UFT: Unifying Fine-Tuning of SFT and RLHF/DPO/UNA through a Generalized Implicit Reward Function

Anonymous ACL submission

Abstract

By pretraining on trillions of tokens, an LLM gains the capability of text generation. However, to enhance its utility and reduce potential harm, SFT and alignment are applied sequentially to the pretrained model. Due to the differing nature and objective functions of SFT and alignment, catastrophic forgetting has become a significant issue. To address this, we introduce Unified Fine-Tuning (UFT), which integrates SFT and alignment into a single training stage using the same objective and loss functions through an implicit reward function.

Our experimental results demonstrate that UFT outperforms SFT on instruction-tuning data alone. Moreover, when combining instruction-tuning data with alignment data, UFT effectively prevents catastrophic forgetting across these two stages and shows a clear advantage over sequentially applying SFT and alignment. This is evident in the significant improvements observed in the **ifeval** task for instruction-following and the **truthful-qa** task for factuality. The proposed general fine-tuning framework UFT establishes an effective and efficient pretraining-UFT paradigm for LLM training.

1 Introduction

To enable large language models (LLMs) to understand and generate natural language, they are constructed with billions of parameters and pretrained on datasets containing trillions of tokens (OpenAI et al., 2024). However, several challenges arise after the pretraining stage of LLMs (Wang et al., 2024b). One major issue is that pretrained LLMs can only continue generation based on the previous context and often struggle to accurately answer user questions. To address this, supervised fine-tuning (SFT) is introduced, using pairs of questions and answers. For example, in models like Mistral, preset instructions such as '[INST]' and '[/INST]' are used to frame a question as a prompt (Jiang et al., 2023). The corresponding answer is then used as the target output. The model's probability of generating the correct answer is maximized through next-token prediction, employing the cross-entropy loss function to classify tokens across the entire token space.

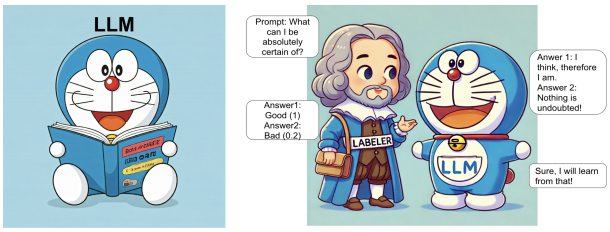

Figure 1: UFT integrates SFT and alignment through a generalized implicit reward function. It likens pretraining and fine-tuning of LLMs to Chinese proveb "Read ten thousand books, travel ten thousand miles". In pre-training, the LLM processes vast amounts of data without feedback, gaining broad language understanding. In fine-tuning, it generates responses to prompts and receives feedback, refining its abilities and improving performance on specific tasks.

The next challenge for LLMs lies in ethical concerns, where LLMs may inadvertently teach humans to engage in unethical activities, such as robbing banks (Ouyang et al., 2022). To address this issue, various alignment methodologies have been proposed, including Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022a) with Proximal Policy Optimization (PPO) (Schulman et al., 2017), Direct Preference Optimization (DPO) (Rafailov et al., 2023), Kahneman & Tversky Optimization (KTO) (Ethayarajh et al., 2024), and UNified Alignment (UNA) (Wang et al., 2024a). The core idea of alignment is to equip LLMs with the ability to reject harmful requests by learning from human feedback.

For RLHF/PPO, a dataset is created consisting of triplets: a prompt, a desired response, and an unde-

sired response. This preference dataset is then used to train a pointwise reward model that evaluates a pair of prompt and response using the Bradley-Terry (BT) model (Bradley and Terry, 1952). During the RL stage of RLHF, a multi-objective optimization is performed to balance maximizing the reward score from the pretrained reward model for a prompt-response pair and minimizing the divergence between the fine-tuned model and the original pretrained model using KL divergence. However, RLHF is known for being unstable and computationally expensive. To address these issues, DPO proposes a method to optimize the reward model and the policy model together, transforming the original two-stage task in RLHF into a single-stage classification process. Although DPO simplifies the training task of RLHF, it only utilizes responses as either desired or undesired, lacking the more nuanced reward scores provided by reward models. KTO extends DPO slightly by replacing preference feedback with binary feedback. Recognizing the limitations that DPO can only handle pairwise feedback and KTO can only handle binary feedback, and they are both significantly different from RLHF/PPO, UNA proposes a unified framework that combines these alignment methods and various types of feedback through a generalized implicit reward function, transforming the unstable RLHF into a stable supervised learning problem.

For pretrained LLMs, SFT and alignment are typically applied in sequential order. This sequential approach often leads to catastrophic forgetting, where the model loses capabilities acquired in earlier stages. To address this, ORPO introduced a new objective, but it is constrained by its reliance on pairwise feedback and its limitation on effectiveness (Hong et al., 2024). On the other hand, PAFT proposed applying SFT and alignment in parallel, followed by merging these adaptors with the pretrained model (Pentyala et al., 2024). However, the PAFT method is cumbersome as it involves three stages: SFT, alignment, and model merging and it requires sparsity of SFT and alignment to avoid interference during merging. This paper aims to tackle the catastrophic forgetting problem by introducing a mathematically proven and efficient methodology, i.e., UFT.

We are inspired by UNA's ability to process score-based feedback effectively. Consequently, we aim to extend UNA to achieve the objectives of SFT. Specifically, SFT involves maximizing the probability of a response given an instruction. Therefore, instruction-tuning data can be considered alignment data with the highest feedback score, such as the desired response for pairwise feedback, a thumbs-up for binary feedback, and a score of 1 for score-based feedback. In the paper, we have demonstrated that UFT and SFT both aim to maximize the likelihood of the responses in instruction-tuning data. By combining the instruction-tuning dataset with the alignment data, we can fine-tune a pretrained LLM with the same objective and loss function while effectively preventing catastrophic forgetting. Our experiments demonstrate that UNA, when applied to instruction-tuning data, can outperform traditional SFT on the same data in downstream tasks. Furthermore, this new perspective allows us to mix the instruction-tuning dataset with the alignment dataset, enabling us to fine-tune LLMs in a single step. Our experiments indicate that this approach prevents the issue of catastrophic forgetting by surpassesing the performance of the previous sequential training pipeline. This is evident in the significant improvements observed in the **ifeval** task for instruction-following (Zhou et al., 2023) and the **truthful-qa** task for factuality (Lin et al., 2022).

Lastly, we have conducted a direct comparison between UFT and the pretraining stage. Drawing from an old Chinese proverb, "Read ten thousand books, travel ten thousand miles", we can liken the training of a LLM to a successful human life as shown in Figure. 1. The pretraining stage corresponds to "Read ten thousand books," where the model is trained on billions of tokens to predict the next token without receiving any feedback. In contrast, the fine-tuning stage is akin to "Travel ten thousand miles," where the LLM is posed with various questions, generates responses, and receives feedback from different labelers and utilize these feedback for improvement. The existing fine-tuning paradigm involves 2-stage training: SFT followed by alignment. Our new UFT unifies the two stages, leading to a post-training framework parallel to pretraining.

The contributions of this paper are listed as follow:

1. Prove that both UNA and SFT maximize the likelihood of the response in instruction-tuning data, and UNA outperforms SFT on downstream tasks when fine-tuning on instruction-tuning data.

2. UFT that unifies SFT and Alignment solves the catastrophic forgetting problem elegantly and surpasses the performance of the original sequential application order in downstream tasks.

3. UFT builds a unified post-training framework that is parallel to pretraining where the goal lies in generating responses for given prompts, receiving score-based feedback from different labelers and improve its capability on downstream tasks.

2 Methodology

In this section, we will explore the methodologies of SFT, RLHF, DPO, UNA, and the integrated framework UFT.

2.1 SFT

The pretrained LLM is limited to either continuing the text or repeating the question, which restricts its usefulness. To address these limitations and enhance the LLM's question-answering capabilities, SFT is applied. The instruction-tuning dataset consists of numerous pairs of prompts (denoted as x) and responses (denoted as y). Given a prompt x, the probability of the pretrained LLM generating the response y is represented as $\pi_\theta(y|x)$. The objective of SFT is to maximize the probability of all response tokens by using cross-entropy loss as shown in Eq. 1 and part (A) of Figure. 2.

$$L_{\text{SFT}}(\pi_\theta) = -\log\left(\pi_\theta(y|x)\right) \tag{1}$$

Suppose y is composed of N tokens, i.e., $y_1, y_2, \ldots, y_N$. Based on Bayes' theorem, $\pi_\theta(y|x) = \prod_{i=1}^{n} \pi_\theta(y_i|x, y_1, \ldots, y_{i-1})$. Applying log to Eq. 1, the SFT loss can be derived based on each token using cross entropy loss function on all candidate tokens for classification, i.e., $L_{\text{SFT}}(\pi_\theta) = -\sum_{i=1}^{n} \log\left(\pi_\theta(y_i|x, y_1, \ldots, y_{i-1})\right)$.

2.2 RLHF, DPO and UNA

Even after the pretraining and SFT stages, LLMs can still produce undesired responses that may lead to bias or ethical issues. To address this, methods such as RLHF, DPO, and KTO have been proposed. A plot on the difference among RLHF, DPO and UNA can be found in part (B) of Figure. 2. RLHF tackles these problems in two stages: reward model training and reinforcement learning. During the reward model training process, an explicit reward model is derived from pairwise data using the BT model, as illustrated in Eq. 2, where r_ϕ represents the explicit reward model. The dataset for training the reward model is composed of triplet of 1. prompt x, 2. desired response y_w and 3. undesired response y_l. The second stage of RLHF involves online reinforcement learning with the pretrained explicit reward model to generate reward signals. These signals are then combined with KL divergence to balance the reward and model capability obtained during the pretraining stage, as shown in Eq. 3. The RL process is optimized using PPO.

$$L_{\text{RM}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\Big[\log\left(\sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right)\Big] \tag{2}$$

$$\pi^*_\theta(y|x) = \max_{\pi_\theta} \mathrm{E}_{x\sim D}\Big[\mathrm{E}_{y\sim\pi_\theta(y|x)}\left[r_\phi(x, y)\right] - \beta D_{\text{KL}}\left(\pi_\theta(y|x) \,\|\, \pi_{\text{ref}}(y|x)\right)\Big] \tag{3}$$

The training of RLHF is memory-intensive because it requires maintaining both the explicit reward model and the policy model. Additionally, reinforcement learning is notorious for its instability. To address this issue, DPO proposes creating a mapping between the optimal policy and the reward model, i.e., an implicit reward model, as illustrated in Eq. 4, and optimizing them together. However, $Z(x)$ is intractable and can only be canceled out by subtracting the implicit reward of the desired response y_w, i.e., $r_\theta(x, y_w)$ from the implicit reward of the undesired response y_l, i.e., $r_\theta(x, y_l)$. This limitation confines DPO to pairwise datasets only. Furthermore, DPO cannot utilize the precise evaluation from the explicit reward model in RLHF. KTO extends DPO to binary feedback by estimating $Z(x)$ from multiple responses to the same prompt.

$$r_\theta(x, y) = \beta \log\left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right) + \beta \log Z(x) \tag{4}$$

To address this issue, UNA offers an alternative proof and demonstrates a novel mapping between the optimal policy and the reward model, i.e., the generalized implicit reward model. The form of the mapping between the generalized implicit reward model and policy is $r_\theta(x, y) =$

$\beta \log \left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) + f(x) + c$ where $f(x)$ measures the quality of the prompts and c adjusts the offset between implicit and explicit rewards. With simplification, i.e., $f(x) = c = 0$, the relationship between the generalized implicit reward model and policy can be shown in Eq. 5. Unlike DPO, which is constrained by $Z(x)$ to pairwise feedback, UNA is versatile enough to handle various types of data, including pairwise feedback, binary feedback, and score-based feedback. These data types are optimized by minimizing the difference between the implicit reward in Eq. 5 and the explicit reward $r_\phi(x, y)$, which is provided by human labelers, other LLMs, or the explicit reward model, as shown in Eq. 6. When LLMs and explicit reward models are used to evaluate responses in real-time, it is referred to as online UNA. Conversely, if feedback is labeled beforehand, it is termed offline UNA. Consequently, UNA can function in both online and offline modes, effectively bridging the gap between online RLHF and offline DPO and KTO.

$$r_\theta(x, y) = \beta \log \left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) \tag{5}$$

$$L_{\text{UNA}}(\pi_\theta) = \mathbb{E}_{x \sim D, y \sim \pi_\theta(\cdot|x)} [g(r_\phi(x, y), r_\theta(x, y))] \tag{6}$$

2.3 UFT Unifying SFT and Alignment

To enhance the capabilities of LLMs in question answering and addressing ethical issues, SFT and alignment are typically applied in a sequential manner due to the distinct nature of these tasks. However, this sequential approach often leads to the well-known issue of catastrophic forgetting. In this study, we introduce UFT which integrates SFT and alignment into a single stage, thereby mitigating this problem.

To be more specific, the instruction-tuning data consist of a prompt x and a corresponding response y, where the response y is considered of high quality, typically labeled by experts in the field. In comparison, the alignment data include a prompt x, response y, and feedback r. This feedback can be categorized as desired or undesired for pairwise feedback, positive or negative for binary feedback, or a scalar in the interval [0, 1] for general score-based feedback. Due to the high quality of instruction-tuning data, they can be regarded as data with a score of 1, i.e., positive feedback. With this consideration, the instruction-tuning dataset can be transformed into alignment data in the format of prompt x, response y and feedback $r = 1$, which is the highest reward in consideration. Eventually, the transformed instruction-tuning dataset and the alignment dataset can be merged for fine-tuning LLM using the loss function in UNA as shown in Eq. 6. Our experiments demonstrate that when fine-tuning using only instruction-tuning dataset, UFT can outperform SFT on downstream tasks, which we attribute to the KL divergence term that focuses on minimization with the pretrained model, a factor ignored in the SFT processes. Additionally, the results indicate that mixing the instruction-tuning data with alignment for UFT can prevent the catastrophic forgetting problem and outperform previous sequential methods. Lastly, we discover that the distribution of mixed data, i.e., the proportion of instruction-tuning data and alignment data will impact the performances of LLMs. More details can be found in the experiment section.

A heuristic proof why UFT can replace SFT is provided by arguing that they achieve the same goal of maximizing the probability of $\pi_\theta(y|x)$ for prompt x and response y in the instruction-tuning dataset. For SFT, the probability of $\pi_\theta(y|x)$ is di-

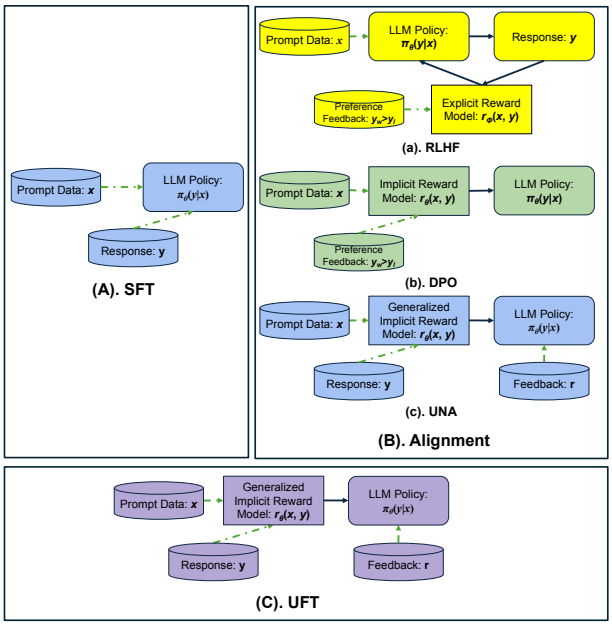

Figure 2: Subfigure (A) refers to SFT, Subfigure (B) refers to Alignment including RLHF, DPO and UNA and Subfigure (C) refers to UFT. Traditionally, the fine-tuning process begins with SFT followed by alignment. However, the proposed UFT method integrates both SFT and alignment into a single, cohesive process.

rectly maximized by cross entropy-loss. In UFT, suppose we apply the Sigmoid function on the implicit reward in Eq. 5 and using mean square error (MSE) to measure the difference of implicit reward and explicit reward, the objective will become Eq. 7. This loss function will drive $r_\theta(x, y) = \beta \log \left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right)$ to positive infinity. Since both β and $\pi_{\text{ref}}(y|x)$ are fixed, this objective will maximize $\pi_\theta(y|x)$. As a result, UFT can replace SFT to maximize the probability of generating the response, and it outperforms SFT by minimizing the difference to the pretrained model.

$$L_{\text{UFT-SFT}}(\pi_\theta) = \mathbb{E}_{(x,y)\sim D}\left[[\sigma(r_\theta(x,y)) - 1]^2 \right]$$
$$= \mathbb{E}_{(x,y)\sim D}\left\{ \left[\sigma\left(\beta \log \left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) \right) - 1 \right]^2 \right\} \quad (7)$$

In summary, UFT reformats the data structure of instruction-tuning data to ensure compatibility with UNA, thereby enabling the UFT framework to unify SFT and alignment processes. An offline version of UFT is illustrated in part (C) of Figure 2. This offline UFT can be transitioned to an online version, provided that the pre-collected feedback is gathered in real-time using a reward model or other LLMs.

3 Experiments

In this section, a series of experiments will be conducted to demonstrate the advantages of UFT across various use cases. Initially, a comparison between UFT and SFT on instruction-tuning data will be performed to illustrate that UFT surpasses SFT. Following this, the second experiment will compare the performance of UFT when applied to SFT and alignment data simultaneously versus applying SFT and alignment sequentially. The third experiment will examine the impact of data distribution between SFT and alignment data, which is crucial for the outcomes of fine-tuning.

3.1 UFT vs. SFT

To begin with, we compare UFT with SFT on the same instruction-tuning dataset. The UltraChat dataset is utilized by unfolding one conversation into multiple training examples to faciliate the training of UFT. From the UltraChat dataset, 20k samples are selected and utilized for training. The Mistral 7B-v0.1 (Jiang et al., 2023) is utilized as the base model, with low rank adaptation (LoRA) of $r = 16$ (Hu et al., 2021). For SFT, the learning rate of $1e^{-4}$ is the best, and for UFT, the learning rate of $3e^{-5}$ and β of 0.01 is the best.

The fine-tuned models are tested on different tasks including 12 tasks on two HuggingFace Open LLM Leaderboards(Beeching et al., 2023; Fourrier et al., 2024). The new Open LLM Leaderboard includes 6 tasks: **bbh** (Suzgun et al., 2022), **gpqa** (Rein et al., 2023), **mmlu-pro** (Wang et al., 2024), **musr** (Sprague et al., 2024), **ifeval** (Zhou et al., 2023), and **math-hard** (Hendrycks et al., 2021). For all these tasks, the average scores are reported. Conversely, the old Open LLM Leaderboard comprises 6 different tasks: **gsm8k** (Cobbe et al., 2021), **truthful-qa** (Lin et al., 2022), **winogrande** (Sakaguchi et al., 2019), **arc** (Allen AI), **hellaswag** (Zellers et al., 2019), and **mmlu** (Hendrycks et al., 2021). In this work, the average match rate in **gsm8k**, mc2 in **truthful-qa**, accuracy in **winogrande**, acc-norm in **arc**, acc-norm in **hellaswag**, and accuracy in **mmlu** will be reported. Additionally, MT-Bench (Zheng et al., 2023) and Alpaca-eval (Li et al., 2023) will be used to evaluate the model's ability to generate text responses, rather than selecting from predefined candidate answers. Specifically, the Length-Controlled Win Rate (LC WR) will be documented for Alpaca-eval. Additionally, the average length of the outputs within this evaluation will be provided for comprehensive analysis.

As demonstrated in Table 1 and Table 2, Mistral+UFT surpasses Mistral+SFT in 8 out of 12 tasks. When considering the average performances across these leaderboards, Mistral+UFT consistently outperforms Mistral+SFT. This indicates that UFT enhances the capabilities of an LLM by maximizing the reward and minimizing the difference from the pretrained model. Furthermore, in the MT-Bench evaluation, Mistral+UFT outperforms Mistral+SFT, whereas in the Alpaca-eval, Mistral+SFT overshadows Mistral+UFT. From a generation capability standpoint, both UFT and SFT exhibit similar performance. As a result, when evaluating the performance on downstream tasks, UFT demonstrates superiority over SFT.

3.2 Catastrophic Forgetting Study

Another advantage of UFT lies in combining SFT and alignment into one stage to avoid catastrophic forgetting. In this stage, we utilize the HelpSteer2 dataset of 20k examples for alignment (Wang et al.,

Model	bbh	gpqa	mmlu-pro	musr	ifeval	math-hard	average
Mistral	44.11	29.53	30.11	41.79	23.22	2.92	28.61
Mistral+SFT	46.04	28.72	29.35	**42.94**	**29.5**	2.66	29.87
Mistral+UFT	**46.55**	**29.24**	**30.25**	41.73	28.89	**3.87**	**30.09**
Mistral+SFT+DPO	44.52	29.98	29.95	40.31	26.64	**3.13**	29.09
Mistral+SFT+KTO	42.89	31	**30.48**	40.59	25.17	2.94	28.85
Mistral+SFT+UNA	43.74	30.78	30.09	40.56	26.82	2.96	29.16
Mistral+UFT	**45.46**	**31.15**	30.05	**41.06**	**46.03**	**3.13**	**32.81**

Table 1: Comparison of Mistral+SFT and Mistral+UFT on instruction-tuning data, and comparison of Mistral+SFT+DPO, Mistral+SFT+KTO, Mistral+SFT+UNA, and Mistral+UFT on both instruction-tuning and alignment data on the new HuggingFace open LLM Leaderboard

Model	gsm8k	truthful-qa	winogrande	arc	hellaswag	mmlu	average
Mistral	38.02	42.58	77.58	61.43	83.44	62.51	60.93
Mistral+SFT	39.65	51.06	78.53	**63.99**	**83.78**	61.99	63.17
Mistral+UFT	**45.57**	**51.18**	**78.93**	63.82	83.54	**62.44**	**64.25**
Mistral+SFT+DPO	42.19	47.83	78.45	62.16	84.03	**62.38**	62.84
Mistral+SFT+KTO	**42.57**	49.67	79.4	61.86	83.83	62.06	63.23
Mistral+SFT+UNA	39.99	49.54	79.72	62.46	84.08	62.3	63.02
Mistral+UFT	41.59	**54.05**	**79.79**	**63.82**	**84.44**	62.33	**64.34**

Table 2: Comparison of Mistral+SFT and Mistral+UFT on instruction-tuning data, and comparison of Mistral+SFT+DPO, Mistral+SFT+KTO, Mistral+SFT+UNA, and Mistral+UFT on both instruction-tuning and alignment data on the old HuggingFace open LLM Leaderboard

Model	MT-Bench	Alpaca-eval	
		LC WR	Length
Mistral	3.15	0.31	6554
Mistral+SFT	6.33	**8.07**	908
Mistral+UFT	**6.55**	7.27	974
Mistral+SFT+DPO	4.81	1.05	5654
Mistral+SFT+KTO	4.76	0.64	6215
Mistral+SFT+UNA	5.24	1.34	4945
Mistral+UFT	**6.78**	**8.28**	1317

Table 3: Comparison of Mistral+SFT and Mistral+UFT on instruction-tuning data, and comparison of Mistral+SFT+DPO, Mistral+SFT+KTO, Mistral+SFT+UNA, and Mistral+UFT on both instruction-tuning and alignment data using MT-Bench and Alpaca-eval

2024c). For UFT, the 20k examples from UltraChat and 20k examples from HelpSteer2 are merged and utilized for training. For comparison, the best performing SFT model of learning rate $1e^{-4}$ in the previous experiment is utilized for further fine-tuning using DPO, KTO and UNA.

The same tasks are utilized for evaluation. In the two HuggingFace open LLM leaderboards, Mistral+UFT outperforms all of Mistral+SFT+DPO, Mistral+SFT+KTO, and Mistral+SFT+UNA in 9 out of 12 tasks. In terms of average scores, Mistral+UFT surpasses all three sequential methods. Several aspects need further discussion. Firstly, the catastrophic forgetting problem is evident as the performances of Mistral+SFT+DPO, Mistral+SFT+KTO, and Mistral+SFT+UNA are worse than Mistral+SFT, a phenomenon also known as alignment tax. Additionally, we observe that Mistral+UFT shows significant improvements on **ifeval**, which tests the model's capability of instruction-following, and **truthful-qa**, which assesses the model's alignment capabilities. These results indicate that UFT greatly enhances instruction-following and alignment capabilities, demonstrating its effectiveness. In terms of generation capability, Mistral+UFT outperforms the other three sequential methods and it does not seem to suffer from catastrophic forgetting, as it performs better than both Mistral+SFT and Mistral+UFT on instruction-tuning data alone. Moreover, Mistral+UFT does not bias towards long generation like the other three sequential methods, which is another advantage of UFT.

3.3 Data Distribution in Training UFT

During the pretraining phase, ensuring a balanced data distribution is crucial for optimizing the capabilities of a LLM. In line with this principle,

we have incorporated varying percentages of the instruction-tuning data into the alignment dataset to create a final training dataset. Specifically, we have kept the alignment dataset constant, i.e., 20k while integrating 260k, 130k, 65k, 32k, and 16k samples from the UltraChat dataset. These combined datasets are then used for fine-tuning, allowing us to observe the impact of different data distributions on the model's performance.

Several intriguing points will be discussed. As shown in Table 4 and Table 5, among the 12 tasks, one task, namely **musr**, shows a minor statistical decrease in performance. In contrast, four tasks-**gpqa**, **gsm8k**, **winogrande**, and **arc**—exhibit statistically small improvements. Notably, two tasks, **ifeval** and **truthful-qa**, demonstrate statistically significant improvements. Specifically, **ifeval** improves from 23.22 to around 44, and **truthful-qa** improves from 42.58 to around 53. Significant advancements are observed in MT-Bench and Alpaca-eval, indicating a substantial enhancement in the generation capability of the LLM as shown in Table 6. This aligns with our expectations, as instruction-tuning data primarily aim to enhance the model's instruction-following capabilities, which positively impacts **ifeval**, MT-Bench, and Alpaca-eval. Meanwhile, alignment data contribute to improvements in tasks like **truthful-qa**. Other tasks remain largely unaffected due to the absence of relevant data, suggesting that incorporating more pertinent data could enhance their performance. When increasing the proportion of instruction-tuningg data, we observe a decrease in **truthful-qa** performance from 56 to 50, indicating that a larger proportion of alignment data benefits the bias and ethical performance of the LLM. On the other hand, simply adding more instruction-tuning data does not improve the performance of **ifeval**, MT-Bench, and Alpaca-eval. Therefore, further investigation into the optimal ratio between instruction-tuning data and alignment data is warranted.

4 Related Work

The field of LLMs has undergone significant advancements, with billions of parameters and trillions of tokens processed in parallel during the pretraining stage (OpenAI et al., 2024; Anthropic, 2024; Team et al., 2023). Following pretraining, SFT is applied to enhance the model's performance on downstream tasks.

However, neither pretraining nor SFT can fully address the issues of bias and ethics in LLMs (OpenAI et al., 2024; Wang et al., 2024b). To tackle these challenges, RLHF with PPO has been proposed and is widely accepted for aligning LLMs, including GPT and Claude (Ouyang et al., 2022; Bai et al., 2022a). Despite its popularity, RLHF/PPO faces several issues, such as high memory requirements, instability in reinforcement learning, and the need for multiple training stages, including reward model (RM) training and RL fine-tuning (Rafailov et al., 2023). To reduce the cost of human labeling, AI feedback can be used to replace human feedback, a method known as reinforcement learning from AI feedback (RLAIF) (Bai et al., 2022b; Lee et al., 2023). RLOO argues that PPO is excessive for LLM alignment since the model has already been pretrained, suggesting that RLOO should suffice (Ahmadian et al., 2024).

To simplify RLHF, DPO has been proposed to map the optimal policy and reward model into a single step, transforming the initial unstable RL into a binary cross-entropy problem (Rafailov et al., 2023). Several downstream studies have been conducted on DPO to expand and generalize its capabilities, such as different loss functions (Pal et al., 2024; Azar et al., 2023), iterative approaches (Yuan et al., 2024; Xu et al., 2024), token DPO (Rafailov et al., 2024; Zeng et al., 2024) and stepwise DPO (Kim et al., 2024).

Previous work focused on pairwise datasets, which are more challenging to gather. In contrast, binary feedback like "thumbs up" and "thumbs down" is easier to collect. KTO leverages the concept of human aversion to undesired data and successfully handles binary feedback (Ethayarajh et al., 2024). DRO focuses on binary data by estimating the policy and value functions and optimizing each sequentially while keeping the other fixed (Richemond et al., 2024). However, these methods cannot accommodate different types of data. To address this, UNA was proposed to handle pairwise, binary, and score-based feedback through an implicit reward model, supporting both online and offline alignment (Wang et al., 2024a).

There have also been efforts to merge SFT with alignment. ORPO proposed a new loss function to increase the ratio of desired responses over undesired responses, achieving both SFT and alignment (Hong et al., 2024). PAFT suggested conducting SFT and alignment in parallel and merging them afterward (Pentyala et al., 2024). However, ORPO's reliance on pairwise datasets and its deteriorating

Model	bbh	gpqa	mmlu-pro	musr	ifeval	math-hard	average
Mistral	44.11	29.53	**30.11**	**41.79**	23.22	2.92	28.61
UNA(16k+20k)	45.17	30.64	29.95	39.19	44.21	2.38	31.92
UNA(20k+20k)	**45.46**	31.15	30.05	41.06	46.03	3.13	**32.81**
UNA(32k+20k)	44.75	31.28	29.76	39.06	**46.76**	**3.51**	32.52
UNA(65k+20k)	44.4	**31.38**	29.66	36.8	41.91	**3.51**	31.28
UNA(130k+20k)	44.35	29.75	30.02	38.66	44.2	2.99	31.66
UNA(260k+20k)	44.31	31	30.1	39.85	45.8	3.26	32.39

Table 4: Impact of different distributions of instruction-tuning and alignment data on the new HuggingFace Open LLM Leaderboard

Model	gsm8k	truthful-qa	winogrande	arc	hellaswag	mmlu	average
Mistral	38.02	42.58	77.58	61.43	83.44	62.51	60.93
UNA(16k+20k)	40.83	**56.69**	79.4	64.85	**84.66**	62.22	**64.78**
UNA(20k+20k)	**41.59**	54.05	79.79	63.82	84.44	62.33	64.34
UNA(32k+20k)	40.83	54.38	79.72	64.42	84.46	**62.88**	64.45
UNA(65k+20k)	41.43	52.35	79.79	**65.27**	84.19	61.89	64.15
UNA(130k+20k)	41.13	50.17	79.48	64.59	84.07	62.26	63.62
UNA(260k+20k)	41.21	50.17	**80.43**	64.85	83.89	62.35	63.82

Table 5: Impact of different distributions of instruction-tuning and alignment data on the old HuggingFace Open LLM Leaderboard

Model	MT-Bench	Alpaca-eval	
		LC WR	Length
Mistral	3.15	0.31	6554
UNA(16k+20k)	6.25	7.9	1378
UNA(20k+20k)	6.78	8.28	1317
UNA(32k+20k)	6.54	8.23	1324
UNA(65k+20k)	6.3	**9.92**	1338
UNA(130k+20k)	**6.83**	7.43	1361
UNA(260k+20k)	6.45	6.85	1378

Table 6: Impact of different distributions of instruction-tuning and alignment data on MT-Bench and Alpaca-eval

performance compared to other SFT and alignment methods pose challenges. In contrast, PAFT requires training separate adaptors for SFT and alignment and merging them through sparsity, which is inefficient. Inspired by UNA's achievements, we aim to unify SFT with alignment based on UNA's principles to avoid catastrophic forgetting.

5 Limitation

This paper has a couple of limitations that warrant attention in future research. First, the study is restricted to English, and the multilingual capabilities of the approach should be thoroughly evaluated. Additionally, the current research primarily utilizes datasets such as Ultrachat and Helpsteer, which are academic in nature. To enhance the applicability of the findings, further testing on industrial datasets is recommended.

6 Conclusion

Despite the extensive pretraining of LLMs using trillions of tokens and billions of parameters, they still fall short of being fully useful. SFT enhances LLMs with the capability to answer prompts effectively. Alignment techniques, including RLHF, DPO, KTO and UNA, can prevent the occurrence of undesired responses. Nevertheless, these sequential SFT+alignment methods can lead to catastrophic forgetting, where the model loses capabilities gained in previous stages. To address this issue, a Unified Fine-Tuning (UFT) approach has been proposed. UFT unifies SFT and alignment by utilizing a generalized implicit reward function, following the UNA work. Trained solely on instruction-tuning data, UFT outperforms SFT, which is attributed to the minimization of divergence from the pretrained model through KL divergence. By mixing instruction-tuning data with alignment data, UFT surpasses all three sequential SFT+alignment methods, mitigating catastrophic forgetting. Ultimately, we establish a unified fine-tuning framework that runs in parallel to pretraining.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *Preprint*, arXiv:2402.14740.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *Preprint*, arXiv:2310.12036.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022b. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard (2023-2024). https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

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

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *Preprint*, arXiv:2403.07691.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Dahyun Kim, Yungi Kim, Wonho Song, Hyeonwoo Kim, Yunsu Kim, Sanghoon Kim, and Chanjun Park. 2024. sdpo: Don't use your data all at once. *Preprint*, arXiv:2403.19270.

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

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch,

Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *Preprint*, arXiv:2402.13228.

Shiva Kumar Pentyala, Zhichao Wang, Bin Bi, Kiran Ramnath, Xiang-Bo Mao, Regunathan Radhakrishnan, Sitaram Asur, Na, and Cheng. 2024. Paft: A parallel training paradigm for effective llm fine-tuning. *Preprint*, arXiv:2406.17923.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024. From r to q^*: Your language model is secretly a q-function. *Preprint*, arXiv:2404.12358.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, Aliaksei Severyn, Jonathan Mallinson, Lior Shani, Gil Shamir, Rishabh Joshi, Tianqi Liu, Remi Munos, and Bilal Piot. 2024. Offline regularised reinforcement learning for large language models alignment. *Preprint*, arXiv:2405.19107.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Zhichao Wang, Bin Bi, Can Huang, Shiva Kumar Pentyala, Zixu James Zhu, Sitaram Asur, and Na Claire Cheng. 2024a. Una: Unifying alignments of rlhf/ppo, dpo and kto by a generalized implicit reward function. *Preprint*, arXiv:2408.15339.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Zixu, Zhu, Xiang-Bo Mao, Sitaram Asur, Na, and Cheng. 2024b. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *Preprint*, arXiv:2407.16216.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024c. Helpsteer2: Open-source dataset for training top-performing reward models. *Preprint*, arXiv:2406.08673.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. 2024. Some things are more cringe than others: Iterative preference optimization with the pairwise cringe loss. *Preprint*, arXiv:2312.16682.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *Preprint*, arXiv:2401.10020.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization. *Preprint*, arXiv:2404.11999.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

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

A SFT on 20k examples with Different LR

The impact of different learning rates on the performances of SFT can be found in Table. 7 and Table. 8 for the new and old HuggingFace Open LLM Leaderboards.

B UFT on 20k examples with Different LR and β

The impact of different learning rates and β on the performances of UFT can be found in Table. 9 and Table. 10 for the new and old HuggingFace Open LLM Leaderboards.

LR	bbh	gpqa	mmlu-pro	musr	ifeval	math-hard	average
$3e^{-6}$	45.87	29.39	**30.21**	42.41	25.2	**3.61**	29.45
$1e^{-5}$	47.14	29.59	**30.21**	41.74	25.77	3.12	29.6
$3e^{-5}$	**47.71**	**29.76**	30.1	40.95	26.96	3.01	29.75
$1e^{-4}$	46.04	28.72	29.35	42.94	**29.5**	2.66	**29.87**
$3e^{-4}$	42.79	28.17	25.22	**43.61**	19.78	2.3	26.98

Table 7: The impact of different learning rates for SFT on the new HuggingFace Open LLM Leaderboard

LR	gsm8k	truthful-qa	winogrande	arc	hellaswag	mmlu	average
$3e^{-6}$	42.95	47.9	**79.32**	61.86	83.34	62.37	62.96
$1e^{-5}$	43.25	49.58	79.08	62.29	83.33	62.15	63.28
$3e^{-5}$	**44.47**	50.5	78.93	62.54	83.25	**62.61**	**63.72**
$1e^{-4}$	39.65	**51.06**	78.53	**63.99**	**83.78**	61.99	63.17
$3e^{-4}$	13.39	46.43	75.85	58.53	81.16	55.73	55.18

Table 8: The impact of different learning rates for SFT on the old HuggingFace Open LLM Leaderboard

LR/ β	bbh	gpqa	mmlu-pro	musr	ifeval	math-hard	average
$1e^{-4}/0.01$	**47.01**	27.86	30.3	39.07	**32.49**	3.57	30.08
$1e^{-4}/0.03$	46.09	29.5	30.09	40.8	29.27	3.84	29.93
$1e^{-4}/0.1$	45.57	30.5	30.15	42.42	27.21	2.94	29.8
$1e^{-4}/0.3$	44.02	30.04	29.65	**43.76**	26.08	2.64	29.37
$3e^{-5}/0.01$	46.55	29.24	30.25	41.73	28.89	**3.87**	**30.09**
$3e^{-5}/0.03$	46.56	28.72	30.44	40.67	29.7	3.08	29.86
$3e^{-5}/0.1$	46.31	29.29	**30.7**	40.95	26.52	3.5	29.55
$3e^{-5}/0.3$	44.91	28.6	30.44	43.1	26.54	3.1	29.45
$1e^{-5}/0.01$	46.44	29.27	30.44	38.15	26.02	3.59	28.99
$1e^{-5}/0.03$	46.46	29.51	30.54	39.21	25.39	3.02	29.02
$1e^{-5}/0.1$	45.8	29.99	30.41	42.96	26.67	3.01	29.81
$1e^{-5}/0.3$	45.03	29.74	30.46	42.97	25	2.57	29.3
$3e^{-6}/0.01$	45.83	29.86	30.5	41.5	25.42	2.8	29.32
$3e^{-6}/0.03$	45.81	29.91	30.45	41.24	25.23	2.58	29.2
$3e^{-6}/0.1$	45.48	**30.61**	30.53	43.09	23.87	3.27	29.48
$3e^{-6}/0.3$	44.3	29.48	30.29	42.71	22.56	3.06	28.73

Table 9: The impact of different learning rates for SFT on the new HuggingFace Open LLM Leaderboard

LR/ β	gsm8k	truthful-qa	winogrande	arc	hellaswag	mmlu	average
$1e^{-4}/0.01$	43.64	49.77	78.77	63.91	**83.88**	62.31	63.71
$1e^{-4}/0.03$	42.57	50.2	78.61	**64.25**	83.55	61.87	63.51
$1e^{-4}/0.1$	40.11	47.95	77.98	63.4	83.37	62.19	62.5
$1e^{-4}/0.3$	35.94	44.4	78.93	62.2	82.92	62	61.07
$3e^{-5}/0.01$	**45.57**	51.18	78.93	63.82	83.54	62.44	**64.25**
$3e^{-5}/0.03$	43.33	**51.24**	78.61	64.08	83.41	**62.61**	63.88
$3e^{-5}/0.1$	41.55	49.25	78.69	62.54	83.31	62.44	62.96
$3e^{-5}/0.3$	38.44	43.94	78.14	61.77	83.5	62.31	61.35
$1e^{-5}/0.01$	42.46	50.41	78.69	63.14	83.27	62.43	63.4
$1e^{-5}/0.03$	41.58	50.35	**79.01**	62.71	83.35	62.59	63.27
$1e^{-5}/0.1$	41.17	48.63	78.69	62.54	83.38	62.35	62.79
$1e^{-5}/0.3$	39.24	44.24	78.45	62.2	83.55	62.5	61.7
$3e^{-6}/0.01$	39.92	48.42	78.53	62.2	83.32	62.35	62.46
$3e^{-6}/0.03$	40.41	48.08	78.69	62.2	83.34	62.5	62.54
$3e^{-6}/0.1$	38.63	46.3	78.22	61.95	83.39	62.58	61.85
$3e^{-6}/0.3$	38.86	43.99	77.9	61.52	83.48	62.59	61.39

Table 10: The impact of different learning rates for SFT on the old HuggingFace Open LLM Leaderboard