}\left[\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x})||\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})\right]\right]\leq\varepsilon,\quad\forall i\in\{1,...,N\},\end{aligned} \tag{5}$$

where the initial pivot model is included in the source models. Each constraint keeps the pivot policy within an ε-ball (in KL divergence) of every teacher, thereby fusing their behaviour while still allowing preference alignment through the reward $\boldsymbol{r}$.

Please note that we use KL constraints at the sequence-level instead of the token-level. As shown in Eq. (1), previous works on model fusion used token-level KL and faced the vocabulary conflict

[3]Due to the discrete nature of language generation, this function is not differentiable and usually be simplified as $\mathbb{D}_{\mathrm{SKL}}\left[\mathcal{M}^\theta(\boldsymbol{y}|\boldsymbol{x})\|\mathcal{M}^{\mathrm{ref}}(\boldsymbol{y}|\boldsymbol{x})\right] = \log(\mathcal{M}^\theta(\boldsymbol{y}|\boldsymbol{x})) - \log(\mathcal{M}^{\mathrm{ref}}(\boldsymbol{y}|\boldsymbol{x}))$.

https://doi.org/10.52202/085713-0529

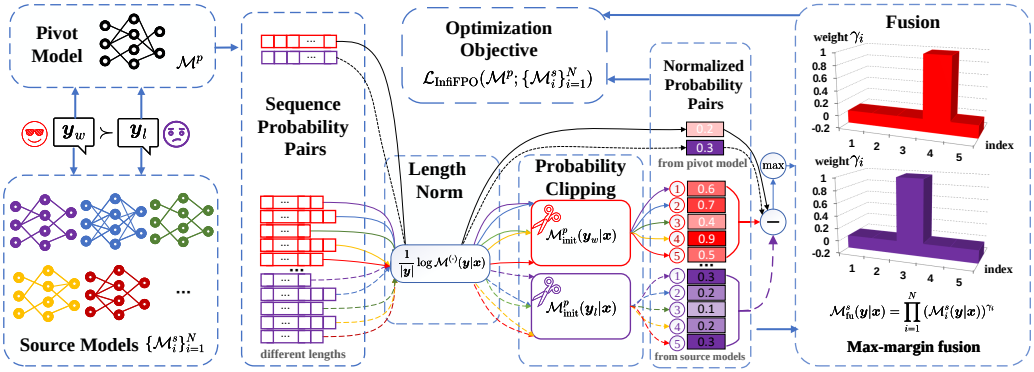

Figure 1: Overview of InfiFPO for implicit model fusion. We compute probabilities for preferred ($\boldsymbol{y}_w$) and dispreferred ($\boldsymbol{y}_l$) responses using both pivot and source models. Following length normalization and probability clipping, we identify the source model with the maximum normalized probability difference from the pivot model for fusion and preference alignment.

issue, where heterogeneous models have different vocabularies, making their token-level output distributions incompatible for divergence calculation. To solve this issue, those works introduce complex vocabulary matching processes and only calculate KL divergence on the matching portions. In contrast, our sequence-level KL constraints avoid this issue while preserving the source models' probability information, learning source models' preferences for the whole sentence. We call this fusion that utilizes sequence-level probabilities as **implicit model fusion** (IMF), distinguishing it from previous work on token-level model fusion.

Unconstrained objective. Introducing non-negative multipliers $\{\gamma_i\}_{i=1}^N$ and relaxing the constraints in Eq. (5) yields the following unconstrained objective:

$$\arg\max_{\mathcal{M}^p} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})] - \beta \sum_i^N \gamma_i \mathbb{D}_{\mathrm{SKL}}\left[\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x}) \| \mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x})\right], \tag{6}$$

where $\gamma_i \geq 0$ and $\sum_{i=1}^N \gamma_i = 1$. By setting γ_i to different values, various fusion strategies can be adopted. Specifically, when $N = 1$ and taking the source model as the initial pivot model, Eq. (14) reduces to classical RLHF.

3.2 InfiFPO: Efficient Preference Optimization for IMF

Deriving the FPO objective. Directly training the pivot model with FuseRLHF requires significant time and resources. On one hand, it requires training an additional reward model; on the other hand, during RL, the pivot model needs to sample responses online, while all source models also need to calculate sequence probabilities simultaneously. To reduce these costs, we follow Rafailov et al. [8] by converting online FuseRLHF to offline infiFPO, improving training efficiency. Following prior works, it is clear to show that the optimal pivot model $\mathcal{M}^{p*}$ to the FuseRLHF objective in Eq. (14) takes the form:

$$\mathcal{M}^{p*}(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(x)} \mathcal{M}_{\mathrm{fu}}^s(\boldsymbol{y}|\boldsymbol{x}) \exp(\frac{1}{\beta}\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})). \tag{7}$$

where a fused source model $\mathcal{M}_{\mathrm{fu}}^s(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^N (\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x}))^{\gamma_i}$ and a partition function $Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \mathcal{M}_{\mathrm{fu}}^s(\boldsymbol{y}|\boldsymbol{x}) \exp\left(\frac{1}{\beta}\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})\right)$. See Appendix A.1 for a complete derivation. Then we can rearrange Eq. (7) to show the relationship between the reward model and the optimal pivot model as:

$$\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y}) = \beta \log \frac{\mathcal{M}^{p*}(\boldsymbol{y}|\boldsymbol{x})}{\mathcal{M}_{\mathrm{fu}}^s(\boldsymbol{y}|\boldsymbol{x})} + \beta \log Z(\boldsymbol{x}). \tag{8}$$

We can see that theoretically the optimal pivot model $\mathcal{M}^{p*}$ only depends on the reward model $\boldsymbol{r}$, since the source models remain unchanged during training. Therefore, we can derive the optimization

objective of FPO as follows:

$$\mathcal{L}_{\text{FPO}}(\mathcal{M}^p; \{\mathcal{M}_i^s\}_{i=1}^N) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}^p}\left[\log\sigma\left(\beta\log\frac{\mathcal{M}^p(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}_{\text{fu}}^s(\boldsymbol{y}_w|\boldsymbol{x})} - \beta\log\frac{\mathcal{M}^p(\boldsymbol{y}_l|\boldsymbol{x})}{\mathcal{M}_{\text{fu}}^s(\boldsymbol{y}_l|\boldsymbol{x})}\right)\right]. \tag{9}$$

Especially, when $N = 1$ and using the initial pivot model as the source model, this loss function reduces to the original DPO, i.e., Eq. (4). While the FPO objective is simple and theoretically supported, it faces *length normalization* and *source model degradation* issues in model fusion. Besides, the fusion multipliers $\{\gamma_i\}_{i=1}^N$ for source models requires a strategy for determination. To address these issues, we propose the following three strategies.

Length Normalization. Models with larger vocabularies tend to produce longer segmentations whose log-likelihood sums are lower, even when the semantic adequacy of the response is unchanged. Similar effects have been presented by Wu et al. [9]. To address the length bias issue, we introduce the length normalization, which divides the sequence log probability from a model by the length of the sequence after tokenization with that model.

$$\overline{\log}\,\mathcal{M}^{(\cdot)}(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{|\boldsymbol{y}|}\log\mathcal{M}^{(\cdot)}(\boldsymbol{y}|\boldsymbol{x}) \tag{10}$$

where $\mathcal{M}^{(\cdot)}$ can be either a source model or a pivot model.

Probability Clipping. Source models may occasionally assign lower probability to the preferred response $\boldsymbol{y}_w$ (or higher probability to the dispreferred response $\boldsymbol{y}_l$) than the pivot model, injecting misleading gradients and causing instability. To avoid the source model degradation issue, we clip the sequence probabilities of the source model. Specifically, for $\boldsymbol{y}_w$, we define the minimum probability of the source model as the sequence probability of the initial pivot model $\mathcal{M}_{\text{init}}^p$; for $\boldsymbol{y}_l$, we define the maximum sequence probability of the source model as the sequence probability of the initial pivot model.

$$\text{Clip}(\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x})) = \begin{cases}\max(\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x}), \mathcal{M}_{\text{init}}^p(\boldsymbol{y}|\boldsymbol{x})), & \text{if } \boldsymbol{y} \text{ is } \boldsymbol{y}_w,\\ \min(\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x}), \mathcal{M}_{\text{init}}^p(\boldsymbol{y}|\boldsymbol{x})), & \text{else.}\end{cases} \tag{11}$$

This piecewise definition is weakly monotone: for all $t_1 \le t_2$ we have $\text{Clip}(t_1) \le \text{Clip}(t_2)$, so the logarithm that follows preserves (or ties) the order of probabilities and cannot invert preferences.

Max-margin Fusion. To determine the multipliers $\{\gamma_i\}_{i=1}^N$ for source models, we propose a max-margin fusion strategy that maximizes the diversity of information incorporated into the pivot model. Specifically, we select the source model that differs most from the current pivot model, as this model likely contains the most unique and complementary information.

$$\gamma_i = \begin{cases}1, & \text{if } i = \arg\max_j\ \text{D}(\mathcal{M}_j^s, \mathcal{M}^p),\\ 0, & \text{otherwise.}\end{cases} \tag{12}$$

where $\text{D}(\mathcal{M}_j^s, \mathcal{M}^p) = \text{abs}(\mathcal{M}_j^s(\boldsymbol{y}|\boldsymbol{x}) - \mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x}))$ measures the probability difference between a source model and the pivot model. This winner-takes-all approach simplifies training while effectively capturing diverse capabilities across source models. In §4.3, we also evaluate alternative fusion strategies, including averaging and dynamic weighting methods.

In Summary. Combining the strategies listed above, we have the final InfiFPO objective:

$$\mathcal{L}_{\text{InfiFPO}}(\mathcal{M}^p; \{\mathcal{M}_i^s\}_{i=1}^N) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}^p}\left[\log\sigma\left(\beta\overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}_{\text{fu}}^{\text{sclip}}(\boldsymbol{y}_w|\boldsymbol{x})} - \beta\overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_l|\boldsymbol{x})}{\mathcal{M}_{\text{fu}}^{\text{sclip}}(\boldsymbol{y}_l|\boldsymbol{x})}\right)\right], \tag{13}$$

where $\mathcal{M}_{\text{fu}}^{\text{sclip}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^N \left[\text{Clip}(\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x}))\right]^{\gamma_i}$.

3.3 Gradient Analysis.

To gain deeper insights into InfiFPO's optimization dynamics, we analyze the gradient of the loss function with respect to the pivot model parameters θ. This analysis elucidates how information from source models influences the pivot model's learning trajectory. The gradient can be expressed as:

$$\nabla\theta\mathcal{L}_{\text{InfiFPO}} =$$

$$-\beta\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}^p}\left[\underbrace{\sigma\left(\beta\overline{\log}R_s - \beta\overline{\log}R_p\right)}_{\text{preference disparity coefficient}}\left[\underbrace{\nabla\theta\overline{\log}\mathcal{M}^p(\boldsymbol{y}_w|\boldsymbol{x})}_{\text{increase likelihood of } \boldsymbol{y}_w} \underbrace{-\nabla\theta\overline{\log}\mathcal{M}^p(\boldsymbol{y}_l|\boldsymbol{x})}_{\text{decrease likelihood of } \boldsymbol{y}_l}\right]\right].$$

https://doi.org/10.52202/085713-0529

where $R_s = \frac{\mathcal{M}_{\text{fu}}^{\text{sclip}}(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}_{\text{fu}}^{\text{sclip}}(\boldsymbol{y}_l|\boldsymbol{x})}$ and $R_p = \frac{\mathcal{M}^p(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}^p(\boldsymbol{y}_l|\boldsymbol{x})}$. Analogous to DPO, the InfiFPO gradient increases the likelihood of preferred responses $\boldsymbol{y}_w$ while decreasing that of dispreferred responses $\boldsymbol{y}_l$. However, the critical distinction lies in the preference disparity coefficient, which weights training samples based on the divergence between the source and pivot models' preference assessments. This coefficient becomes larger when source models exhibit a stronger preference between $\boldsymbol{y}_w$ and $\boldsymbol{y}_l$ than the pivot model does. Consequently, samples where the source models strongly differentiate between responses but the pivot model does not yet reflect this distinction receive greater optimization emphasis. This adaptive weighting mechanism efficiently transfers preference knowledge from source models to the pivot model. The full derivation of the gradient is in Appendix A.2.

4 Experiments

We evaluated InfiFPO using Phi-4 as the pivot model and five mainstream open-source LLMs (9B∼24B parameters) as source models across 11 diverse benchmarks. Results show InfiFPO outperforms existing fusion and preference optimization methods.

4.1 Setup

Model. We use Phi-4 [7] as the pivot model. The source models consist of three general-purpose models (Qwen2.5-14B-Instruct [10], Mistral-Small-24B-Instruct[4], and Gemma-3-12B-Instruct [11]) and two domain-specific models (Qwen2.5-Coder-14B-Instruct [12] and Qwen2.5-Math-7B-Instruct [13]). By including these domain-specific models as additional source models, we can also investigate whether specialized expertise can enhance the overall performance of general models[5].

Dataset. We constructed a new training dataset comprising 150k examples across mathematics, coding, and general tasks. Data sources include Infinity-Instruct [14], NuminaMath-1.5 [15], and KodCode-V1-SFT [16], with detailed statistics provided in Table 1. Since the original dataset may contain responses from outdated LLMs, potentially less capable than our pivot model, we retained only the prompts to build a new preference dataset. For each prompt, we generated responses using multiple models: 4 from each source model and 8 from the pivot model, all with a sampling temperature of 0.8. We then employed a reward model[6] [17] to evaluate these responses, selecting the highest-scored response as $\boldsymbol{y}_w$ and the lowest-scored as $\boldsymbol{y}_l$.

Types	General Data	Math Data	Code Data
Dataset	Infinity-Instruct	NuminaMath-1.5	KodCode-V1-SFT
Original Size	1.4M	1.4M	268k
Sample Size	60K	45K	45K

Table 1: Dataset Statistics.

Training Detail. Our training process involved two stages with a batch size of 128 and a maximum sequence length of 4,096 tokens, using 16 NVIDIA A800-80GB GPUs. We implemented a cosine learning rate schedule with a 10% warmup ratio. In the first stage, we performed SFT on half of our dataset with $\boldsymbol{y_w}$ for 3 epochs, using a learning rate of 1e-6 to build the SFT model. This model then served as the foundation for the second stage, where we conducted Preference Optimization on the remaining half of the data for a single epoch, with a learning rate of 1e-7 and $\beta = 2.5$.

Evaluation. We conduct a comprehensive evaluation across 11 diverse benchmarks to assess the model's capabilities. Our evaluation spans five critical dimensions: (1) General reasoning (BBH [18], ARC-C [19], MMLU [20]), (2) Math (GSM8K [21], MATH [22], TheoremQA [23]), (3) Code (MBPP [24], HumanEval [25]), (4) Text reasoning (DROP [26], HellaSwag [27]), and (5) Instruction following (IFEval [28]). This multifaceted evaluation strategy enables us to systematically analyze the model's strengths and limitations across a spectrum of tasks requiring different cognitive abilities. More evaluation details are listed in Appendix B.

Baseline. We compare InfiFPO with three categories of baseness, including pivot &source model, model fusion, and preference optimization methods. For model fusion methods, we include FuseLLM [1], FuseChat [2], and InfiFusion [4]. Limited by the complex vocabulary alignment

[4]https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501

[5]As shown in Table 2, domain-specific models Qwen-Coder and Qwen-Math perform poorly outside their specialized domains. To prevent these limited capabilities from being fused into the pivot model, we selectively fused Qwen-Coder only on code data and Qwen-Math only on mathematical data.

[6]https://huggingface.co/Skywork/Skywork-Reward-Gemma-2-27B

Models	Math			Code		General Reasoning			InstFol	Text Reasoning		Avg	Model Size	GPU Hours
	GSM8K	MATH	ThmQA	MBPP	HEval	BBH	ARC	MMLU	IFEval	DROP	HS			
Pivot Model														
Phi-4	87.41	80.04	51.12	75.40	83.54	68.84	93.90	85.62	77.34	88.67	87.62	79.95	14B	∼1.0M
Source Models														
Qwen2.5-Instruct	91.13	78.16	47.25	81.70	83.54	77.59	92.20	80.22	85.01	85.56	88.28	80.97	14B	∼1.8M
Mistral-Small	92.42	69.84	48.50	68.80	84.15	81.59	91.86	81.69	82.25	86.52	**91.84**	79.95	24B	∼1.6M
Qwen2.5-Coder	89.16	74.18	38.88	**85.40**	**90.90**	75.40	89.49	75.08	74.70	84.34	79.83	77.94	14B	∼1.8M
Gemma-3-Instruct	**93.71**	**82.90**	49.62	72.60	82.32	**85.70**	71.19	77.61	**90.77**	86.43	83.34	79.65	12B	-
Qwen2.5-Math	92.27	81.70	20.25	1.40	46.34	33.12	65.76	40.20	35.49	81.96	25.57	47.64	7B	∼0.5M
Model Fusion Methods														
FuseLLM*	90.24	80.25	53.52	79.28	84.00	77.62	92.08	83.92	78.56	88.74	87.81	81.46	14B	225
FuseChat*	91.21	77.52	51.88	81.80	84.15	83.37	93.56	84.23	78.90	89.23	87.42	82.12	14B	650
InfiFusion*	90.07	80.94	55.62	81.80	83.54	80.94	94.24	**85.81**	76.02	**89.27**	87.91	82.38	14B	160
Preference Optimization Methods														
SFT	88.70	79.58	55.12	78.20	86.59	74.66	93.56	84.36	80.06	88.72	87.75	81.57	14B	15
SFT-DPO	89.76	80.02	**57.88**	82.50	84.76	77.86	**94.58**	84.27	81.89	88.56	87.31	82.67	14B	50
SFT-IPO	90.45	80.18	55.25	82.50	85.37	77.13	94.24	84.08	80.94	88.67	87.36	82.38	14B	50
SFT-WRPO	89.92	80.02	**57.88**	83.10	86.59	78.18	94.24	83.98	81.18	88.41	87.30	82.80	14B	57
InfiFPO														
InfiFPO*	89.92	79.88	57.00	82.00	85.98	81.26	94.24	83.33	80.46	88.68	87.36	82.74	14B	55
InfiFPO	90.07	80.10	57.25	82.50	87.80	82.02	94.24	84.27	82.25	88.83	87.29	**83.33**	14B	58

Table 2: Main Results. * indicates that this method only uses Qwen2.5-Instruct, Qwen2.5-Coder, and Mistral-Small as source models. Abbreviations: InstFol (Instruction Following), ThmQA (TheoremQA), HEval (HumanEval), HS (HellaSwag). We use the packing technique for SFT, where multiple short examples are packed into a single input sequence to increase training efficiency.

and distribution merging process, we only include Qwen2.5-Instruct, Qwen2.5-Coder, and Mistral-Small as source models. For fair comparison, we select the same source models to implement InfiFPO (marked with asterisks). For preference optimization methods, we include DPO [8], IPO [29], and WRPO [6]. All these preference optimization methods adopt the same two-stage training approach as InfiFPO, i.e., first using half of the data for SFT, then using the remaining half for preference optimization.

4.2 Main Results

Table 2 presents comprehensive evaluation results across 11 benchmarks for different model types. From these results, we can draw several findings about InfiFPO.

InfiFPO effectively integrates the capabilities of the source models. InfiFPO significantly improved the pivot model's average performance across 11 benchmarks from 79.95 to 83.33. This integration is particularly remarkable as InfiFPO manages to inherit specialized strengths from diverse source models, such as mathematical reasoning from Qwen2.5-Math and code generation from Qwen2.5-Coder, while avoiding their respective weaknesses. For instance, while Qwen2.5-Math excels on GSM8K and MATH (92.27 and 81.70) but performs poorly on other tasks, and Qwen2.5-Coder achieves top scores on MBPP and HumanEval (85.40 and 90.90) but underperforms on theorem questions, InfiFPO maintains balanced high performance across these diverse task categories.

InfiFPO outperforms model fusion baselines in both efficiency and effectiveness. Compared to InfiFusion*, the best-performing baseline on average, InfiFPO* shows an average improvement of 0.36 across 11 benchmarks. The improvements are particularly significant for instruction-following and code tasks, with a 4.4 improvement on IFEval and a 2.4 improvement on HumanEval. More importantly, InfiFPO* requires only 34% of the GPU hours compared to InfiFusion* (55 vs 160). This is thanks to our implicit model fusion objective, which replaces token-level KL with sequence-level KL. This strategy preserves probability information while avoiding complex vocabulary alignment processes, making InfiFPO more efficient and scalable.

InfiFPO consistently outperforms preference optimization baselines. After training on half of the data's preferred responses $\boldsymbol{y}_w$, SFT improved by 1.62 on average compared to the original model. Then, using the remaining half of the data for preference optimization, InfiFPO outperformed all preference optimization baselines. Compared to SFT-WRPO (the best-performing baseline), InfiFPO showed an average improvement of 0.53 across 11 benchmarks while using about the same

https://doi.org/10.52202/085713-0529

amount of GPU hours (58 vs 57). This demonstrates that our InfiFPO can better incorporate source model knowledge to enhance performance without significantly increasing training time compared to existing preference optimization baselines.

4.3 Analysis

We conduct two analyses: 1) examining the effectiveness of InfiFPO when combined with other preference optimization objectives; and 2) evaluating how different fusion strategies influence InfiFPO's performance. Due to space limitations, we only report three metrics: **Math** represents the average results from GSM8K, MATH, and ThmQA; **Code** represents the average of MBPP and HEval; and **All** represents the average results across all 11 benchmarks.

Adaptation to Other Preference Optimization Objectives. InfiFPO is fundamentally an improvement on the reference model, making it applicable not only to DPO but also to other DPO variants. To verify the versatility of InfiFPO, we integrated it with IPO and WRPO, creating InfiFPO$_{\text{IPO}}$ and InfiFPO$_{\text{WRPO}}$ respectively. Table 3 demonstrates InfiFPO's versatility as a general enhancement framework that can be effectively integrated with various preference optimization objectives beyond standard DPO. When integrated with WRPO, InfiFPO delivers modest but consistent gains in mathematical reasoning (+0.4), code generation (+0.2), and overall performance (+0.5). The improvements are more pronounced when combining InfiFPO with IPO, yielding notable enhancements in Math (+1.1) and Code (+0.5) benchmarks, with an overall improvement of +0.8 across all tasks. These results empirically validate that InfiFPO's fusion-based approach provides complementary benefits to existing preference optimization techniques by leveraging diverse source models while maintaining the pivot model's optimization objective. The loss functions for InfiFPO$_{\text{IPO}}$ and InfiFPO$_{\text{WRPO}}$ can be found in Appendix C.

Method	Math	Code	All
WRPO	75.94 ↓0.0	84.84 ↓0.0	82.80 ↓0.0
InfiFPO$_{\text{WRPO}}$	76.31 ↑0.4	85.20 ↑0.2	83.32 ↑0.5
IPO	75.29 ↓0.0	83.93 ↓0.0	82.38 ↓0.0
InfiFPO$_{\text{IPO}}$	76.37 ↑1.1	84.54 ↑0.5	83.15 ↑0.8

Table 3: Results of InfiFPO combined with different preference optimization objectives.

Different Fusion Strategies. In §3.2, we proposed the max-marginal fusion strategy to extract unique information from source models. Additionally, we explored two other fusion strategies: an average-based strategy and a confidence-based strategy. For the average-based strategy, we treat all source models equally, assigning each a weight of $1/N$. For the confidence-based strategy, we weight source models according to their confidence in the response, measured by sequence probability, meaning models with higher confidence receive greater weights. Figure 2 illustrates the performance of InfiFPO under different fusion strategies. We observe that the Confidence-based strategy slightly outperforms the Average-based strategy, likely because it emphasizes high-confidence models during fusion, thus acquiring better information. The Max-marginal strategy consistently outperforms both the Average-based and Confidence-based strategies, as it focuses on the most distinctive models, thereby learning more unique and complementary information. Detailed fusion strategies can be found in Appendix D.

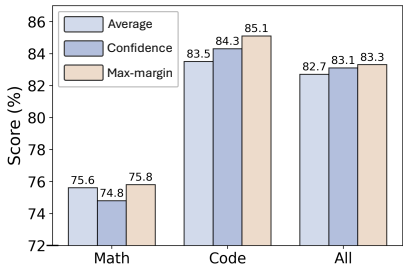

Figure 2: Different fusion strategies.

4.4 Albation Study

To investigate the effects of different components in InfiFPO, we conducted ablation experiments in three parts: Length Normalization & Probability clipping, Number of Source Models.

Length Normalization and Probability Clipping. Table 4 demonstrates the critical importance of both techniques. Without these optimizations, InfiFPO actually underperforms the baseline Phi4 model. Length Normalization alone provides substantial improvements across all metrics (+2.4 in Math, +4.5 in Code, +2.9 in All), addressing the length bias issue. Similarly, Probability Clipping

Method	LN	PC	Math	Code	All
Phi4			72.85 ↓0.0	79.47 ↓0.0	79.95 ↓0.0
InfiFPO			72.72 ↓0.1	78.42 ↓1.0	79.51 ↓0.4
	✓		75.06 ↑2.4	83.99 ↑4.5	82.86 ↑2.9
		✓	74.39 ↑1.3	81.32 ↑1.9	81.41 ↑1.5
	✓	✓	**75.80** ↑**3.1**	**85.15** ↑**5.6**	**83.33** ↑**3.3**

Table 4: Albation Study on Length Normalization (LN) and Probability Clipping (PC).

yields notable gains (+1.3 in Math, +1.9 in Code,
+1.5 in All) by preventing source model degradation. The combination of both techniques delivers the strongest performance boost, with significant improvements across all benchmarks (+3.1 in Math, +5.6 in Code, +3.3 in All), confirming their complementary nature in enhancing model capabilities.

Number of Source Models. We conducted experiments on InfiFPO with an increasing number of source models, adding different models one by one for fusion. The experiment started with a single source model, Qwen2.5-Instruct, then gradually added Mistral-Small, Qwen2.5-Coder, Gemma-3-Instruct, and Qwen2.5-Math. Table 5 shows clear performance improvements as the number of source models increases. With each additional model, we observe consistent gains across all metrics, particularly in Code tasks (+3.2 points with 5 models). The diminishing returns after 4-5 models suggest a balance between performance benefits and computational resources when selecting source models for fusion.

Num	Math	Code	All
1	74.91 ↓0.0	81.91 ↓0.0	81.54 ↓0.0
2	75.53 ↑0.6	82.49 ↑0.5	82.20 ↑0.6
3	75.60 ↑0.6	83.99 ↑2.0	82.74 ↑1.2
4	75.25 ↑0.3	**85.21 ↑3.3**	83.14 ↑1.6
5	**75.80 ↑0.8**	85.15 ↑3.2	**83.33 ↑1.7**

Table 5: Albation Study on Number of Source Models.

5 Related Work

Preference Optimization. Aligning LLMs with human preferences often relies on RLHF [30, 31], where a reward model is trained from human comparisons and used in PPO to optimize the policy. While effective, RLHF is resource-intensive and complex to converge. To simplify preference alignment, DPO [8] reformulates the objective as offline learning from preference pairs, removing the need for a reward model and RL. Azar et al. [29] unify RLHF and DPO under a general preference optimization framework, and propose IPO, a variant that avoids overfitting by using the identity transformation, offering more stable learning in low-data or biased settings. Gu et al. [32] further propose the PAD framework, which models preference knowledge as a probability distribution over responses, providing more nuanced supervisory signals for preferences distilling.

Model Fusion. Integrating models aims to combine multiple LLMs into a single model that inherits their respective strengths. While earlier methods like model merging [33–38] require architectural compatibility, fusion techniques relax such constraints, making them more suitable for heterogeneous models. Another line of work, knowledge distillation (KD) [39–41], offers an alternative approach for transferring capabilities across models without requiring structural homogeneity. KD transfers the generalization ability of larger models to smaller ones by leveraging "soft targets" the output probability distributions (logits) of the teacher model enabling efficient deployment while preserving performance. However, traditional KD typically assumes that the teacher model is larger than the student and fully covers the target capability space. Moreover, most prior work has focused on the single teacher setting [42, 43], limiting its flexibility in multi-teacher scenarios.

FuseLLM [2] introduced LLM fusion, showing gains in reasoning and code tasks. Later works extended this idea across domains [44], but struggled with structural mismatches. To address this, pairwise fusion [2] sequentially integrates models into the pivot model, and InfiFusion [4] improved it via adaptive merging and unified output aggregation. The most explicit fusion approaches operate at the token level and face vocabulary alignment issues. WRPO [6] mitigates this by shifting fusion to the sequence level using reinforcement learning, enabling implicit fusion without vocabulary conflicts. However, WRPO has two major limitations: it discards probabilistic outputs using only response-level supervision and focuses solely on preferred responses while ignoring contrastive signals from dispreferred ones, resulting in only partial leverage of source model capabilities.

6 Conclusion

We introduce **InfiFPO**, a novel framework that enables model fusion during the preference alignment phase by replacing the reference model in DPO with a fused sequence-level distribution over multiple source models. Unlike prior work, InfiFPO leverages full-sequence probabilities rather than token-level outputs, thereby avoiding vocabulary alignment issues across heterogeneous models while preserving rich preference signals. To make this optimization practical and efficient, we derive an offline training objective from a sequence-KL constrained RLHF formulation, termed FuseRLHF.

https://doi.org/10.52202/085713-0529

We further enhance stability through three key strategies: length normalization to mitigate sequence bias, probability clipping to suppress noisy gradients, and max-margin fusion to prioritize diverse and informative sources. Experiments across 11 benchmarks demonstrate that InfiFPO consistently outperforms strong baselines in both model capability and alignment quality. Our results highlight that preference optimization not only accommodates but can significantly benefit from principled model fusion, offering a robust and scalable path to integrating diverse LLMs.

Limitation

Despite InfiFPO demonstrating significant empirical performance, it still relies on existing preference optimization methods such as DPO. More rigorous theoretical analysis is needed to better understand InfiFPO's fusion mechanisms and further strengthen its theoretical foundation. Additionally, due to computational resource constraints, we only selected five mainstream open-source LLMs as source models for our experiments, which cannot represent the SOTA performance of current advanced LLMs. Experiments with larger-scale models and datasets remain unexplored.

Funding Transparency Statement

Funding in direct support of this work. This research was supported by the team of Prof. Hongxia Yang at The Hong Kong Polytechnic University. GPU resources were donated by InfiX.ai.

Competing Interests. The authors declare no competing interests.

References

[1] Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion of large language models. In ICLR, 2024.

[2] Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. Fusechat: Knowledge fusion of chat models, 2025.

[3] Yuanyi Wang, Zhaoyi Yan, Yiming Zhang, Qi Zhou, Yanggan Gu, Fei Wu, and Hongxia Yang. Infigfusion: Graph-on-logits distillation via efficient gromov-wasserstein for model fusion, 2025. URL https://arxiv.org/abs/2505.13893.

[4] Zhaoyi Yan, Yiming Zhang, Baoyi He, Yuhao Fu, Qi Zhou, Zhijie Sang, Chunlin Ji, Shengyu Zhang, Fei Wu, and Hongxia Yang. Infifusion: A unified framework for enhanced cross-model reasoning via llm fusion, 2025.

[5] Qi Zhou, Yiming Zhang, Yanggan Gu, Yuanyi Wang, Zhijie Sang, Zhaoyi Yan, Zhen Li, Shengyu Zhang, Fei Wu, and Hongxia Yang. Democratizing AI through model fusion: A comprehensive review and future directions. Nexus, October 2025. ISSN 2950-1601. URL https://doi.org/10.1016/j.ynexs.2025.100102.

[6] Ziyi Yang, Fanqi Wan, Longguang Zhong, Tianyuan Shi, and Xiaojun Quan. Weighted-reward preference optimization for implicit model fusion. In The Thirteenth International Conference on Learning Representations, 2025.

[7] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024.

[8] 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 NeurIPS, 2023.

[9] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016.

[10] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[11] Gemma Team. Gemma 3 technical report, 2025.

[12] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical report, 2024.

[13] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. arXiv preprint arXiv:2409.12122, 2024.

[14] Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14255–14273, 2024.

[15] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. Hugging Face repository, 13:9, 2024.

[16] Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding. 2025.

[17] Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. arXiv preprint arXiv:2410.18451, 2024.

[18] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In ACL (Findings), 2023.

[19] Vikas Yadav, Steven Bethard, and Mihai Surdeanu. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2578–2589, 2019.

[20] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In International Conference on Learning Representations, .

[21] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

[22] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), .

[23] Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2023.

 https://doi.org/10.52202/085713-0529

[24] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. arXiv preprint arXiv:2108.07732, 2021.

[25] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021.

[26] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2368–2378, 2019.

[27] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2019.

[28] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. arXiv preprint arXiv:2311.07911, 2023.

[29] Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences, 2023.

[30] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. Advances in neural information processing systems, 30, 2017.

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

[32] Yanggan Gu, Junzhuo Li, Sirui Huang, Xin Zou, Zhenghua Li, and Xuming Hu. Capturing nuanced preferences: Preference-aligned distillation for small language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, Findings of the Association for Computational Linguistics: ACL 2025, pages 15959–15973, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.822. URL https://aclanthology.org/2025.findings-acl.822/.

[33] Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, Advances in Neural Information Processing Systems, volume 36, pages 66727–66754. Curran Associates, Inc., 2023.

[34] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. Advances in Neural Information Processing Systems, 36:7093–7115, 2023.

[35] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In International conference on machine learning, pages 23965–23998. PMLR, 2022.

[36] Yuanyi Wang, Yanggan Gu, Yiming Zhang, Qi Zhou, Zhaoyi Yan, Congkai Xie, Xinyao Wang, Jianbo Yuan, and Hongxia Yang. Model merging scaling laws in large language models, 2025. URL https://arxiv.org/abs/2509.24244.

[37] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch, 2024.

[38] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. Nature Machine Intelligence, pages 1–10, 2025.

[39] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

[40] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. arXiv preprint arXiv:1908.09355, 2019.

[41] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108, 2019.

[42] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In The Twelfth International Conference on Learning Representations, 2024.

[43] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. arXiv preprint arXiv:2306.08543, 2023.

[44] Yiming Zhang, Baoyi He, Shengyu Zhang, Yuhao Fu, Qi Zhou, Zhijie Sang, Zijin Hong, Kejing Yang, Wenjun Wang, Jianbo Yuan, Guanghan Ning, Linyi Li, Chunlin Ji, Fei Wu, and Hongxia Yang. Unconstrained model merging for enhanced llm reasoning, 2024. URL https://arxiv.org/abs/2410.13699.

[45] OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

[46] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=1qvx610Cu7.

[47] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems, 30, 2017.

[48] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In Advances in Neural Information Processing Systems, pages 5574–5584, 2017.

[49] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled ai feedback, 2024. URL https://arxiv.org/abs/2310.01377.

A Mathematical Derivations

A.1 Deriving the FPO Objective

In this appendix, we will derive Eq. (7). Based on Eq. (14), we optimize the following objective:

$$\arg\max_{\mathcal{M}^p} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})] - \beta\sum_i^N \gamma_i\bigg(\log\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x}) - \log\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x})\bigg), \tag{14}$$

then we have:

$$\begin{aligned}
&\arg\max_{\mathcal{M}^p} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})] - \beta\sum_i^N \gamma_i\mathbb{D}_{\mathrm{SKL}}\left[\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})\|\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x})\right] \\
&\quad= \arg\max_{\mathcal{M}^p} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{y}\sim\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})}\left[\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y}) - \beta\log\frac{\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})}{\mathcal{M}_{\mathrm{fu}}^s(\boldsymbol{y}|\boldsymbol{x})}\right] \\
&\quad= \arg\min_{\mathcal{M}^p} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{y}\sim\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})}\left[\log\frac{\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})}{\mathcal{M}_{\mathrm{fu}}^s(\boldsymbol{y}|\boldsymbol{x})} - \frac{1}{\beta}\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})\right] \\
&\quad= \arg\min_{\mathcal{M}^p} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{y}\sim\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})}\left[\log\frac{\mathcal{M}^p(\boldsymbol{y}|\boldsymbol{x})}{\frac{1}{Z(\boldsymbol{x})}\mathcal{M}_{\mathrm{fu}}^s(\boldsymbol{y}|\boldsymbol{x})\exp(\frac{1}{\beta}\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y}))} - \log Z(\boldsymbol{x})\right]
\end{aligned} \tag{15}$$

 https://doi.org/10.52202/085713-0529

where the fused source model $\mathcal{M}^s_{\text{fu}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{N} \left(\mathcal{M}^s_i(\boldsymbol{y}|\boldsymbol{x})\right)^{\gamma_i}$ and the partition function $Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \mathcal{M}^s_{\text{fu}}(\boldsymbol{y}|\boldsymbol{x}) \exp\left(\frac{1}{\beta}\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})\right)$.

Note that the partition function is a function of only $\boldsymbol{x}$ and the fused source model $\mathcal{M}^s_{\text{fu}}$, but does not depend on the pivot model $\mathcal{M}^p$. Hence we have the optimal solution $\mathcal{M}^{p*}$:

$$\mathcal{M}^{p*}(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(x)}\mathcal{M}^s_{\text{fu}}(\boldsymbol{y}|\boldsymbol{x}) \exp(\frac{1}{\beta}\boldsymbol{r}(\boldsymbol{x},\boldsymbol{y})). \tag{16}$$

This completes the derivation.

A.2 Deriving the Gradient of the InfiFPO Objective

In this section, we derive the gradient of the InfiFPO objective, where the parameters of the pivot model are denoted as θ:

$$\nabla\theta\mathcal{L}_{\text{InfiFPO}}(\mathcal{M}^p;\{\mathcal{M}^s_i\}_{i=1}^N) = -\nabla\theta\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}^p}\left[\log\sigma\left(\beta\overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}_w|\boldsymbol{x})} - \beta\overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_l|\boldsymbol{x})}{\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}_l|\boldsymbol{x})}\right)\right], \tag{17}$$

where $\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{N}\left[\text{Clip}(\mathcal{M}^s_i(\boldsymbol{y}|\boldsymbol{x}))\right]^{\gamma_i}$.

We can rewrite Eq. (17) as

$$\nabla\theta\mathcal{L}_{\text{InfiFPO}}(\mathcal{M}^p;\{\mathcal{M}^s_i\}_{i=1}^N) = -\nabla\theta\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}^p}\left[\log\sigma\left(\frac{\sigma'(u)}{\sigma(u)}\nabla\theta(u)\right)\right], \tag{18}$$

where $u = \beta\overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}_w|\boldsymbol{x})} - \beta\overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_l|\boldsymbol{x})}{\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}_l|\boldsymbol{x})}$.

With the properties of sigmoid function $\sigma'(x) = \sigma(x)(1-\sigma(x))$ and $\sigma(x) = 1-\sigma(-x)$, we can get the gradient:

$$\nabla\theta\mathcal{L}_{\text{InfiFPO}} = -\beta\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}^p}\left[\sigma\left(\beta\overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_l|\boldsymbol{x})}{\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}_l|\boldsymbol{x})} - \beta\overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}_w|\boldsymbol{x})}\right)\left[\nabla\theta\log\mathcal{M}^p(\boldsymbol{y}_w|\boldsymbol{x}) - \nabla\theta\log\mathcal{M}^p(\boldsymbol{y}_l|\boldsymbol{x})\right]\right].$$

After rewriting the first term, we obtain the final form of the gradient in § 3.3.

B Evaluation Setup

We adopt OpenCompass [45] and EvalPlus [46] to conduct evaluation on 11 benchmark datasets. We revise the prompts of certain datasets within OpenCompass to ensure more reliable answer extraction via its regex-based matching mechanism. The detailed prompts are listed in Table 6.

For TheremQA and HumanEval, we follow the default evaluation settings without modifying the prompts. For MBPP, we employ EvalPlus [46] for rigourous evaluation of LLM-synthesized code.

C Adaptation to Other Preference Optimization Objectives

The original objective of WRPO is

$$\mathcal{L}_{\text{WRPO}}(\mathcal{M}_\theta;\mathcal{M}_{\text{ref}}) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_{w_s},\boldsymbol{y}_{w_p},\boldsymbol{y}_l)\sim\mathcal{D}}\left[\log\sigma\left(\alpha\cdot\beta\log\frac{\mathcal{M}_\theta(\boldsymbol{y}_{w_s}|x)}{\mathcal{M}_{\text{ref}}(\boldsymbol{y}_{w_s}|x)} + (1-\alpha)\cdot\beta\log\frac{\mathcal{M}_\theta(\boldsymbol{y}_{w_p}|x)}{\mathcal{M}_{\text{ref}}(\boldsymbol{y}_{w_p}|x)} - \beta\log\frac{\mathcal{M}_\theta(\boldsymbol{y}_l|x)}{\mathcal{M}_{\text{ref}}(\boldsymbol{y}_l|x)}\right)\right], \tag{19}$$

where $\boldsymbol{y}_{w_s}$ is the preferred response generated by source models, and $\boldsymbol{y}_{w_p}$ is the dispreferred response generated by the pivot model. α represents the fusion coefficient that dynamically balances the reward of $\boldsymbol{y}_{w_s}$ and $\boldsymbol{y}_{w_p}$. After adapting InfiFPO to this objective, we can obtain

$$\mathcal{L}_{\text{InfiFPO-WRPO}}(\mathcal{M}^p;\{\mathcal{M}^s_i\}_{i=1}^N\}) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_{w_s},\boldsymbol{y}_{w_p},\boldsymbol{y}_l)\sim\mathcal{D}}\left[\overline{\log}\sigma\left(\alpha\cdot\beta\overline{\log}\frac{\mathcal{M}_\theta(\boldsymbol{y}_{w_s}|x)}{\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}_{w_s}|x)} + (1-\alpha)\cdot\beta\overline{\log}\frac{\mathcal{M}_\theta(\boldsymbol{y}_{w_p}|x)}{\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}_{w_p}|x)} - \beta\overline{\log}\frac{\mathcal{M}_\theta(\boldsymbol{y}_l|x)}{\mathcal{M}^{\text{sclip}}_{\text{fu}}(\boldsymbol{y}_l|x)}\right)\right]. \tag{20}$$

Table 6: Prompt format used for different datasets.

Dataset	Prompt Format
IFEval	`{prompt}\nPlease directly give the correct answer:`
ARC-C	`Question: {question}\nA . {textA}\nB . {textB}\nC . {textC}\nD . {textD}\nDirectly give me the correct answer option, and then explain:`
Hellaswag	`{ctx}\nQuestion : Which ending makes the most sense?\nDirectly give me the correct choice, you can further explain it or not.\nA . {A}\nB . {B}\nC . {C}\nD . {D}\nYou may choose from 'A', 'B', 'C', 'D'.\nAnswer :`
BBH	`Follow the given examples and answer the question.\n {_hint}\nQ : {{input}}\nA : Let's think step by step.`
DROP	`You will be asked to read a passage and answer a question. Some examples of passages and Q\&A are provided below.\n {drop_examples}\n \n ## Your Task\n --\n {prompt}\nThink step by step, then write a line of the form "Answer: \$ANSWER" at the end of your response.`
MMLU	`{_hint}\nQuestion : {{input}}\nA . {{A}}\nB . {{B}}\nC . {{C}}\nD . {{D}}\n \nFor simple problems:\nDirectly provide the answer with minimal explanation.\n \nFor complex problems:\nUse this step-by-step format:\n #### Step 1: [Concise description]\n [Brief explanation]\n #### Step 2: [Concise description]\n [Brief explanation]\n \nRegardless of the approach, always conclude with:\nThe answer is [the_answer_letter].\nwhere the [the_answer_letter] is one of A, B, C or D.\n \nLet 's think step by step.`
GSM8K	`{question}\nPlease reason step by step, and put your final answer within \\boxed\{\}.`
MATH	`{problem}\nPlease reason step by step, and put your final answer within \\boxed\{\}.`

The original objective of the IPO is

$$\mathcal{L}_{\text{IPO}}(\mathcal{M}_\theta; \mathcal{M}_{\text{ref}}) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}^p}\left[\left(\beta\overline{\log}\frac{\mathcal{M}_\theta(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}_{\text{ref}}(\boldsymbol{y}_w|\boldsymbol{x})} - \beta\overline{\log}\frac{\mathcal{M}_\theta(\boldsymbol{y}_l|\boldsymbol{x})}{\mathcal{M}_{\text{ref}}(\boldsymbol{y}_l|\boldsymbol{x})}\right) - \frac{1}{2\beta}\right]^2. \quad (21)$$

Since IPO already implements length normalization[7], the main changes when adapting to InfiFPO are in the reference model part.

$$\mathcal{L}_{\text{InfiFPO-IPO}}(\mathcal{M}^p; \{\mathcal{M}_i^s\}_{i=1}^N) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}^p}\left[\left(\overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_w|\boldsymbol{x})}{\mathcal{M}_{\text{fu}}^{\text{sclip}}(\boldsymbol{y}_w|\boldsymbol{x})} - \overline{\log}\frac{\mathcal{M}^p(\boldsymbol{y}_l|\boldsymbol{x})}{\mathcal{M}_{\text{fu}}^{\text{sclip}}(\boldsymbol{y}_l|\boldsymbol{x})}\right) - \frac{1}{2\beta}\right]^2 \quad (22)$$

D Different Fusion Strategies

When we focus on the fused source model $\mathcal{M}_{\text{fu}}^s$, which is the weighted geometric mean of all source model sequence probabilities, we can find that it originates from the model fusion objective. By changing its weights, we can adopt different fusion strategies. In addition to the Max-margin fusion strategy mentioned in § 3.2, we design two following strategies. Their results can be viewed in § 4.3.

Average-based Fusion. A common fusion strategy is to treat N source models equally, adopting the same weight of $\frac{1}{N}$, thus the fusion model becomes

$$\mathcal{M}_{\text{fu}}^s = \prod_{i=1}^{N} \left(\mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x})\right)^{\frac{1}{N}} \quad (23)$$

Confidence-based Fusion. To dynamically weight the source models, we introduce a confidence-based strategy and define the confidence of each source model $\mathcal{M}_i^s$ as the inverse of its negative log-likelihood:

$$\text{Confidence}_i(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{-\log \mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x}) + \epsilon}, \quad (24)$$

[7] https://huggingface.co/blog/pref-tuning

https://doi.org/10.52202/085713-0529

where ϵ is a small constant ensuring numerical stability [47]. This design follows the intuition that $-\log \mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x})$ quantifies the information content of a prediction, where smaller values indicate higher confidence. Taking its inverse reflects the common practice of inverse-uncertainty weighting [48], assigning larger weights to more confident models. This fusion mechanism emphasizes source models with higher confidence, allowing $\mathcal{M}_{\text{fu}}^s$ to adaptively inherit more reliable information from the sources. The corresponding normalized weight for $\mathcal{M}_i^s$ is given by SoftMax:

$$w_i(\boldsymbol{y}|\boldsymbol{x}) = \frac{\text{Confidence}_i(\boldsymbol{y}|\boldsymbol{x})}{\sum_{j=1}^{N} \text{Confidence}_j(\boldsymbol{y}|\boldsymbol{x})}. \tag{25}$$

Based on these confidence-derived weights, the fused model $\mathcal{M}_{\text{fu}}^s$ aggregates source models via:

$$\mathcal{M}_{\text{fu}}^s(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{N} \mathcal{M}_i^s(\boldsymbol{y}|\boldsymbol{x})^{w_i(\boldsymbol{y}|\boldsymbol{x})}. \tag{26}$$

E Supplementary Experiments and Analysis

E.1 Additional Experimental Results

E.1.1 Data Diversity Validation

To validate InfiFPO's robustness to data diversity, we conducted experiments on UltraFeedback [49], which contains approximately 60k preference pairs and has been widely adopted in preference optimization research. We treated the preferred (chosen) responses as labels for model fusion baselines.

We selected three general-purpose models for fusion: 1) Qwen2.5-14B-Instruct; 2) Mistral-Small; 3) Gemma-3-Instruct. We compared against InfiFusion and SFT-WRPO, which demonstrated strong performance in our main experiments. As shown in Table 7, InfiFPO remains effective on UltraFeedback and outperforms relevant baselines.

Model/Method	Math	Code	All
Phi-4	72.86	79.47	79.95
InfiFusion	73.23	82.43	81.44
SFT-WRPO	73.15	82.21	81.58
InfiFPO	**73.59**	**83.15**	**82.19**

Table 7: Experimental results on UltraFeedback

However, the optimal performance on UltraFeedback (82.19) is notably lower than on our constructed dataset (83.33), validating the effectiveness of our data construction process.

E.1.2 Strong Backbone Validation

We conducted experiments using Qwen2.5-14B-Instruct as the target (pivot) model, selecting three source models: 1) Phi-4; 2) Mistral-Small; 3) Gemma-3-Instruct. We used FuseChat, InfiFusion, and WRPO as baselines with identical experimental settings to our main experiments.

Model / Method	Math	Code	All
Qwen-2.5-Instruct	72.18	82.62	80.97
FuseChat	75.96	83.50	83.22
InfiFusion	76.09	83.38	83.16
SFT-WRPO	76.42	84.29	83.46
InfiFPO	**76.46**	**85.41**	**84.01**

Table 8: Experimental results using Qwen2.5-14B-Instruct as target model

Table 8 demonstrates that when using Qwen2.5-Instruct as the target model, our method still outperforms other baselines, proving its generalizability. In our main experiments, we chose Phi-4 over Qwen2.5-Instruct as the target model because Phi-4 is heterogeneous with all source models (different architectures, vocabularies, etc.), representing a more general and typical scenario for model fusion.

E.1.3 Reward Model Sensitivity Analysis

To examine data sensitivity, we tested the impact of different reward models on method performance. We tested two reward models: Skywork-Reward-Llama-3.1-8B-v0.2 and ArmoRM-Llama3-8B-v0.1, both widely used in preference optimization.

Model / Method	Math	Code	All
Skywork-Reward-Llama-3.1-8B-v0.2			
Phi-4	72.86	79.47	79.95
InfiFusion	74.32	82.47	81.96
SFT-WRPO	74.28	83.44	81.93
InfiFPO	**74.53**	**84.88**	**82.67**
ArmoRM-Llama3-8B-v0.1			
Phi-4	72.86	79.47	79.95
InfiFusion	73.38	82.54	81.59
SFT-WRPO	73.07	83.04	81.62
InfiFPO	**73.45**	**84.94**	**82.46**

Table 9: Results using different reward models

We observe two phenomena from Table 9: (1) InfiFPO consistently outperforms baselines when using different reward models; (2) Different reward models yield different performance improvements. According to Reward Bench v2, Skywork-Reward-Gemma-2-27B-v0.2 significantly outperforms the other two reward models, indicating a positive correlation between reward model performance and final model performance.

E.1.4 Sequence Alignment Ablation

We conducted ablation experiments to investigate the impact of sequence alignment on model fusion methods. We removed sequence alignment from all model fusion methods for fair comparison.

Model / Method	Math	Code	All
Phi-4	72.86	79.47	79.95
FuseChat	73.54	82.97	82.12
- without sequence alignment	72.30	79.16	79.43
InfiFusion	75.54	82.67	82.38
- without sequence alignment	73.47	79.98	80.11
InfiFPO	75.60	83.99	82.74
- without sequence alignment	**73.72**	**81.28**	**80.82**

Table 10: Impact of sequence alignment on model fusion methods

Results in Table 10 show that even without sequence alignment, InfiFPO outperforms other model fusion baselines. Token-level fusion methods like FuseChat without sequence alignment even perform worse than the base Phi-4 model, indicating greater dependence on sequence alignment. Therefore, we consider addressing vocabulary conflicts essential for model fusion methods.

E.1.5 Data Scaling Analysis

We tested different data volumes (25k/50k/75k/100k) during preference optimization to analyze performance scaling.

https://doi.org/10.52202/085713-0529

Model / Method	25k	50k	75k	100k
Phi-4	79.95	79.95	79.95	79.95
SFT-WRPO	81.58	82.14	82.80	82.89
InfiFPO	**82.55**	**82.99**	**83.33**	**83.56**

Table 11: Performance scaling with data volume

Table 11 shows consistent performance improvements across different data volumes, with gains increasing with larger datasets, demonstrating the scalability of our approach.

E.1.6 Small-scale Model Validation

We experimented with Qwen2.5-Instruct-1.5B/3B/7B as pivot models to validate our method's effectiveness across different model scales.

Model / Method	Math	Code	All
Qwen2.5-1.5B-Instruct			
Base Model	46.02	47.16	50.31
InfiFusion	49.59	50.49	53.11
SFT-WRPO	50.16	51.21	53.70
InfiFPO	**50.70**	**52.75**	**54.61**
Qwen2.5-3B-Instruct			
Base Model	55.88	62.26	64.07
InfiFusion	58.82	65.33	66.72
SFT-WRPO	59.31	65.63	67.01
InfiFPO	**60.05**	**66.64**	**67.96**
Qwen2.5-7B-Instruct			
Base Model	62.48	70.13	73.04
InfiFusion	65.09	72.24	75.35
SFT-WRPO	65.45	72.51	75.65
InfiFPO	**66.14**	**73.31**	**76.24**

Table 12: Results with different small-scale pivot models

Table 12 demonstrates that our method outperforms model fusion and preference optimization baselines across different pivot model sizes, confirming scalability and effectiveness.

E.2 Technical Details and Method Analysis

E.2.1 Technical Implementation Details

Pre-computation Strategy: In practice, we pre-compute log probabilities from target (pivot) and source models before training, which provides two key advantages: (1) Reduced training memory footprint by avoiding the need to load multiple source models during training; (2) Accelerated computation through efficient third-party inference libraries.

We employ vLLM for acceleration, requiring only approximately 8 GPU hours to compute log probabilities for 5 source models. This explains why our GPU hours increase by merely 10% compared to vanilla DPO when fusing multiple source models.

Memory Efficiency: InfiFPO offers significant memory efficiency advantages. It performs model fusion at the sequence level, requiring only sequence-level log probabilities (a few floating-point numbers per sequence). When the number of source models increases, training efficiency remains largely unaffected. In contrast, token-level fusion methods must load probability distributions over

the entire vocabulary (100k tokens per token postion), requiring 100k-scale floating-point numbers per token that scale linearly with source model count.

E.2.2 In-depth Analysis of Method Principles

We provide a detailed discussion of how source models are selected during Probability Clipping (PC) and Max-Margin fusion strategies.

Probability Clipping and Max-Margin Process: PC aims to avoid interference from weak source models by clipping the corresponding log probabilities, ensuring that source models do not assign lower probabilities to preferred responses or higher probabilities to dispreferred responses than the pivot model. Max-Margin selects the source model with the most different preference knowledge from the pivot model as the reference model.

When using PC, Max-Margin selects the "smartest" source model, meaning the source model that assigns probabilities to preferred responses no lower than the pivot model, probabilities to dispreferred responses no higher than the pivot model, and has the largest margin between preferred and dispreferred responses.

Example Illustration: Given two responses, the probability values assigned by models are:

Model type	Logps (preferred)	Logps (dispreferred)
Pivot model	-0.4	-0.7
Source model 1	-0.6	-0.5
Source model 2	-0.3	-0.8

Table 13: Example of model probabilities before PC

After PC, the source models' log probabilities become:

Model type	Logps (preferred)	Logps (dispreferred)	Selected
Pivot model	-0.4	-0.7	-
Source model 1	-0.4	-0.7	-
Source model 2	-0.3	-0.8	✓

Table 14: Example of model probabilities after PC

Based on the Max-Margin strategy, source model 2 serves as the reference model. It is the "smartest" source model, having the highest log probabilities for preferred responses and the lowest log probabilities for dispreferred responses.

This strategy aligns with findings in FuseLLM, which adopts the MinCE strategy to select source models with the lowest cross-entropy for ground truth. Both approaches indicate that in model fusion, fusing knowledge from multiple source models may not be as effective as fusing knowledge from the "best" source model.

F Case Study

F.1 Advantages in Code Generation

To evaluate the effectiveness of our proposed InfiFPO strategy, we conduct a detailed comparison with the baseline Supervised Fine-Tuning (SFT) model on the HumanEval benchmark.

Through case studies, we observe that InfiFPO consistently exhibits superior capabilities in two critical aspects: (1) precise task understanding and (2) concise and robust code generation. For instance, in the remove_duplicates task, SFT misconstrues the requirement by retaining the first occurrence of duplicates, leading to incorrect outputs. In contrast, InfiFPO accurately identifies the need to eliminate all repeated elements, employing a frequency-counting approach that aligns with the task's semantics. Similarly, in the vowels_count task, InfiFPO adopts a minimal yet effective implementation

 https://doi.org/10.52202/085713-0529

OpenAi HumanEval - **Task Type:** Code Generation

InfiFPO Behavior	**Baseline Behavior**
"prompt_1": "from typing import List\n\n\ndef remove_duplicates(numbers: List[int]) ->List[int]:\n \"\"\" From a list of integers, remove all elements that occur more than once.\n Keep order of elements left the same as in the input.\n >>>remove_duplicates([1, 2, 3, 2, 4])\n [1, 3, 4]\n \"\"\"\n"	
"completion": "def remove_duplicates(numbers: List[int]) ->List[int]:\n \"\"\" From a list of integers, remove all elements that occur more than once.\n Keep order of elements left the same as in the input.\n >>>remove_duplicates([1, 2, 3, 2, 4])\n [1, 3, 4]\n \"\"\"\n counts = {}\n for number in numbers:\n counts[number] = counts.get(number, 0) + 1\n \n result = []\n for number in numbers:\n if counts[number] == 1:\n result.append(number)\n \n return result\n","result": "passed","passed": true	"completion": "from typing import List\n\ndef remove_duplicates(numbers: List[int]) ->List[int]:\n \"\"\" From a list of integers, remove all elements that occur more than once.\n Keep order of elements left the same as in the input.\n >>>remove_duplicates([1, 2, 3, 2, 4])\n [1, 3, 4]\n \"\"\"\n seen = set()\n duplicates = set()\n result = []\n\n for number in numbers:\n if number in duplicates:\n continue\n if number in seen:\n duplicates.add(number)\n else:\n seen.add(number)\n result.append(number)\n\n return result\n","result": "failed: ","passed": false
"prompt_2": "\nFIX = \"\"\"\nAdd more test cases.\n\"\"\"\n\ndef vowels_count(s):\n \"\"\"Write a function vowels_count which takes a string representing\n a word as input and returns the number of vowels in the string.\n Vowels in this case are 'a', 'e', 'i', 'o', 'u'. Here, 'y' is also a\n vowel, but only when it is at the end of the given word.\n\n Example:\n >>>vowels_count(\"abcde\")\n 2\n >>>vowels_count(\"ACEDY\")\n 3\n \"\"\"\n"	
"completion": "def vowels_count(s):\n \"\"\"Write a function vowels_count which takes a string representing\n a word as input and returns the number of vowels in the string.\n Vowels in this case are 'a', 'e', 'i', 'o', 'u'. Here, 'y' is also a\n vowel, but only when it is at the end of the given word.\n\n Example:\n >>>vowels_count(\"abcde\")\n 2\n >>>vowels_count(\"ACEDY\")\n 3\n \"\"\"\n vowels = \"aeiouAEIOU\"\n count = 0\n for char in s:\n if char in vowels:\n count += 1\n if s[-1].lower() == 'y':\n count += 1\n return count\n", "result": "passed", "passed": true	"completion": "def vowels_count(s):\n \"\"\"Write a function vowels_count which takes a string representing\n a word as input and returns the number of vowels in the string.\n Vowels in this case are 'a', 'e', 'i', 'o', 'u'. Here, 'y' is also a\n vowel, but only when it is at the end of the given word.\n\n Example:\n >>>vowels_count(\"abcde\")\n 2\n >>>vowels_count(\"ACEDY\")\n 3\n \"\"\"\n vowels = \"aeiouAEIOU\"\n count = 0\n for i, char in enumerate(s):\n if char in vowels:\n count += 1\n elif char == 'y' and i == len(s) - 1:\n count += 1\n return count\n\n# Additional test cases\nprint(vowels_count(\"hello\")) # 2\nprint(vowels_count(\"sky\")) # 1\nprint(vowels_count(\"rhythm\")) # 0\nprint(vowels_count(\"fly\")) # 1\nprint(vowels_count(\"boy\")) # 2\nprint(vowels_count(\"try\")) # 1\nprint(vowels_count(\"my\")) # 1\nprint(vowels_count(\"y\")) # 1\nprint(vowels_count(\"a\")) # 1\nprint(vowels_count(\"e\")) # 1\nprint(vowels_count(\"i\")) # 1\nprint(vowels_count(\"o\")) # 1\nprint(vowels_count(\"u\")) # 1\n", "result": "failed: Test 5","passed": false,

Table 15: Case study on code generation.

that avoids boundary errors, whereas SFT introduces unnecessary loop-index conditions prone to edge-case failures.

These results suggest that InfiFPO is not merely replicating seen patterns but actively refining the reasoning chain to bridge the gap between instruction semantics and executable code. By reducing spurious complexity and enhancing robustness against implicit corner cases, InfiFPO demonstrates a stronger alignment with real-world coding practices.

F.2 Mathematical Reasoning Perspective

Case	Problem Prompt	SFT Output	FPO Output	Gold	Comment
0	`Problem:` How many ways are there to divide a set of 8 elements into 5 non-empty ordered subsets? Please analyze and write your final answer after the phrase "The answer is":	SFT predicts 126000, by mistakenly recalling $S(8,5)$ and applying 5! permutations, leading to a significant overestimate.	InfiFPO correctly computes $S(8,5) = 105$, applies $5! = 120$, yielding $105 \times 120 = 12600$.	11760	InfiFPO demonstrates recursive computation of Stirling numbers, while SFT simply recalls a wrong value, lacking procedural reasoning.
1	`Problem:` What is the value of $\int_{-\infty}^{+\infty} \sin(3t) \cdot \sin(t/\pi)/t^2\, dt$? Please analyze and write your final answer after the phrase "The answer is":	SFT predicts 0, oversimplifying based on oscillation intuition, neglecting the structure of the integral.	InfiFPO expands product of sines, applies Fourier integral properties, identifies nonzero singular contribution, yielding the correct result.	1.0	InfiFPO's success stems from correct application of orthogonality and singularity analysis, while SFT lacks symbolic manipulation depth.
2	`Problem:` Given x = 0.157, what is the value of $x \times \frac{\prod_{n=1}^{\infty}(1-\frac{x^2}{n^2\pi^2})}{\sin(x)}$? Please analyze and write your final answer after the phrase "The answer is":	Both SFT and InfiFPO predict 0.157, recognizing Euler's reflection formula, correctly simplifying numerator and denominator.	Both predict 0.157. FPO's derivation explicitly links product form to sine function identity.	0.157	Both models give correct answer, but InfiFPO provides a more explicit and pedagogical derivation path.

Table 16: Comparison between SFT and InfiFPO on TheoremQA benchmark. InfiFPO shows stronger symbolic reasoning, procedural consistency, and structural understanding, leading to better robustness and interpretability.

We conduct a detailed comparison of InfiFPO and SFT on the TheoremQA benchmark to assess their mathematical reasoning capabilities. Our evaluation reveals that InfiFPO demonstrates superior performance in symbolic manipulation, stepwise reasoning, and the application of mathematical identities. While SFT often relies on pattern matching and heuristic recall, InfiFPO constructs solutions via complete, logically consistent derivation chains.

For instance, concerning the evaluation of the definite integral $\int_{-\infty}^{+\infty} \frac{\sin(3t)\sin(t/\pi)}{t^2}\, dt$, InfiFPO successfully decomposes the product of sine functions using trigonometric identities and correctly identifies the contribution of singularities and symmetry in determining the integral's value. In contrast, SFT prematurely concludes that the integral vanishes due to oscillation, neglecting deeper analysis of the integrand's behavior.

https://doi.org/10.52202/085713-0529

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction align with the paper's contributions and scope, providing a clear and accurate overview.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Appendix 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The complete mathematical derivation can be found in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided implementation details in section 4, and included a reproducible anonymous code repository URL in the abstract.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

https://doi.org/10.52202/085713-0529

Answer: [Yes]

Justification: In section 4, we provide the data sources, and in the abstract, we include an anonymous code repository URL for reproducibility.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided implementation details in section 4 and appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Given that calculating statistical significance requires additional resources, we did not perform these calculations. Most of our experimental results show improvements of more than 1 percentage point, indicating a certain level of significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In section 4, we provide information about the computational resources used. Moreover, in Table 2, we show the GPU hours required for model training.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research fully adheres to the NeurIPS Code of Ethics, ensuring integrity, fairness, and transparency throughout the study.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Our method focuses on improving the performance of general models, which does not directly cause social impact. Additionally, we will provide licenses for our code and model weights.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

https://doi.org/10.52202/085713-0529

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: We will provide licenses for our code and model weights in the future.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: Based on the double-blind review process, all code we provide now has been anonymized and is intended only for reviewers to check and reproduce results.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: Based on the double-blind review process, all code we provide now has been anonymized and is intended only for reviewers to check and reproduce results.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

https://doi.org/10.52202/085713-0529

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.