Select Before Use: On the Importance of Reference Model Selection in Preference Alignment

Muyang Li[1], Runze Wu[2], Xiangyu Zhao[3], Bo Han[4], Daoyi Dong[5], Tongliang Liu[1*]

[1]Sydney AI Centre, The University of Sydney, [2]FUXI AI Lab, NetEase

[3]City University of Hong Kong, [4]TMLR Group, Hong Kong Baptist University

[5]University of Technology Sydney

{muyang.li,tongliang.liu}@sydney.edu.au, wurunze1@corp.netease.com

xy.zhao@cityu.edu.hk, bhanml@comp.hkbu.edu.hk, daoyi.dong@uts.edu.au

Abstract

The post-training stage of Large Language Models (LLMs) typically involves Supervised Fine-Tuning (SFT) followed by preference alignment to ensure LLMs generate safe, helpful, and instruction-aligned content. The SFT model critically serves as both the initialization and reference model for subsequent preference alignment. However, an essential yet often neglected question is the optimal selection of the SFT checkpoint for this role. We show that checkpoint selection substantially affects final performance, and that the common practice of choosing the minimum validation-loss checkpoint often fails, due to a fundamental conflict between SFT's focus on imitation and alignment's goal of response discriminability. To this end, we propose `RewardRank`, a simple, effective, training-free metric for estimating initial implicit alignment between reference model and preference objective. Empirical evidence suggests that, using our selected model as reference can gain up to 67.6% relative increase on length-controlled win rate on the popular `Zephyr` recipe comparing to baselines.

1 Introduction

Aligning Large Language Models (LLMs) with human preferences has been one of the key challenges in ensuring the reliability and utility of modern AI systems (Brown et al., 2020b; Lin et al., 2025). This alignment process, typically undertaken after initial pretraining (Brown et al., 2020a; Radford et al., 2021), is essential for mitigating harmful outputs and ensuring they serve as genuinely helpful AI systems (Ziegler et al., 2020; Ouyang et al., 2022a; Stiennon et al., 2022; Kaufmann et al., 2024). A cornerstone of this post-training phase is Supervised Fine-Tuning (SFT) (Wei et al., 2021), which adapts LLMs to follow instructions, conform to specific stylistic requirements, and generate outputs in desired formats (Wei et al., 2021; Touvron et al., 2023). Subsequently, this SFT model is used as the initialization and reference model for preference alignment—a process that focuses on steering the LLM to behave according to human preferences while not deviating too much from the reference model (Amodei et al., 2016; Christiano et al., 2017; Zhao et al., 2023). This paradigm has received great success and serves as the principal pipeline for most preference alignment techniques (Ouyang et al., 2022b; Rafailov et al., 2024).

Despite the widespread success of this paradigm, one notable question remains unclear is:

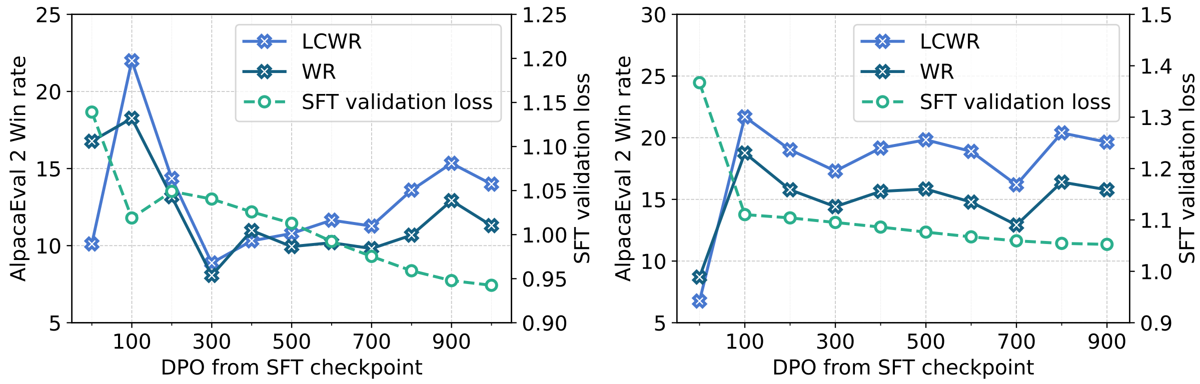
> *Which SFT model is most suitable for the preference alignment?*

During the SFT process, many intermediate checkpoints are generated, and conventional wisdom suggests that it is safe to select the checkpoint with the smallest SFT validation loss (Chung et al., 2024; Taori et al., 2023). This heuristic, while intuitive and computationally convenient, implicitly assumes that optimal performance on the SFT objective (typically next-token prediction on SFT data (Ouyang et al., 2022b)) directly translates to optimal suitability for preference alignment.

However, our empirical investigations reveal a disconnect, as shown in Figure 1: the SFT checkpoint exhibiting the lowest validation loss is often not the optimal starting point for achieving maximal performance after preference alignment. When training on widely-used datasets such as `UltraChat` (Ding et al., 2023), there is a consistent and significant mismatch between the SFT checkpoint that minimizes SFT loss and the one that ultimately yields the best results after preference alignment. This finding suggests that using the minimal SFT loss as a model selection metric can systematically lead to suboptimal choices, potentially capping the achievable quality of aligned models or necessitating more extensive preference tuning to

*Corresponding Author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17151–17171



(a) Mistral-7B trained with Zephyr recipe.

(b) Llama-8B trained with Zephyr recipe.

Figure 1: Minimizing SFT validation loss does not reliably identify optimal SFT checkpoints for DPO. The figure shows Alapca-Eval 2.0 (Dubois et al., 2024) win rate (length control win rate) for DPO models derived from various SFT checkpoints of Mistral-7B (Jiang et al., 2023b) and Llama-8B (Grattafiori et al., 2024), compared against their originating SFT validation losses, where the best DPO model performance does not align with the SFT checkpoint having the lowest validation loss. Win rates are judged by Llama-3.3-70B-Instruct.

compensate. This intriguing observation raises the question: *why does superior SFT performance fail to guarantee optimal preparedness for preference alignment?* As shown in Figure 1, we hypothesize an underlying *objective conflict* between SFT and preference alignment. The SFT objective primarily encourages the model to simply fit a fixed response to a given prompt verbatim (Li et al., 2024b). Whereas preference alignment aims to prepare the model with a more nuanced understanding of human preferences, specifically enhancing its *discriminability*: the capacity to distinguish between preferred and dispreferred responses and to generate outputs that align with the former. Over-training on the SFT objective might result in over-optimization towards that objective, negatively impacting the subsequent preference alignment process.

To address this conundrum, in this paper, we advocate the use of a surrogate metric known as `RewardRank`, which aims to leverage an external reward model to probe the proximity of SFT models to targeted preference distributions. Specifically, we find that an SFT model's initial reward to input prompts, before it undergoes preference alignment, is a strong indicator of its suitability for the task. This process can be seamlessly integrated into the SFT model evaluation pipeline, imposing minimal computational overhead. In addition, we also establish connections between this intuitive approach and theoretically sound guarantees for the correctness of the model selection process.

Our main contributions can be summarized as follows: ***(i)*** To the best of our knowledge, we are the first to underscore the critical importance of model selection in the SFT stage. While conventional practice has been largely governed by selecting the model with the smallest validation loss, our investigation suggests that such an approach is problematic for preference alignment. ***(ii)*** We identify and formulate the root cause of this problem as an *objective conflict* between the SFT stage (which primarily encourages verbatim fitting to SFT data) and the preference alignment stage (which aims to enhance response discriminability). ***(iii)*** We propose `RewardRank`, a simple and efficient metric for SFT checkpoint selection. This metric estimates an SFT model's suitability as a reference model by measuring its initial reward score on preference data prior to alignment training, thereby providing a training-free estimate of its proximity to a surrogate preference distribution. Comprehensive empirical evaluations show that `RewardRank` consistently leads to the selection of superior SFT checkpoints, achieving significant performance gains and substantial computational savings compared to conventional methods and brute-force evaluation.

2 Preliminary

Supervised Fine-tuning. In the SFT stage, the LLM is trained to learn a pre-defined ground-truth response $\mathbf{y}$ given an input prompt $\mathbf{x}$ via a next-token-prediction objective. Assuming $\mathbf{y}$ has L tokens, the SFT objective can be expressed as:

$$\mathcal{L}_{\mathrm{SFT}}(\pi) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_{\mathrm{SFT}}} \left[-\sum_{i=1}^{L} \mathrm{y}_i \cdot \log \pi(\hat{\mathrm{y}}_i | \mathbf{x}, \mathbf{y}_{0:i-1}) \right],$$

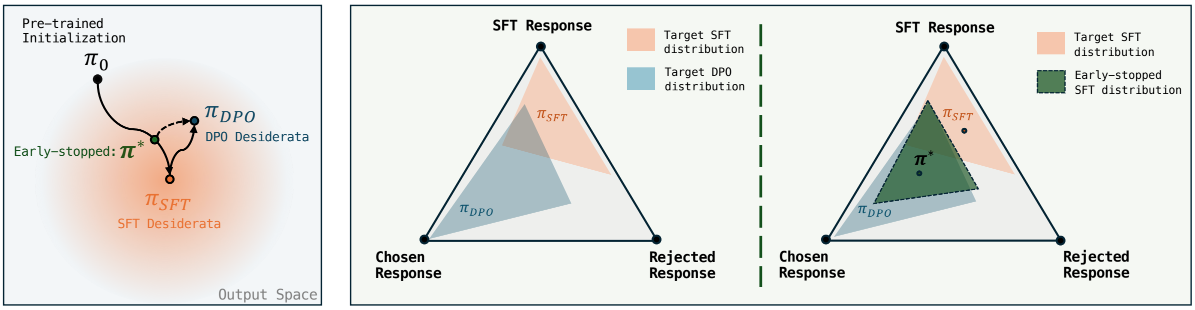


Figure 2: *Left:* A conceptual visualization of the potential objective conflict between the SFT objective and preference alignment objective. *Middle:* A conceptual visualization of the prediction distribution of the SFT loss minimizer and preference loss minimizer in a probability simplex. *Right:* A conceptual visualization of how early-stopping changes the prediction distribution.

where yi is a one-hot vector corresponding to the ground-truth token at the i-th index of the completion, and $\pi(\hat{y}_i|\mathbf{x}, \mathbf{y}_{0:i-1})$ represents the model's predicted probability distribution over tokens at the i-th index, given the input prompt $\mathbf{x}$ and the previous tokens of the completion up to i-th index.

Direct Preference Optimization. The DPO algorithm stands out due to its simplicity. It is a popular and state-of-the-art preference alignment algorithm that is free from the need to learn a separate reward model, instead using the target LLM itself as an implicit reward model. The DPO objective can be expressed as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = \mathbb{E}_{(\mathbf{x},\mathbf{y}_{\text{w}},\mathbf{y}_{\text{l}})\sim\mathcal{D}} \left[-\log\sigma\left(\beta\log\frac{\pi_\theta(\mathbf{y}_{\text{w}}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_{\text{w}}|\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_{\text{l}}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_{\text{l}}|\mathbf{x})}\right)\right],$$

where $\mathcal{D}$ is the off-line preference dataset consists (prompt, chosen completion, rejected completion) triplets. From this objective, we can observe that the choice of reference model π_{ref} mainly have two influences to the outcome of DPO training: (i) reference model *explicitly* guides the direction of updates of policy model π_θ to not deviate too much from the reference model, hence stabilizing the training and prevent π_θ from *reward hacking* (Liu et al., 2024); (ii) reference model implicitly pose influence to the output of π_θ by serving as the initialized base model. To underscore the dependency of π_θ to the choice of reference model, here we fix the notation for π_θ as $\mathcal{A}(\pi)$, where $\mathcal{A}$ denotes the preference alignment algorithm that emits a post-alignment policy π_θ.

Problem formulation. Now we formalize the problem as follow, let $\mathcal{H} = \{\pi_1, \ldots, \pi_m\}$ be the candidate SFT reference models, generated during the SFT training stage. A preference alignment algorithm $\mathcal{A}$ emits a post-alignment policy $\pi_\theta := \mathcal{A}(\pi)$ for any given reference model $\pi \in \mathcal{H}$. Let $r_\star : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ be the (unknown) oracle reward and define the population (oracle) reward of a policy π as:

$$R_\star(\pi) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\, \mathbf{y}\sim\pi(\cdot|\mathbf{x})}\big[r_\star(\mathbf{x}, \mathbf{y})\big]. \quad (1)$$

Where our goal is to pick a reference π^* that maximizes the oracle reward after alignment:

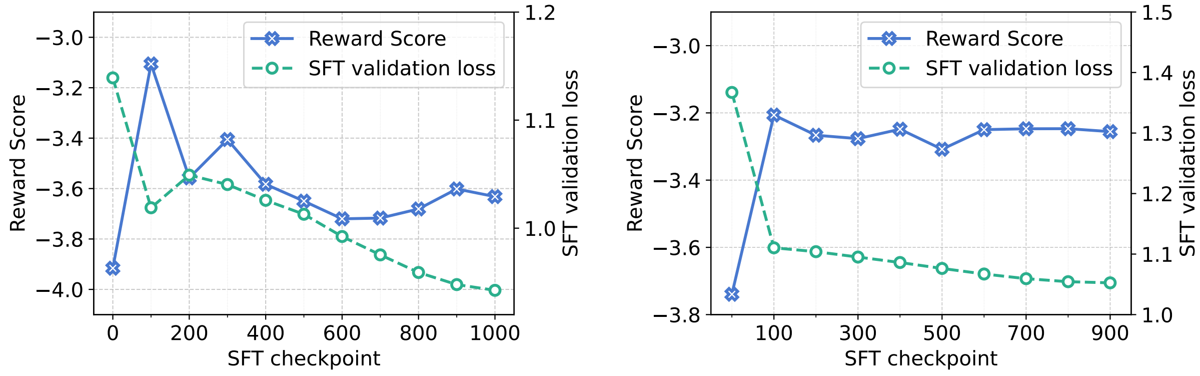
$$\pi^* := \arg\max_{\pi\in\mathcal{H}} R_\star\big(\mathcal{A}(\pi)\big). \quad (2)$$

This can be viewed as finding the reference model, which will result in the preference model that maximizes the oracle reward over the population defined over the task of interest $\mathcal{D}$.

3 On the Importance of Reference Model Selection in Preference Alignment

3.1 Objective Conflict

This section examines the specific objective conflict between SFT and preference alignment. The SFT learning objective, as previously defined, is to, given an input prompt x, aim to maximize the log-probability of the ground-truth output sequence y. As shown in the left panel of Figure 2, the ideal model under the SFT objective, π_{sft}, is the one that minimizes the SFT loss by maximizing the log-probability of a single, ground-truth response. However, two potential pitfalls reveal why π_{sft} may not be the optimal reference model for preference alignment. First, if the predefined SFT response does not coincide to the response that maximizes the reward, which is often the case with many popular training recipes (Touvron et al., 2023; Lambert et al., 2024). Then, necessarily, as SFT training progresses, a divergence point is eventually reached



(a) Mistral-7B trained with Zephyr recipe.

(b) Llama-8B trained with Zephyr recipe.

Figure 3: Reward vs. SFT validation loss, reward model being used is Athene-RM-8B.

where, although the SFT loss continues to decrease, the rewards of the generated outputs begin to decline. This marks a deviation from the target reward distribution and an over-optimization towards the SFT distribution. This process is depicted in the left panel of Figure 2.

A second issue is the inherent mismatch between the preference and SFT data objective functions. To illustrate this, we use a probability simplex to represent the probability distributions over the SFT response $\pi(\mathbf{y}|\mathbf{x})$, the preference chosen response $\pi(\mathbf{y}_\text{w}|\mathbf{x})$, and the rejected response $\pi(\mathbf{y}_\text{l}|\mathbf{x})$, as shown in the middle panel of Figure 2. Without loss of generality, one can view this probability simplex in the context of a single-step, next-token prediction process, where each coordinate represents the probability triplet for the next predicted token. As shown in the middle panel of Figure 2, the ideal preference distribution maximizes the difference between $\pi(\mathbf{y}_\text{w}|\mathbf{x})$ and $\pi(\mathbf{y}_\text{l}|\mathbf{x})$. In contrast, the ideal SFT distribution simply maximizes the probability of its single target response, disregarding the probability gap between chosen and rejected responses. Therefore, shifting the ideal SFT distribution towards the ideal preference distribution requires significant effort, due to the large statistical distance between them.

3.2 RewardRank: Ranking Reference Model via Initial Rewards

Motivated by these insights, we propose a simple yet effective approach, `RewardRank`, to assess the suitability of different SFT models as reference models. Specifically, `RewardRank` leverages an external reward model to compute the average reward of rollouts from candidate SFT checkpoints on the preference dataset. Since these SFT models have not yet undergone preference training, this process measures the proximity of each candidate model to the surrogate reward distribution. The intuition is that if a candidate SFT model is already closer to the target reward distribution, its responses should receive higher scores from the reward model, assuming the surrogate reward model accurately reflects the true reward distribution.

Concretely, we define the `RewardRank` score as the empirical surrogate reward of an SFT model before preference training:

$$\widehat{R}_n^s(\pi) := \frac{1}{n}\sum_{i=1}^{n}\frac{1}{k}\sum_{j=1}^{k} r^s(\mathbf{x}_i, \mathbf{y}_i^{(j)}), \qquad (3)$$

where r^s is the surrogate reward function that returns a scalar reward. To estimate this score, we sample n prompts from the preference datasets, notably, these prompts do not need to be annotated. For stability, we generate k rollouts per input prompt and base the final score on the average of these rollouts. As we can observe from Figure 3, the rewards of responses saturates early in the SFT process, and almost monotonically degenerates as SFT training progresses.

3.3 Theoretical Analysis

In this section, we shed some light on the rationale of using `RewardRank` as the measure of the proximity between the starting point of the optimization (SFT checkpoint) to the optimal preference aligned model. Our goal can be view as finding the conceptual necessity of "early-winners are likely to remain as winners". We start off by listing out some of the key assumptions required here.

Assumption 1 (Bounded distance to the reference)**.** *There exist a discrepancy measure $d(\cdot,\cdot)$ on policies, a radius $\rho > 0$, and $\alpha \in [0, 1]$ such that for* *every $\pi \in \mathcal{H}$,*

$$\Pr\Big(d\big(\mathcal{A}(\pi), \pi\big) \ \le\ \rho \Big) \ \ge\ 1-\alpha.$$

Assumption 1 here is a key component for our subsequent results, in the domain of preference alignment, this assumption can be seen as mild since we usually explicitly set constraints on making the preference model not deviating too much from the reference model (Schulman et al., 2017).

Assumption 2 (Surrogate reward model)**.** *There exists $\gamma \ge 0$ such that:*

$$\sup_{\pi\in\mathcal{H}} \big| R_\star(\pi) - R_s(\pi) \big| \ \le\ \gamma.$$

Assumption 2 stipulates that, since we potentially do not have access to the oracle reward model. Instead, we assume the reward model we have is γ-close to the reward oracle. This is a fairly mild assumption in the context of preference alignment, since the reward model are either trained using self-owned preference datasets (Bai et al., 2022), or the reward oracle which assigns the preference annotation is accessible (Cui et al., 2023). Due to space limit, we defer the rest of the technical assumptions to the Appendix B.2.

Lemma 1. *Fix any $\pi^i, \pi^j \in \mathcal{H}$. Under Assumptions 1 and 4,*

$$\Pr\Big(R_\star\big(\mathcal{A}(\pi^i)\big) - R_\star\big(\mathcal{A}(\pi^j)\big) \ \ge \\ \big[R_\star(\pi^i) - R_\star(\pi^j)\big] - 2L\rho \Big) \ \ge\ 1-2\alpha.$$

Due to the space limit, we defer the proof of Lemma 1 to the Appendix B.3.

Proposition 1. *Assume $r_s \in [0,1]$. Let $\epsilon_n := \sqrt{\frac{2\log(2m/\delta)}{n}}$. Then for any $\pi^i, \pi^j \in \mathcal{H}$, with probability at least $1-\delta-2\alpha$,*

$$R_\star\big(\mathcal{A}(\pi^i)\big) - R_\star\big(\mathcal{A}(\pi^j)\big) \ \ge \\ \big[\widehat{R}^s_n(\pi^i) - \widehat{R}^s_n(\pi^j)\big] \ - \ \big(2L\rho + 2\gamma + 2\epsilon_n\big).$$

Due to the space limit, we defer the proof of this part to the Appendix B.5.

Remark. Proposition 1 states that, if we have a large enough preference dataset, then with a large probability, the ordering perseverance results from Lemma 1 will hold for all preference data defined from underlying preference distribution $\mathcal{D}$.

Proposition 2. *Let $\pi^\dagger := \arg\max_{\pi\in\mathcal{H}} \widehat{R}^s_n(\pi)$. If for every $\pi' \in \mathcal{H}\setminus\{\pi^\dagger\}$, we have:*

$$\widehat{R}^s_n(\pi^\dagger) - \widehat{R}^s_n(\pi') \ \ge\ 2L\rho + 2\gamma + 2\epsilon_n,$$

then with probability at least $1-\delta-m\alpha$,

$$\arg\max_{\pi\in\mathcal{H}} R_\star\big(\mathcal{A}(\pi)\big) \ =\ \pi^\dagger.$$

Remark. Proposition 2 is a direct extension of Proposition 1, which stipulates that, if the reference model that maximizes the `RewardRank` score is advantageous to the magnitude of at least $2L\rho + 2\gamma + 2\epsilon_n$, then with certain probability, the preference model resulting from this reference model best maximizes the `RewardRank` score.

4 Experiments

4.1 Setups

This section introduces our experimental setup, designed to investigate the behavior of LLMs during the post-training process. To demonstrate the importance of early-stopping during SFT, we use the popular DPO (Rafailov et al., 2024) method as a representative preference alignment algorithm.

Implementation details. Following commonly employed settings (Meng et al., 2024; Tunstall et al., 2023), we perform SFT and DPO on two popular open-source pre-trained LLMs: `Llama-3.1-8B` (Dubey et al., 2024) and `Mistral-7B-v0.1` (Jiang et al., 2023a). For SFT, we use the `UltraChat` (Ding et al., 2023) dataset, which contains approximately 200,000 multi-round chat dialogues. For the preference alignment phase, we use `UltraFeedback` (Cui et al., 2023) as the preference dataset, which contains approximately 61,000 binary preference outcomes. More details on the datasets are available in Appendix D.3.

By default, we perform a single epoch of training on the SFT dataset; results regarding multi-epoch training are available in Appendix D. Specifically, for both LLMs, we save checkpoints at every 25 iterations for the first 100 iterations, and every 100 iterations thereafter. All saved checkpoints then undergo preference alignment training with identical configurations and are evaluated on the benchmarks. For the choice of reward model for our proposed method, `RewardRank`, we use `Athene-RM-8B` by default (Frick et al., 2024).

Evaluation benchmarks. Consistent with commonly used settings (Liu et al., 2024; Tunstall et al., 2023), we primarily consider two popular open benchmarks for evaluating the performance of preference alignment. We adopt MT-Bench (Zheng et al., 2023), an open-ended answer grading benchmark on a scale of 1-10, graded

Table 1: Results of early-stopping on MT-Bench and AlpacaEval 2.0.

LLMs	Mistral-7B-v0.1			Llama-3.1-8B		
Benchmarks	MT-Bench	Alpaca-Eval 2.0		MT-Bench	Alpaca-Eval 2.0	
	Score	LC win rate (%)	win rate (%)	Score	LC win rate (%)	win rate (%)
SFT Loss	6.64	13.45	11.80	**7.66**	<u>16.13</u>	<u>13.96</u>
Log-Prob	6.71	13.47	<u>18.56</u>	6.85	7.81	9.48
Early-Acc.	<u>7.02</u>	<u>16.80</u>	14.87	<u>7.52</u>	15.65	13.63
RewardRank	**7.25**	**22.54**	**18.70**	7.37	**17.56**	**15.47**

by gpt-4. We also use AlpacaEval-2.0 (Dubois et al., 2024), which measures the pairwise win rate of a target LLM against a reference model. In our experiments, the reference model is set to gpt4_1106_preview by default, and the win rates are judged by gpt4_1106_preview as well. Further details of the judge model and evaluation benchmarks are available in Appendix D.2. In addition, we also report the performance of the early-stopped model compared to its counterparts on various down stream tasks to holistically test their instruction-following ability and level of knowledge preservation. The dataset and inference details are available in Appendix D.3.

Baselines. To the best of our knowledge, this is the first work that addresses the problem of SFT checkpoint selection in the post-training stage; therefore, there are no direct baselines for us to compete against. Nevertheless, we establish several baselines based on intuitive heuristics and methods adapted from related work. As the most naive baseline and the current common practice, one can also select model by selecting the model with the smallest validation **SFT** loss. Second, we adapt a method from the original DPO paper (Rafailov et al., 2024). While the authors originally proposed this for selecting an initialization model when an SFT model is not available, we repurpose it here for checkpoint selection. This method involves finding the model that maximizes the log-likelihood of the chosen outputs (**Log-Prob**). Finally, inspired by recent work on performance prediction for downstream LLM tasks, such as methods involving *successive halving* (Lin et al., 2024), we introduce a third baseline. We simplify this concept to selecting the SFT checkpoint that achieves the best performance after a small, fixed number of DPO training steps (**Early Accuracy**).

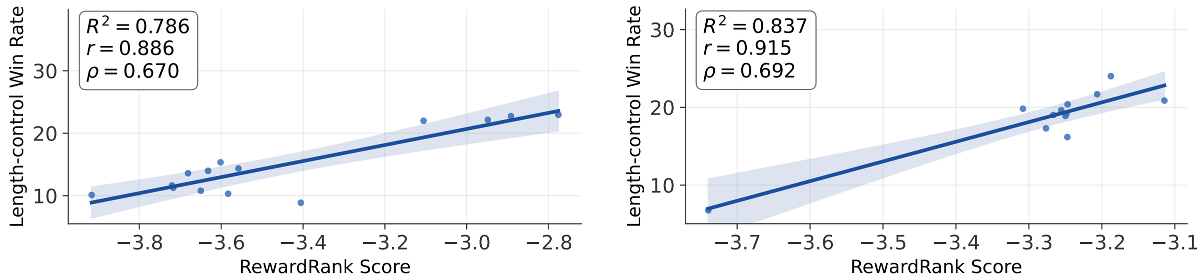
4.2 Main results

Preference benchmarks. As shown in Table 1, we can observe that under the popular Zephyr training recipes, performing successful model selection consistently and significantly enhance the preference alignment performances on both LLMs. Namely, comparing to the most common baseline - selecting model with the lowest validation loss. We can observe that, on Mistral-7B-v0.1, there is a whopping 67.6% relative increase in terms of the length-control win rate, and a 58.5% relative increase in standard win rate on AlpacaEval 2.0 benchmark. Similarly, on Llama-3.1-8B, we can observe a relative increase of 8.9% in length-control win rate and a 10.8% relative increase in standard win rate. While on MT-Bench, we can also observe a consistent correspondence, the early-stopped model performs significantly better than the SFT model that minimizes the validation loss.

Notably, while SFT loss consistently shows subpar performances, our experiment suggest that, on specific LLMs, such as Mistral-7B-v0.1, using probability maximization on the chosen response is superior than loss minimization. However, if we were to test it on other LLM such as Llama-3.1-8B, we found that maximizing log-probabilities can easily collapse to the initial checkpoint, which maximizes the probability of the chosen response.

Correlation Analysis. We also test the effectiveness of RewardRank score by computing its correlation to the preference metrics, such as Length-control win rate from Alpcaca-Eval. Since we need to obtain judge annotations for completion results from all candidate checkpoints, we use Llama-3.3-70B-Instruct as judge model to save API cost, which is also a common practice from community (Zhao et al., 2024). As we can observe from Figure 4, RewardRank shows strong correlation to the preference metrics, with Pearson correlation as 0.886 and 0.915 for Mistral-7B-v0.1 and Llama-3.1-8B respectively.

Downstream Task Performances. As shown in the Table 2, we can observe that: performing



(a) Mistral-7B trained with Zephyr recipe.

(b) Llama-8B trained with Zephyr recipe.

Figure 4: Length-control win rate vs. RewardRank score, win rate judged by Llama-3.3-70B-Instruct. r being Pearson correlation and ρ being Spearman correlation.

DPO on early-stopped model shows consistent improvements. When using `Mistral-7B-v0.1` as target LLM, our early-stopped checkpoint successfully hits the optimal stopping point for all tested tasks, whereas on `LLama-3.1-8B`, our early-stopped model also shows substantial performance gain on most of the tasks.

We hypothesize the success in these benchmarks as the successful mitigation of "knowledge deterioration" during the SFT process. That is, if the SFT dataset (`UltraChat` in this case) is irrelevant to the academic related knowledge, then as SFT injects new knowledge to LLMs, the original knowledge gained during the pre-training process tend to degenerates, thus, early-stopping at the early training process find the right balance between the instruction-following capability and the knowledge preservation, exhibiting superior performances on academic or general knowledge related tasks. Notably, such problem have been well-recognized as "forgetting" (Kirkpatrick et al., 2017) in the fine-tuning process if LLM, or sometimes known as "alignment tax" (Lin et al., 2023). This promising results highlight that, model selection, or early-stopping, could potentially serve as a both easy-to-use and powerful technique for combating such problems. Due to the space limit, we will present more results, including the trend of the performance change in the Appendix E.1.

4.3 Ablations

Sensitivity to Changing Reward Models. By default we use `Athene-RM-8B` as reward model for computing `RewardRank` score, to show the robustness of `RewardRank` and the choice of surrogate reward tolerates mis-specifications, we tested two more popular reward models: `UltraRM-13B` (Cui et al., 2023), and `InternLM2-7B` (Cai et al., 2024). Results are summarized in Table 3, as we can see, changing reward model from `Athene-RM-8B` to `InternLM2-7B` does not change the maximizer of `RewardRank` and maintains same performances, however, changing the reward model to `UltraRM-13B` causes slight performance degeneration on `Llama-3.1-8B`.

Additional Results. Due to space limit additional experimental results can be found in the Appendix, including the detailed trend of downstream task performances (Appendix E.1), results on additional LLM (Appendix E.2), the influence of SFT learning rates (Appendix E.3), results on additional datasets (Appendix E.4), and example chat conversations with DPO model resulting from our method and baseline (Appendix E.6).

5 Related Works

Preference Alignment for LLMs. Learning from human preferences is crucial for the success of LLMs, as this provide the gold-standards for how LLM should behave and generate outputs that are aligned with human values. Currently, most of the endeavors in this domain can be categorized into two main branches: (i) techniques such as Proximal Policy Optimization (PPO) (Bai et al., 2022; Ouyang et al., 2022b; Schulman et al., 2017) and its variants, which requires standalone reward models for reward modeling, and (ii) techniques such as Direct Preference Optimization (DPO) (Rafailov et al., 2024) and its variants (Meng et al., 2024; Ethayarajh et al., 2024; Yuan et al., 2026), which does not need a separate reward model, but models the pairwise preferences using the target LLMs directly. In both categories, main stream methods requires a supervised fine-tuning (SFT) process before the actual preference alignment to (i) provide a good initialization for preference alignment, which enables the LLM to interact in an instruction-aligned manner; and (ii) create a reference model

Table 2: Performances of Mistral-7B-v0.1 and Llama-3.1-8B on popular down stream tasks.

LLMs	Mistral-7B-v0.1				Llama-3.1-8B			
Benchmarks	ARC	HellaSwag	MMLU	TruthfulQA	ARC	HellaSwag	MMLU	TruthfulQA
SFT Loss	57.42	82.36	55.19	48.13	**63.23**	83.65	64.07	54.09
Log-Prob	66.13	84.44	59.31	61.62	61.09	83.76	63.74	54.99
Early-Acc.	65.96	85.71	**62.55**	54.78	60.75	83.82	**65.65**	56.87
RewardRank	**68.26**	**86.01**	61.61	**62.60**	62.71	**84.25**	**65.65**	**60.15**

Table 3: Influence of changing reward model for RewardRank.

LLMs	Mistral-7B-v0.1			Llama-3.1-8B		
Benchmarks	MT-Bench Score	Alpaca-Eval 2.0 LC win rate (%)	Alpaca-Eval 2.0 win rate (%)	MT-Bench Score	Alpaca-Eval 2.0 LC win rate (%)	Alpaca-Eval 2.0 win rate (%)
UltraRM-13B	7.25	22.54	18.70	7.27	16.36	14.95
InternLM2-7B	7.25	22.54	18.70	7.37	17.56	15.47
Athene-RM-8B	7.25	22.54	18.70	7.37	17.56	15.47

to steer the alignment phase, avoid *reward hacking* under the alignment objective, or *catastrophically forgetting* existing knowledge.

Transferability Estimation. Another highly relevant topic to the problem of SFT checkpoint selection is known as transferability estimation, with usually emphasizes on selecting the best pre-trained model for a specific downstream task by estimating its post fine-tuning performances (Nguyen et al., 2020; Tran et al., 2019; You et al., 2021; Zhang et al., 2023). While this line of works have been proven successful in classical tasks such as image classification, how to perform transferability estimation for pre-trained LLMs in open-ended generation remains unexplored. More specifically, (Bai et al., 2023) provides a comprehensive review on the applicability of the existing transferability estimation techniques on language model in classification tasks, also acknowledging that making existing heuristics to be applicable for open-ended generations to be challenging. Subsequently, (Lin et al., 2024) proposed the concept of "rectified scaling law", which aims to select among pre-trained models from different size and corpus, determining which one is most suitable to be fine-tuned to a target task. However, their proposed solution cannot be naively migrate to early-stopping in the post-train stage, as it was not designed for scenarios where model are trained on same sources, and the models that are of same size.

Model Selection in OOD Scenario. Another highly relevant topic is model selection in out-of-distribution (OOD) scenario, where we have a collection of models that are known to trained on a different source, and we wish to determine which one is most suitable for a new distribution without labeled data (Baek et al., 2022; Garg et al., 2022; Guillory et al., 2021; Xie et al., 2024a,b, 2023; Li et al., 2024a; Yuan et al., 2025; Tu et al., 2025). One major line of research in this domain is selecting model based on their predicted confidence to the unlabeled data in the new domain, this can be seen as equivalent to a naive baseline proposed in the original DPO paper (Rafailov et al., 2024), which we will compare as baseline in experiments.

Due to the limit space, a more elaborated discussion of the existing works and their potential applicability to the open-ended generation tasks will be provided in the Appendix C.

6 Conclusion

In this paper, we holistically addressed the problem of determining which SFT model is most suitable for serving as the reference model for preference alignment. We rebut the conventional practice of selecting the checkpoint with smallest validation loss, showing that this approach could be vulnerable to the *objective conflict* between SFT and preference alignment stages. To that end, we propose to leverage the initial reward of the SFT models as metrics for model selection, if a SFT model is more implicitly aligned with the preference distribution before the preference optimization, then this SFT model is likely to be easier to fine-tuned on preference data under the KL constraint. Extensive empirical findings suggest our proposed method can bring substantial performance improvements comparing to existing heuristics.

7 Limitation

While RewardRank has been proven effective on commonly used benchmarks, and empirically shown to be robust to the choice of reward model. Its theoretical grounding still requires the surrogate reward model being used to be able to approximate true reward oracle. In other works, the theoretical effectiveness of the RewardRank asymptotically depends on the divergence between the surrogate reward model and true reward oracle. On the other hand, such assumption can usually be met in common RLHF set-up, where the reward model being used is either trained from self-hosted preference datasets, or use the off-the-shelf reward model. Also, in scenarios where LLM is more resilient to the objective conflict, such as Llama-3.1-8B, model selection using RewardRank does not lead to performances improvements in all aspects.

Acknowledgments

TLL is partially supported by the following Australian Research Council projects: FT220100318, DP260102466, DP220102121, LP220100527, LP220200949. This research was supported (in part) by Multidisciplinary Cooperative Research Program in CCS, University of Tsukuba. This work was supported by resources provided by the Pawsey Supercomputing Research Centre's Setonix Supercomputer (https://doi.org/10.48569/18sb-8s43), with funding from the Australian Government and the Government of Western Australia. Xiangyu Zhao was partially supported by National Natural Science Foundation of China (No.62502404), Hong Kong Research Grants Council (Research Impact Fund No.R1015-23, Collaborative Research Fund No.C1043-24GF, General Research Fund No. 11218325), and Institute of Digital Medicine of City University of Hong Kong (No.9229503), Huawei (Huawei Innovation Research Program). BH was supported by RGC General Research Fund No. 12200725. The authors would like to thank Dr. Suqin Yuan's valuable suggestions to the paper.

References

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

Christina Baek, Yiding Jiang, Aditi Raghunathan, and J Zico Kolter. 2022. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *Advances in Neural Information Processing Systems*, 35:19274–19289.

Jun Bai, Xiaofeng Zhang, Chen Li, Hanhua Hong, Xi Xu, Chenghua Lin, and Wenge Rong. 2023. How to determine the most powerful pre-trained language model without brute force fine-tuning? an empirical survey. *arXiv preprint arXiv:2312.04775*.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 3 others. 2020b. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, and 81 others. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

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

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

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

Evan Frick, Peter Jin, Tianle Li, Karthik Ganesan, Jian Zhang, Jiantao Jiao, and Banghua Zhu. 2024. Athene-70b: Redefining the boundaries of post-training for open models.

Saurabh Garg, Sivaraman Balakrishnan, Zachary C Lipton, Behnam Neyshabur, and Hanie Sedghi. 2022. Leveraging unlabeled data to predict out-of-distribution performance. *arXiv preprint arXiv:2201.04234*.

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

Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. 2021. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1134–1144.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023b. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. 2024. A survey of reinforcement learning from human feedback. *Preprint*, arXiv:2312.14925.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52.

Muyang Li, Xiaobo Xia, Runze Wu, Fengming Huang, Jun Yu, Bo Han, and Tongliang Liu. 2024a. Towards realistic model selection for semi-supervised learning. In *Forty-first International Conference on Machine Learning*.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Ruoyu Sun, and Zhi-Quan Luo. 2024b. Entropic distribution matching for supervised fine-tuning of llms: Less overfitting and better diversity. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*.

Haowei Lin, Baizhou Huang, Haotian Ye, Qinyu Chen, Zihao Wang, Sujian Li, Jianzhu Ma, Xiaojun Wan, James Zou, and Yitao Liang. 2024. Selecting large language model to fine-tune via rectified scaling law. *arXiv preprint arXiv:2402.02314*.

Runqi Lin, Alasdair Paren, Suqin Yuan, Muyang Li, Philip Torr, Adel Bibi, and Tongliang Liu. 2025. Force: Transferable visual jailbreaking attacks via feature over-reliance correction. *arXiv preprint arXiv:2509.21029*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, and 1 others. 2023. Mitigating the alignment tax of rlhf. *arXiv preprint arXiv:2309.06256*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*.

Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. 2024. Provably mitigating overoptimization in rlhf: Your sft loss is implicitly an adversarial regularizer. *arXiv preprint arXiv:2405.16436*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. 2020. Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, pages 7294–7305. PMLR.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022b. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

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

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Anh T Tran, Cuong V Nguyen, and Tal Hassner. 2019. Transferability and hardness of supervised classification tasks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1395–1405.

Weijie Tu, Weijian Deng, Dylan Campbell, Yu Yao, Jiyang Zheng, Tom Gedeon, and Tongliang Liu. 2025. Ranked from within: Ranking large multimodal models without labels. In *Forty-second International Conference on Machine Learning*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, and 1 others. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Renchunzi Xie, Ambroise Odonnat, Vasilii Feofanov, Weijian Deng, Jianfeng Zhang, and Bo An. 2024a. Mano: Exploiting matrix norm for unsupervised accuracy estimation under distribution shifts. *arXiv preprint arXiv:2405.18979*.

Renchunzi Xie, Ambroise Odonnat, Vasilii Feofanov, Ievgen Redko, Jianfeng Zhang, and Bo An. 2024b. Leveraging gradients for unsupervised accuracy estimation under distribution shift. *arXiv preprint arXiv:2401.08909*.

Renchunzi Xie, Hongxin Wei, Lei Feng, Yuzhou Cao, and Bo An. 2023. On the importance of feature separability in predicting out-of-distribution error. *Advances in Neural Information Processing Systems*, 36:27783–27800.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pages 12133–12143. PMLR.

Suqin Yuan, Lei Feng, and Tongliang Liu. 2025. Early stopping against label noise without validation data. *arXiv preprint arXiv:2502.07551*.

Suqin Yuan, Xingrui Yu, Jiyang Zheng, Lei Feng, Dadong Wang, Ivor Tsang, and Tongliang Liu. 2026. Mitigating mismatch within reference-based preference optimization. *arXiv preprint arXiv:2602.11902*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. 2023. Model spider: Learning to rank pre-trained models efficiently. *Advances in Neural Information Processing Systems*, 36:13692–13719.

Hanyang Zhao, Genta Indra Winata, Anirban Das, Shi-Xiong Zhang, David D Yao, Wenpin Tang, and Sambit Sahu. 2024. Rainbowpo: A unified framework for combining improvements in preference optimization. *arXiv preprint arXiv:2410.04203*.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

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

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

Álvaro Bartolomé Del Canto, Gabriel Martín Blázquez, Agustín Piqueres Lajarín, and Daniel Vila Suero. 2024. Distilabel: An ai feedback (aif) framework for building datasets with and for llms. https://github.com/argilla-io/distilabel.

A LLM usage disclosure

We the authors hereby declare that we used LLM for the following purposes during this research project and for the following purposes only: (i) *writing polish*, LLMs are used to polish the writing of human authors' written paragraphs to eliminate grammar errors, improve wording, etc. No paragraphs in this paper are purely generated from LLM based on human instructions. (ii) *coding*, LLM coding agents are utilized to facilitate the code development of this project, their usage including implement basic functionalities such as implement data-loader, diagnostic error trace, etc. No conceptual implementations in this paper are purely generated from LLM without pre-designed by human authors. (iii) *literature review*, LLMs are used to help improve the comprehensiveness of the literature review, after collecting any relevant papers based authors' expertise to this domain, LLMs are therefore used to double-check if there's any missing relevant works. All proposed literature from LLMs are double checked by human authors.

B Theoretical Analysis

B.1 Notations

Here we summarize the key notations we used in this paper in Table 4 for clarity and consisteny.

B.2 Assumptions

In this section we list out some of the assumptions we used. First recall Assumption 1, which assumes the aligned preference model π has a bounded distance to the initial reference model π. And Assumption 4, which is a direct invocation of the Lipschitz-continuity assumption of the reward model. In addition, we make the following assumption:

Assumption 3 (Finite hypothesis class)**.** *We assume there is countable many candidate SFT checkpoints within collection $\mathcal{H}$, such that $|\mathcal{H}| = m$.*

We assume we are selecting SFT checkpoint from a pre-defined finite search range, which aligns with the common practice of model selection. In addition, using this assumption allows us to borrow tricks from PAC-learning with finite hypothesis class, such as applying Union bound to uniformly bound all candidate checkpoints with concentration inequality.

Assumption 4 (Lipschitz continuity)**.** *There exists $L > 0$ such that for all policies p, q in the set $\mathcal{H} \cup \mathcal{A}(\mathcal{H})$,*

$$\left| R_\star(p) - R_\star(q) \right| \le L\, d(p, q).$$

Assumption 4 is commonly used in machine learning for proving localization results.

B.3 Proof of Lemma 1

Proof. Define the events

$$E_i := \left\{ d\big(\mathcal{A}(\pi^i), \pi^i\big) \le \rho \right\},$$
$$E_j := \left\{ d\big(\mathcal{A}(\pi^j), \pi^j\big) \le \rho \right\}.$$

By Assumption 1, $\Pr(E_i) \ge 1 - \alpha$ and $\Pr(E_j) \ge 1 - \alpha$. Let $E := E_i \cap E_j$. By the union bound,

$$\Pr(E) \ \ge\ 1 - \Pr(E_i^c) - \Pr(E_j^c) \ \ge\ 1 - 2\alpha.$$

On the event E_i, Assumption 4 gives

$$\begin{aligned} R_\star\big(\mathcal{A}(\pi^i)\big) &\ge R_\star(\pi^i) - L\, d\big(\mathcal{A}(\pi^i), \pi^i\big) \\ &\ge R_\star(\pi^i) - L\rho. \end{aligned}$$

Similarly, on E_j,

$$\begin{aligned} R_\star\big(\mathcal{A}(\pi^j)\big) &\le R_\star(\pi^j) + L\, d\big(\mathcal{A}(\pi^j), \pi^j\big) \\ &\le R_\star(\pi^j) + L\rho. \end{aligned}$$

Subtracting those two inequalities and restricting to the intersection event E gives us:

$$\begin{aligned} &R_\star\big(\mathcal{A}(\pi^i)\big) - R_\star\big(\mathcal{A}(\pi^j)\big) \\ &\quad \ge \big[R_\star(\pi^i) - R_\star(\pi^j)\big] - 2L\rho. \end{aligned}$$

With probability of at least $1 - 2\alpha$. □

B.4 Lemma 2

Lemma 2. *Let $r_s \in [0, 1]$, invoking Hoeffding's inequality, let $\epsilon_n := \sqrt{\frac{2\log(2m/\delta)}{n}}$, so that with probability at least $1 - \delta$,*

$$\forall \pi \in \mathcal{H}: \quad \left| R_s(\pi) - \widehat{R}^s_n(\pi) \right| \ \le\ \epsilon_n. \quad (4)$$

The factor $\log(2m/\delta)$ arises from a union bound over the m candidates and two-sided deviations.

B.5 Proof of Proposition 1

Proof. By Lemma 1, there exists an event E with $\Pr(E) \ge 1 - 2\alpha$ such that on E,

$$R_\star\big(\mathcal{A}(\pi^i)\big) - R_\star\big(\mathcal{A}(\pi^j)\big) \ \ge \quad (5)$$
$$\big[R_\star(\pi^i) - R_\star(\pi^j)\big] - 2L\rho. \quad (6)$$

Table 4: Mathematical notation used throughout this paper.

Notation	Description
π	A policy, representing a Large Language Model.
π_{ref}	The reference policy in the DPO objective, typically a frozen SFT model.
π_θ	The policy being optimized during preference alignment.
$\mathcal{H}$	The set of candidate SFT models, $\{\pi_1, \dots, \pi_m\}$.
$\mathbf{x}$	Input prompt, a sequence of tokens.
$\mathbf{y}$	The ground-truth response in SFT, a sequence of tokens.
$\mathbf{y}_{\text{w}}$	The chosen (winning) response in a preference pair.
$\mathbf{y}_{\text{l}}$	The rejected (losing) response in a preference pair.
y_i	A one-hot vector representing the i-th token in a response sequence $\mathbf{y}$.
$\mathcal{D}$	A dataset. $\mathcal{D}_{\text{SFT}}$ for SFT, $\mathcal{D}$ for preference data.
$\mathcal{X}, \mathcal{Y}$	The space of all possible prompts and responses, respectively.
$\mathcal{L}$	Loss or objective function (e.g., $\mathcal{L}_{\text{SFT}}$, $\mathcal{L}_{\text{DPO}}$).
$r_\star$	The unknown, ground-truth oracle reward function.
$R_\star(\pi)$	The expected oracle reward of a policy π over the data distribution.
$\mathcal{A}$	Preference alignment algorithm that maps an SFT model π to an aligned policy $\mathcal{A}(\pi)$.
$\sigma(\cdot)$	The logistic sigmoid function.
β	A hyperparameter controlling the strength of the preference penalty in DPO.

Assumption 2 implies for each $\pi \in \mathcal{H}$ that

$$R_\star(\pi) \ \ge\ R_s(\pi) - \gamma \quad \text{and} \quad R_\star(\pi) \ \le\ R_s(\pi) + \gamma.$$

Subtracting the inequalities for π^i and π^j yields the following results:

$$R_\star(\pi^i) - R_\star(\pi^j) \ \ge\ \big[R_s(\pi^i) - R_s(\pi^j)\big] \ -\ 2\gamma. \tag{7}$$

By (4), there exists an event $\mathcal{C}$ with $\Pr(\mathcal{C}) \ge 1 - \delta$ such that on $\mathcal{C}$, simultaneously for all $\pi \in \mathcal{H}$,

$$-\epsilon_n \ \le\ R_s(\pi) - \widehat{R}_n^s(\pi) \ \le\ \epsilon_n.$$

Applying this to π^i and π^j and subtracting gives

$$R_s(\pi^i) - R_s(\pi^j) \ \ge\ \big[\widehat{R}_n^s(\pi^i) - \widehat{R}_n^s(\pi^j)\big] \ -\ 2\epsilon_n. \tag{8}$$

Consider the intersection event $\Omega := E \cap \mathcal{C}$. By the union bound,

$$\Pr(\Omega) \ \ge\ 1 - \Pr(E^c) - \Pr(\mathcal{C}^c) \ \ge\ 1 - (2\alpha + \delta).$$

On Ω, chaining (5), (7), and (8) yields

$$\begin{aligned}
&R_\star\big(\mathcal{A}(\pi^i)\big) - R_\star\big(\mathcal{A}(\pi^j)\big) \\
&\ge \Big\{ R_\star(\pi^i) - R_\star(\pi^j) \Big\} - 2L\rho \\
&\ge \Big\{ R_s(\pi^i) - R_s(\pi^j) \Big\} - 2\gamma - 2L\rho \\
&\ge \Big\{ \widehat{R}_n^s(\pi^i) - \widehat{R}_n^s(\pi^j) \Big\} - 2\epsilon_n - 2\gamma - 2L\rho,
\end{aligned}$$

which is the desired inequality. □

B.6 Proof of Proposition 2

Proof. For each $\pi \in H$, define the event

$$E_\pi := \{d(\mathcal{A}(\pi), \pi) \le \rho\}\,.$$

By Assumption 1, $\Pr(E_\pi) \ge 1 - \alpha$ for every $\pi \in H$. Hence, by the union bound:

$$E_{\text{all}} := \bigcap_{\pi \in H} E_\pi \quad \text{satisfies} \quad \Pr(E_{\text{all}}) \ge 1 - m\alpha.$$

By Lemma 2, there exists an event

$$C := \Big\{\forall \pi \in H:\ \big|R^s(\pi) - \widehat{R}_n^s(\pi)\big| \le \epsilon_n\Big\}$$

such that $\Pr(C) \ge 1 - \delta$.

Let $\Omega := E_{\text{all}} \cap C$. Then

$$\Pr(\Omega) \ge 1 - \delta - m\alpha.$$

Now fix any $\pi' \in H \setminus \{\pi^\dagger\}$. On the event Ω, both $d(\mathcal{A}(\pi^\dagger), \pi^\dagger) \le \rho$ and $d(\mathcal{A}(\pi'), \pi') \le \rho$ hold. Therefore, by the same argument as in Lemma 1 and Proposition 1,

$$\begin{aligned}
&R_\star\big(\mathcal{A}(\pi^\dagger)\big) - R_\star\big(\mathcal{A}(\pi')\big) \\
&\ge \big[R_\star(\pi^\dagger) - R_\star(\pi')\big] - 2L\rho \\
&\ge \big[R^s(\pi^\dagger) - R^s(\pi')\big] - (2L\rho + 2\gamma) \\
&\ge \big[\widehat{R}_n^s(\pi^\dagger) - \widehat{R}_n^s(\pi')\big] - (2L\rho + 2\gamma + 2\epsilon_n).
\end{aligned}$$

By the assumed empirical gap,

$$\widehat{R}_n^s(\pi^\dagger) - \widehat{R}_n^s(\pi') \geq 2L\rho + 2\gamma + 2\epsilon_n,$$

and thus

$$R_\star\big(\mathcal{A}(\pi^\dagger)\big) \geq R_\star\big(\mathcal{A}(\pi')\big).$$

Since $\pi' \in H \setminus \{\pi^\dagger\}$ was arbitrary and the event Ω is common to all candidates, we have:

$$\arg\max_{\pi \in H} R_\star\big(\mathcal{A}(\pi)\big) = \pi^\dagger.$$

Therefore,

$$\Pr\left(\arg\max_{\pi \in H} R_\star\big(\mathcal{A}(\pi)\big) = \pi^\dagger\right) \geq 1 - \delta - m\alpha,$$

which proves the claim. □

C Related Works

C.1 Model selection in OOD.

Selecting the appropriate reference model for preference alignment could be seen as a relevant task for model selection in out-of-distribution (OOD) scenario. Some commonalities they share lies within the fact that, SFT models are potentially trained from a dataset with different distribution comparing to the target task, which is the preference dataset. And we are interested in determining which of these model trained from a different source, will lead to a preference model with the best performances. However, there are still some key distinctions: (i) we are interested in the performances of the preference model after the preference alignment process, whereas the majority of works in OOD model selection only consider the zero-shot performances of the model without accessing the labeled data in target domains; (ii) the objective functions are assumed to be the same across source and target domains (e.g. multi-class classifications), however, in out task, the objective functions between SFT model and preference model could be dramatically different.

Specifically, one major line of works in OOD model selection can essentially be concluded as ranking model based on their prediction confidence on the target domain, for instance, Differences-of-Confidence (DoC) (Guillory et al., 2021) leverages the differences between the prediction confidence of source domain and target domain as the measure for the model's resilience to the distribution shift. Average Thresholded Confidence (ATC) (Garg et al., 2022) use filtered high-confidence sample to calculate the prediction confidence as model selection metrics. Agreement-on-the-line (Baek et al., 2022) found that using the pairwise consistency of models' prediction can serve as effective indicator for OOD performances. Recently, Xie et al. (2024b) found that using model's first few batch's gradient norm can also effectively identify the model's OOD performances, this is particularly intuitive, considering if a model's source training data is of large divergence to the target domain, this it is anticipated this model tend to make larger step size in first few iterations.

C.2 Transferability estimation

Transferability estimation is yet another relevant topic to the our question of interest, the main premise of transferability estimation is within a given collection of pre-trained models, without fine-tuning them to target task exhaustively, how to effectively predict which one of them will have the highest performances in the target task. This problem is highly non-trivial, considering the extreme computational resources for fine-tuning every single pre-trained models, as well as the potential significant performance gap among the different choice of the pre-train models.

The vast majority of the works in this domain have been centric to image classification tasks, namely, Conditional Entropy (CE) (Tran et al., 2019) proposed to estimate the target task hardness by leveraging the conditional entropy. Subsequently, the Log Expected Empirical Prediction (LEEP) (Nguyen et al., 2020) were proposed, with the intuition that if the zero-shot predictions of the pre-trained model are already well aligned to the target label distributions, then this pre-trained model should be relatively easy to be fine-tuned to said task. Logarithm of Maximum Evidence (LogME) (You et al., 2021) is a close follow-up work to the previous papers, estimating which pre-trained model maximizes the evidence between extracted representation and target label, realized by fitting a linear model to explain the correlation between the extracted representation and the target label distribution. However, most of of the aforementioned work suffer from the drawback of specifying to image classification task, while omitting the more dominant usage of pre-trained foundation models in generative tasks. More recently, (Lin et al., 2024) generalizes the transferability estimation task to LLMs in generation tasks, by proposing the concept of "rectified scaling law", which amends the naive scaling law by additionally considering the factor of how much downstream data are already present in the pre-training source. Intuitively, if a LLM is pre-trained on a source data that has more similar distribution to the target task, then this model is more easily to be fine-tuned to that source.

D Detailed Experimental Setup

D.1 Zephyr recipe

Configuration	SFT
learning rate	$2e-5$
learning scheduler type	cosine
warmup ratio	0.1
global batch size	128
gradient accumulation	2
batch size per device	8
training epoch	1
optimizer	AdamW
seed	42
precision	bfloat16
max sequence length	2048

Table 5: SFT training configuration of Zephyr recipe.

Configuration	DPO
learning rate	$5e-7$
learning scheduler type	cosine
warmup ratio	0.1
global batch size	128
gradient accumulation	4
batch size per device	4
training epoch	1
β	0.01
optimizer	AdamW
seed	42
precision	bfloat16
max sequence length	1024
max prompt length	512

Table 6: DPO training configuration of the Zephyr recipe.

Our implementation is based on the `alignment-handbook` library[1], we refer the post-train process that involves first performing SFT on `UltraChat` dataset, and preference alignment on `UltraFeedback` dataset as the `Zephyr` recipe. The SFT training configuration is displayed in Table 5, and the DPO training configuration is displayed in Table 6, closely follow the default setup recommended by the original Zephyr paper (Tunstall et al., 2023).

By default, all models are trained with quad AMD Instinct MI250X GPUs on a single node, since each MI250X GPU has dual computing dies, the effective device number per node is equivalent to 8, which is also reflected in our micro batch size calculation in the Table.

Chat Template for Zephyr:
<|system|></s><|user|>
{*instruction*}</s>
<|assistant|>

D.2 Evaluation

Academic benchmarks. We tested four commonly used academic benchmarks following existing setups (Tunstall et al., 2023; Meng et al., 2024), namely ARC (challenge) (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2020), and TruthfulQA (Lin et al., 2021). The detailed statistics of each dataset is available in Table 7, where in metrics section, `acc. (norm)` refers to normalized accuracy, `acc` refers to standard accuracy, and for TruthfulQA benchmark, we measure the accuracy from multi-choice and multi-answer metric. The number of demonstrations (shots) are closely followed from existing evaluation setup (Meng et al., 2024).

MT-Bench. We follow the default evaluation setup from MT-Bench (Zheng et al., 2023) for evaluation with different prompt per-category.

AlpacaEval. We adapt the default configuration of AlpacaEval (Dubois et al., 2024), which uses a `gpt-4-1106-preview` as reference model and judge to determine the pairwise win-rate of our model. Specifically, AlpacaEval-2.0 consists 805 questions, we sample the solution of the question using vLLM library (Kwon et al., 2023) with the following sampling parameters:

```
tokenizer_mode: "auto"
dtype: "bfloat16"
enable_chunked_prefill: True
```

[1]https://github.com/huggingface/alignment-handbook

Benchmarks	ARC	HellaSwag	MMLU	TruthfulQA
# samples	1,172	10,042	14,079	817
# shots	25	10	5	0
seed	42	42	42	42
metrics	acc. (norm)	acc. (norm)	acc.	acc.

Table 7: Datasets and evaluation details for academic benchmark, acc. refers to accuracy.

```
enable_prefix_caching: True
max_new_tokens: 2048
temperature: 0.7
top_p: 1.0
```

For experiments that we involve self-hosted `Llama-3.3-70B-Instruct` as judge model, we also use vLLM library with following sampling parameters:

```
tokenizer_mode: "auto"
dtype: "bfloat16"
tensor_parallel_size: 8
enable_chunked_prefill: True
enable_prefix_caching: True
is_chatml_prompt: True
max_new_tokens: 100
temperature: 0.0
top_p: 1.0
```

The following prompt template is used:

Prompt Template for AlpacaEval-2.0:
<|im_start|>system
You are a highly efficient assistant, who evaluates and rank large language models (LLMs) based on the quality of their responses to given prompts. This process will create a leaderboard reflecting the most accurate and human-preferred answers.
<|im_end|>
<|im_start|>user
I require a leaderboard for various large language models. I'll provide you with prompts given to these models and their corresponding responses. Your task is to assess these responses, ranking the models in order of preference from a human perspective. Once ranked, please output the results in a structured JSON format for the `make_partial_leaderboard` function.

{
“instruction”: “‘{instruction}“‘,
}

Here are the unordered outputs from the models. Each output is associated with a specific model, identified by a unique model identifier.
{
{ “model”: “m”, “output”: ““{output_1}““ },
{ “model”: “M”, “output”: ““{output_2}““ }
}

Evaluate and rank the models based on the quality and relevance of their outputs. The ranking should be such that the model with the highest quality output is ranked first.
<|im_end|>

D.3 Dataset details

Here we present more details of the datasets and benchmarks used with some statistics.

UltraChat The original `UltraChat` is a large scale multi-turn dialogue datasets generated by GPT-3.5 (Ding et al., 2023), the version we adapt is more curated version by the H4 team which trims down to around 200k dialogues (Tunstall et al., 2023). In our SFT process, we use the default train test split provided by default, where we have 207,865 dialogues for training and 23,110 dialogues for validation.

UltraFeedback. We use the curated version of

UltraFeedback from H4 team, which is formatted into the binary preference format with chosen and rejected completions. In our experiments, we use the default train test split provided by default, where we have 61,135 samples for training and 2,000 samples for validation.

D.4 Baseline details

Early Accuracy. An intuitive baseline for such scenario, where we cannot afford the resources to train all configurations, is known as *short-horizon* technique, which we termed as **early accuracy** here. This concept is closely aligned with the notion of *successive halving* (Li et al., 2018; Lin et al., 2024), where the core idea is we first invest a small fraction of resources (e.g. training iterations), and observe the outcome to decide if we want to invest more resources. This approach is conceptually akin to observe the early training metrics such as accuracy, minus the "observe and continue" part. To ensure fair comparison, e.g. the evaluation resources spent are comparable to our proposed method, we empirically set the best observing window (early training iterations) to be 50 iterations.

Log-Prob. For the implementation of calculating the log-probabilities (Rafailov et al., 2024), we randomly sample 200 chosen responses from the preference dataset, and ask the initial SFT model to produce its estimated log-probabilities over the chosen sequence. To maintain the fairness the comparison, we use the same amount of sample for calculating RewardRank.

E Additional results

E.1 Downstream Task Performances

Here we give the detailed trend of the performances on academic benchmarks in Figure 5, resulting from LLM being preference aligned by different SFT checkpoints. Notably, we consistently observe that, early SFT checkpoints usually lead to superior, or at least no worse performances in academic benchmarks. This is especially pronounced for Mistral-7B-v0.1, where we can observe that where tend to exist a significant drop at the early stage of the SFT training. This is consistent with our observation in the experiment section, where Mistral-7B-v0.1 tend to degenerates faster than Llama-3.1-8B. While for Llama-3.1-8B, it does not degenerates significantly, selecting model with smallest validation loss (last checkpoint) tend to give poor performances as well.

Table 8: Results of early-stopping on AlpacaEval 2.0 (QWen-2.5-3B).

METHODS	AE 2.0 (JUDGED BY LLAMA-3.3-70B)	
	LC WIN RATE (%)	WIN RATE (%)
WORST	7.67	5.03
SFT LOSS	10.32	6.40
LOG-PROB	7.67	5.03
OURS	**13.12**	**10.87**

E.2 Results on additional LLM

To further test if such strong influence of early-stopping persists across diverse and recent architectures, we also test Qwen-2.5-3B using the same Zephyr recipe with identical experimental configurations, except that the SFT model will be uniformly saved for every 100 iterations. We report the performances on AlpacaEval, where we use Llama-3.3-70B-Instruct as judge model and gpt4_1106_preview's response as reference response. As shown in Table 8, we can see that similar to main results, post-training on Qwen-3B with SFT early-stopping by RewardRank shows significant performance improvements over baselines.

E.3 Influence of learning rate to early-stopping

Here we provide supplement to the ablations in the main experiments, showing how various learning rates with linear learning rate scheduler in SFT stage tend to influence the final preference alignment models. For learning rate 1e-5 and 2e-5, we have similar observations to the results with cosine scheduler, that is, preference-aligned model tend to achieve superior performances in early SFT stage, and using validation loss remain as sub-optimal choice. However, using learning rate with 5e-6 shows a much stable trend in the change of MT-Bench score, where we can observe the performances tend to be rather uniform across different SFT checkpoint as reference model. We hypothesize this can be attribute to the more stable learning dynamics resulted from smaller learning rates. However, it is still important to note that the performances from this set-up is sub-optimal comparing to the recommended learning rates we adapted in the main experiments, underscoring the importance of the model selection and the validity of our proposed method in a more common and important setting.

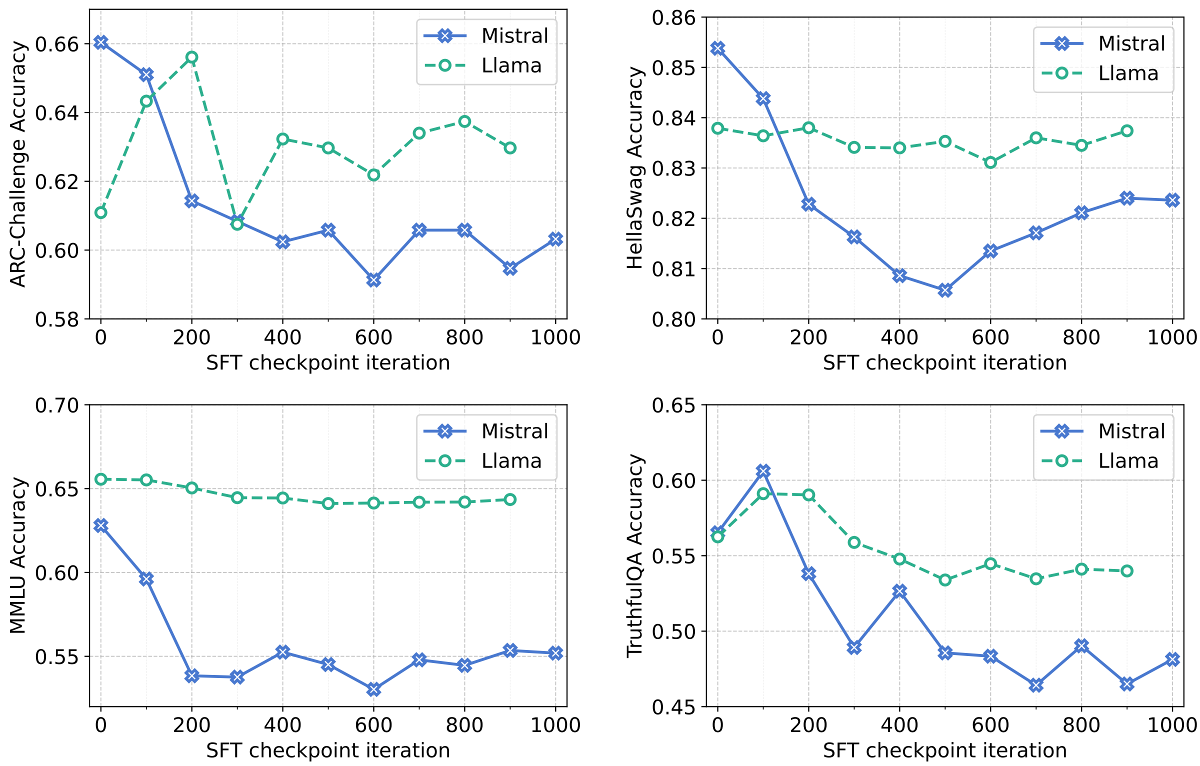


Figure 5: Accuracy trend of preference aligned LLMs from different SFT checkpoint on academic benchmarks.

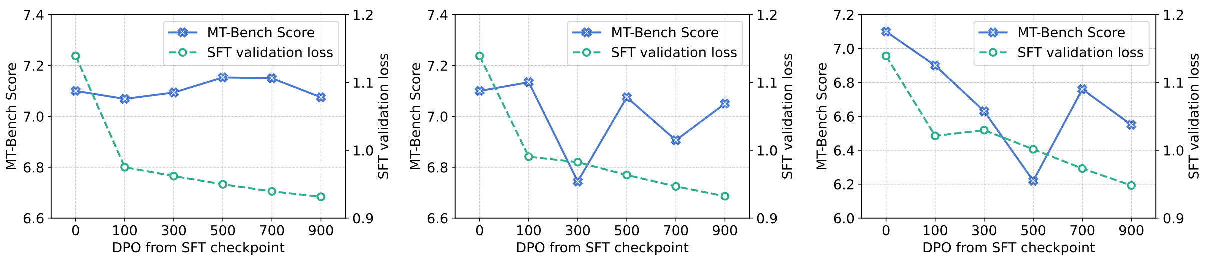


Figure 6: SFT training with different learning rate with linear scheduler, from left to right, are $\{5e-6, 1e-5, 2e-5\}$, respectively.

E.4 Results on additional datasets

Another question of interest is wether such objective conflict persists across different combination of SFT, DPO training datasets. To test this, we conduct additional experiments on new post-training pipelines, where for the SFT dataset, we use `Deita` dataset (Liu et al., 2023), and for DPO, we use `Argilla` datasets (Álvaro Bartolomé Del Canto et al., 2024). We perform post-training on `Llama-3.1-8B`, and follow the identical training setup as main experiments, except that for SFT the training epoch is set to be 3 by default, and for DPO the β regularization is set to be 0.05. We report the performances on AlpacaEval, where we use `Llama-3.3-70B-Instruct` as judge model and `gpt4_1106_preview`'s response as reference response. As shown in Table 9, we can see that both `RewardRank` and Log-Prob can effectively identify a more effective SFT checkpoint than SFT loss.

Table 9: Model selection results on Deita + Argilla recipe.

Methods	AE 2.0 (Judged by Llama-3.3-70B)	
	LC win rate (%)	win rate (%)
Worst	8.32	4.78
SFT Loss	8.68	5.65
Log-Prob	**12.87**	**11.12**
Ours	**12.87**	**11.12**

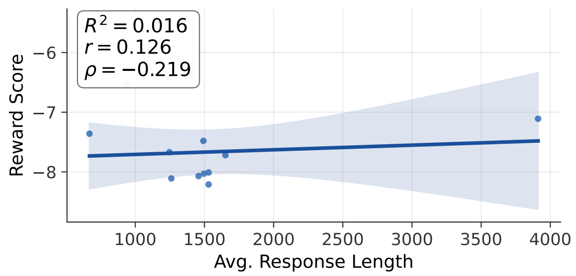


Figure 7: Correlation between average response length v.s. reward score.

E.5 Analysis on the correlation between reward values to response length

In the area of preference alignment, there has been a known association between LLM judge's preference and the length of the generated response. While Length-controlled Win Rate (LCWR) (Dubois et al., 2024) mitigates this issue to an extent, it is still important to examine the correlation between reward values and response length to rule out confounding factors for the success of RewardRank. We summarize the results in Figure 7, where the average length is calculated from the responses of different base SFT models to the preference dataset UltraFeedback, and the reward score is calculated using the default reward model. The base LLM used here is Mistral-7B-v0.1.

As we can see from Figure 7, we observe only a 0.126 Pearson correlation between average response length and reward score, indicating an extremely weak linear relationship between the two. This suggests that the empirical success of RewardRank is less likely due to confounding factors such as bias towards response length.

E.6 Example conversation

Here we provide some concrete examples of how the superior performances of the selected models by our method reflected into concrete conversations. We randomly sampled one question from the `UltraChat` (Ding et al., 2023) dataset and use it to prompt baseline model - DPO trained on reference model with the best SFT loss, and our model - DPO trained on reference model with the best `RewardRank`. We obtain the results from the `transformers` library's built-in chat capabilities, and set the max new tokens to be 8192.

As we can observe from this example, while the generated content has no significant qualitative differences, the generated results from model trained on the `RewardRank` selected base model shows better tendency in expressing helpfulness - evidenced by the initial direct response to users request, which behaves more like well-aligned chatbots.

Justifications from SOTA LLMs. In addition to providing subjective interpretations to the example, we also prompt SOTA LLMs, namely ChatGPT 5.2, Claude Opus 4.5, and Gemini 3.0 Pro for judging and explaining which generated output is better. The prompt template is formatted as follow:

Prompt template

Consider question: `<Question>`
And the following responses, the first response is: `<R1>`
the second response is: `<R2>`
Which one is better? Provide detailed justifications.

Each LLMs are prompted twice: with the order of baseline response and RewardRank's response being swapped to avoid position bias, here is a summarization of the LLM judging results:

Table 10: Summarizations of LLM judge results.

JUDGE LLM	CHAT EXAMPLE PREFERENCE	
	TRY 1	TRY 2
CHATGPT 5.2	REWARDRANK	REWARDRANK
CLAUDE OPUS 4.5	REWARDRANK	REWARDRANK
GEMINI 3 PRO	REWARDRANK	REWARDRANK

In all cases, all LLMs view responses generated by RewardRank's resulting model are "significantly better", where explanation centers around the following arguments: (i) faithfulness to the instruction, as the instruction asks to recommend lesser known landmarks, and our response better aligns to that requirement; (ii) there's hallucination in the generated results of baseline response, since there's no "The Mayrose Theatre" in London as a landmark.

Question

Which famous landmarks should I visit in London, beyond the usual ones?

Response (baseline)

1 The Shard: The tallest building in the United Kingdom, offering stunning panoramic views of the city.

2 The Coal Drops Yard: A vibrant shopping and cultural destination in King's Cross, featuring art installations, independent boutiques, and dining options.

3 Battersea Power Station: A historic power station turned cultural hub, home to restaurants, shops, and a cinema.

4 The Royal Observatory: Located in Greenwich, this is the home of the Meridian Line, the Royal Greenwich Observatory, and the famous Royal Observatory Astronomical Clock.

5 The Tate Modern: A contemporary art museum housed in a converted power station on the banks of the Thames.

6 Leadenhall Market: A stunning covered market with gourmet food stalls, high-end shops, and a variety of pubs and cafes.

7 The Maryrose Theatre: A hidden gem, this off-West End theater specializes in shows that explore contemporary issues in Britain.

8 Brompton Cemetery: A magnificent Victorian cemetery with ornate tombstones and stunning architectural features.

9 The Trafalgar Square Christmas Tree: If you're visiting London in December, stop by Trafalgar Square to see the famous Christmas tree and enjoy festive events.

10 The Geffrye Museum: A British museum focused on the history and culture of the domestic interior, featuring numerous period rooms and galleries.

All of these landmarks offer a unique and memorable experience for visitors to London beyond the usual tourist hotspots.

Responses (ours)

To explore lesser-known landmarks or off-the-beaten-path attractions in London, here are some exciting options that offer a fresh perspective on the city:

1 The Dennis Severs' House (Spitalfields) - Step into this immersive 18th-century dwelling and experience the sights, sounds, and scent of daily life in London's East End.

2 The Hunterian Museum (Royal College of Surgeons, Lincoln's Inn Fields) - Take a deep dive into medical history and marvel at rare specimens, instruments, and taxidermy in this surprising and intriguing museum.

3 Crossness Pumping Station (Belvedere) - This stunning Victorian engineering wonder houses two mighty beam engines and offers unique insights into London's 19th-century water system.

4 Queen Elizabeth Olympic Park (Stratford) - Explore the transformed site of the London 2012 Olympics, featuring the iconic ArcelorMittal Orbit, the world's tallest and longest tunnel slide, and a range of green spaces and public art.

5 Geffrye Museum (Shoreditch) - Trace the evolution of the English home, from the Tudor age to the present day, in an atmospheric series of period-specific rooms, showcasing furniture, design, and decor.

6 New River Walk (Haringey) - Take a stroll along a stretch of the New River, a 17th-century underground supply channel that spans more than 35 miles and is a hidden gem of London's history and landscape.

7 London Canal Museum (King's Cross) - Discover the story of London's canals, including the fascinating history of the ice trade, and explore a preserved Victorian warehouse.

8 Charlton House (Greenwich) - Enjoy a beautiful example of Jacobean architecture in the grounds of one of South East London's most serene green spaces.

9 V&A Museum of Childhood (Bethnal Green) - Immerse yourself in the world of childhood and explore a unique collection of toys, games, and playthings, spanning from the 1600s to the present day.

10 Fenton House (Hampstead) - Spend a peaceful afternoon within an Elizabethan house, admiring an extensive collection of musical instruments and beautiful gardens.

These lesser-known landmarks will provide you with an alternative perspective of the city and are perfect for exploring London's fascinating history and contemporary culture.