Intuitive Fine-Tuning: Towards Simplifying Alignment into a Single Process

Anonymous ACL submission

Abstract

Supervised Fine-Tuning (SFT) and Preference Optimization (PO) are key processes for aligning Language Models (LMs) with human preferences post pre-training. While SFT excels in efficiency and PO in effectiveness, they are often combined sequentially without integrating their optimization objectives. This approach ignores the opportunities to bridge their paradigm gap and take the strengths from both. In this paper, we interpret SFT and PO with two sub-processes — *Preference Estimation* and *Transition Optimization* — defined at token level within the Markov Decision Process (MDP). This modeling shows that SFT is only a special case of PO with inferior estimation and optimization. PO estimates the model's preference by its entire generation, while SFT only scores model's subsequent predicted tokens based on prior tokens from ground truth answer. These priors deviates from model's distribution, hindering the preference estimation and transition optimization. Building on this view, we introduce ***Intuitive Fine-Tuning (IFT)*** to integrate SFT and PO into a single process. Through a temporal residual connection, IFT brings better estimation and optimization by capturing LMs' intuitive sense of its entire answers. But it solely relies on a single policy and the same volume of non-preference-labeled data as SFT. Our experiments show that IFT performs comparably or even superiorly to SFT and some typical PO methods across several tasks, particularly those requires generation, reasoning, and fact-following abilities. An explainable Frozen Lake game further validates the effectiveness of IFT for getting competitive policy.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable powerful potential across various downstream tasks after pre-training on large-scale corpora (Brown et al., 2020; Achiam et al., 2023; Zhou and Ding, 2024). However, their instruction-following skills and trustworthiness still fall short of expectations (Bender et al., 2021; Bommasani et al., 2021; Li et al., 2022). Therefore, algorithms such as Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al., 2019; Ouyang et al., 2022; Lee et al., 2023) are used to further enhance LLMs' abilities and align them better with human preferences.

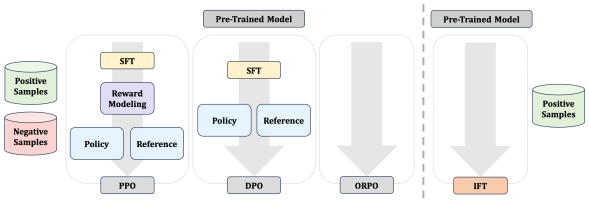

Figure 1: Comparison of Alignment Methods. IFT conducts alignment solely relying on positive samples and a single policy, starting from a pre-trained base model. IFT shows similar efficiency as SFT and effectiveness as PO methods.

Considering the limited effectiveness of SFT and the high cost of data construction and training computation for RLHF, these two methods are often combined to leverage their respective strengths. Unfortunately, they are typically implemented as a sequential recipe constrained by the paradigm gap between SFT and early RLHF methods, stemming from differences in loss functions, data formats, and the requirement for auxiliary models.

Recently, a method named Direct Preference Optimization (DPO) (Rafailov et al., 2024) was proposed to integrate Reward Modeling and Policy Optimization into one single procedure using a loss function derived from Proximal Policy Optimization (PPO) (Schulman et al., 2017). This approach demonstrates the potential to unify SFT and RLHF for the first time. Henceforth, many extended methods have been tried to realize this objective by bridging the gap between SFT and DPO.

Some of them (Ethayarajh et al., 2024; Hong et al., 2024; Zhang et al., 2024) aim to transform the contrastive loss of DPO into a SFT-like cross-entropy loss, learning positive samples similar to SFT while unlearning negative samples resort to Unlikelihood Training (Welleck et al., 2019). Some others get rid of the preference-labeling process before training, switching to collect samples and labels/rewards in an online manner (Liu et al., 2023a; Yuan et al., 2024; Guo et al., 2024a; Calandriello et al., 2024; Tajwar et al., 2024), or just treating the SFT targets and online policy generations as positive and negative samples respectively (Xiong et al., 2023; Chen et al., 2024; Mitra et al., 2024; Liu et al., 2024). Nevertheless, preference-labeled pairwise data is still essential, and the need for reference model only becomes unnecessary in some cases. Thus the core differences between SFT and Preference Optimization (PO) are not eliminated thoroughly. To address this challenging issue, a deeper and more unified understanding of them are needed.

In this paper, we attempt to explain the similarities and differences between SFT and PO by defining Preference Estimation and Transition Optimization in terms of state-action pairs within the Markov Decision Process (MDP) framework. Through this modeling, we demonstrate that SFT is simply a specialized case of PO with inferior estimation and optimization than other methods. To estimate the policy preference, PO collects sentence-level negative samples from policy for each initial instruction. However, SFT only samples subsequent token for each intermediate state of ground truth answer, which leads to a biased estimation of policy preference and an inferior alignment performance.

Depending on this understanding, we introduce a unified alignment algorithm named Intuitive Fine-Tuning (IFT). Drawing inspiration from the human ability to grasp a intuitive sense of an answer after hearing a question, IFT employs a Temporary Residual Connection across tokens to approximate policy's entire answer for each instruction. This approach helps IFT better estimate the policy's preference than SFT, achieving alignment performance comparable or even superior to the sequential recipe of SFT and Preference Optimization. Additionally, IFT requires only a single policy model, and the same volume and format of data as SFT, enjoying both data and computation efficiency. These characteristics of IFT are advantageous in domains where preference data is unavailable or expensive to collect.

Our main contribution are three folds:

(1) Through defining Preference Estimation and Transition Optimization using the MDP, we demonstrate that SFT is only a special case of Preference Optimization. The similarities and differences of SFT, PPO and online/offline DPO are also compared within this framework;

(2) We introduce Intuitive Fine-tuning (IFT), a deeply unified version of SFT and Preference Optimization. It utilizes temporary residual connections to extract the model's generation preference given the initial instructions. IFT enjoys the similar efficiency as SFT on negative sampling, but can better estimate and optimize the policy preference.

(3) Through experiments on several benchmarks, we validate that IFT performs comparably or superiorly to SFT and various Preference Optimization methods. An explainable toy-setting Frozen Lake further demonstrates the effectiveness of IFT.

2 Preliminaries

2.1 MDP in Language Models

The MDP applied to LMs can be formally described as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \rho_0)$, where $\mathcal{S}$ is the state space comprising ordered permutations of vocabularies, $\mathcal{A}$ is the action space consisting of vocabularies defined by the tokenizer, $\mathcal{T}$ is the transition matrix indicating token generation probabilities for given states, r represents rewards for state-action pairs, and ρ_0 is the initial state typically based on given instructions. See more details in Appendix A.1.

The primary objective of Language Modeling is to train a policy π_θ with $\mathcal{T}_\theta$ to mimic a human policy π^* with $\mathcal{T}^*$, aiming for the two transition matrices to become identical:

$$\forall s \in \mathcal{S}, a \in \mathcal{A} : \mathcal{T}_\theta(a|s) \rightarrow \mathcal{T}^*(a|s) \quad (1)$$

This process can also be expressed using another state-state transition matrix T:

$$\forall\ s, s' \in S : T_\theta(s'|s) \rightarrow T^*(s'|s) \quad (2)$$

where T is equivalence to $\mathcal{T}$, but instead, indicating the transition probability between states.

2.2 Preference Estimation

We define the preference $\mathcal{P}$ of policy π given an initial instruction ρ_0 as a mapping:

$$\mathcal{P}(\rho_0) : \rho_0 \rightarrow [\pi(\rho_0), \pi(s_1), \pi(s_2), ...] \quad (3)$$

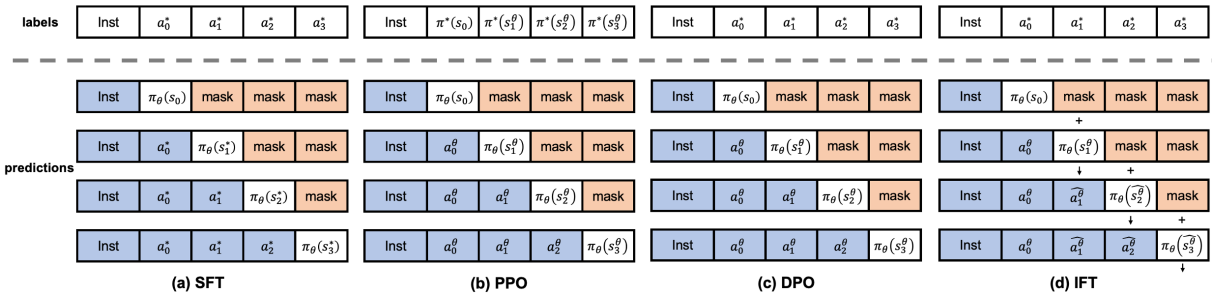

Figure 2: The Training Paradigm of Different Methods. Symbol $*$ and θ denote human and model respectively, with $a_i^* = \pi^*(s_i^*)$ and $s_{i+1}^* = [s_i^*, a_i^*]$, similarly for θ. SFT uses priors deviating from model distribution, resulting in a more biased estimation of model preferences compared to PPO and DPO. IFT achieves a better estimation than SFT by Temporary Residual Connections across tokens. This approach passes the residual embedding from one token to the next, creating a more accurate prior while maintaining the data and computational efficiency of SFT.

where $s_{i+1} = [s_i, a_i]$, $a_i = \pi(s_i)$ and $s_0 = \rho_0$.

During alignment, the model preference gradually approaches the human preference:

$$\mathcal{P}_\theta(\rho_0) \to \mathcal{P}^*(\rho_0) \tag{4}$$

$$\begin{aligned} \mathcal{P}_\theta(\rho_0) &: \rho_0 \to [\pi_\theta(\rho_0), \pi_\theta(s_1^\theta), \pi_\theta(s_2^\theta), \ldots] \\ \mathcal{P}^*(\rho_0) &: \rho_0 \to [\pi^*(\rho_0), \pi^*(s_1^*), \pi^*(s_2^*), \ldots] \end{aligned} \tag{5}$$

As the truly preferences are difficult to obtain, alignment is usually conducted based on the Preference Estimation of model and human, denoted as $\hat{\mathcal{P}}_\theta$ and $\hat{\mathcal{P}}^*$ respectively. The estimations from some typical methods are listed in Table 1.

To make preference optimizable, the policy's preference can also be expressed as follows:

$$\mathcal{P}(\rho_0) = \{\mathcal{T}(a|s) | \forall a \in \mathcal{A}, s \in \mathcal{S}_{\rho_0}\} \tag{6}$$

Here, $\mathcal{S}_{\rho_0}$ denotes a conditional state space that constrained by the initial state ρ_0, within which each state can only be initially derived from ρ_0. Consequently, the model preference can be optimized through transition matrix, named Transition Optimization.

2.3 Transition Optimization

Ideally, we want to align the state-action transition matrix between model and human in a ρ_0-constrained state space:

$$\forall a \in \mathcal{A}, s \in \mathcal{S}_{\rho_0} : \mathcal{T}_\theta(a, s) \to \mathcal{T}^*(a, s) \tag{7}$$

which is equivalent to the following format expressed by state-state transition matrix:

$$\forall s \in \mathcal{S}_{\rho_0} : T_\theta(s, \rho_0) \to T^*(s, \rho_0) \tag{8}$$

However, considering the limited data, only matrix elements representing state-action/state-state pairs contained in the dataset $\mathcal{D}$ would be aligned. Given a data sample with instruction ρ_0 and target answer with length-N, the objective would be $\forall a \in \mathcal{A}, n \in [0, N], \rho_0 \in \mathcal{D}, s_n^* \in \mathcal{S}_{\rho_0}^*$:

$$\mathcal{T}_\theta(a, s_n^*) \to \mathcal{T}^*(a, s_n^*) \tag{9}$$

Or equivalent to $\forall n \in [0, N], \rho_0 \in \mathcal{D}, s_n^* \in \mathcal{S}_{\rho_0}^*$:

$$T_\theta(s_n^*, \rho_0) \to T^*(s_n^*, \rho_0) \tag{10}$$

where $s_0^* = \rho_0$, $T^*(\rho_0|\rho_0) = T_\theta(\rho_0|\rho_0) = 1$, and s_i^* denotes the intermediate state of target answer.

Consequently, the loss function can be derived from the disparities of the transition matrices between model and human. Some typical loss function are listed in Appendix A.4.

3 From SFT to Preference Optimization

We reformulate SFT, PPO and DPO using the aforementioned framework, detailed in Table 1 and Appendix A.4. A more comprehensible version is presented in Figure 2. To compare the differences between them, we begin by introducing a fundamental theorem and corollary:

Theorem *Given a set of events $\mathcal{Z}$, the probability of any event $z \in \mathcal{Z}$ is between 0 and 1, i.e., $\forall z \in \mathcal{Z} : 0 \leq P(z) \leq 1$. If all events are mutually independent, the sum of their probabilities equals 1, i.e., $1 = \sum_{z \in \mathcal{Z}} P(z)$. The event z^* with the highest probability has a probability greater than or equal to any other event, i.e., $\forall z \in \mathcal{Z} : 0 \leq P(z) \leq P(z^*) \leq 1$.*

Corollary *LMs consistently assign higher probabilities to their own greedy predictions than to human preference:*

$$\forall s \in S : \mathcal{T}_\theta(\pi^*(s), s) \leq \mathcal{T}_\theta(\pi_\theta(s), s) \leq 1 \tag{11}$$

thus LMs tend to assign higher probabilities to its own generation than to target answer given the same initial instruction $\forall n \in [0, N], s_n^* \in \mathcal{S}_{\rho_0}^*, s_n^\theta \in \mathcal{S}_{\rho_0}^\theta$*:*

$$T_\theta(s_n^*, \rho_0) \leq T_\theta(s_n^\theta, \rho_0) \leq 1 \tag{12}$$

where N represents the length when the generation reaches the EOS token or the truncation length.

SFT provides an unbiased estimation of human preference, but a biased estimation for model:

$$\hat{\mathcal{P}}_\theta(\rho_0) : \rho_0 \rightarrow [\pi_\theta(\rho_0), \pi_\theta(s_1^*), \pi_\theta(s_2^*), \ldots] \tag{13}$$

which is caused by wrong prior state when predicting each subsequent token. Consequently, the Transition Optimization objective of SFT:

$$T_\theta(s_n^*, s_{n-1}^*) \rightarrow T^*(s_n^*, s_{n-1}^*) \tag{14}$$

secretly sets $T_\theta(s_{n-1}^*, \rho_0) = 1$ during aligning $T_\theta(s_n^*, \rho_0)$ with $T^*(s_n^*, \rho_0)$. This makes an overestimation of the transition probabilities and preference of model, leading to an inferior optimization progress in SFT. Thus Preference Optimization is needed for further preference alignment.

PPO shows an unbiased estimation of model preference, while employing a progressively unbiased estimation of human preference:

$$\hat{\mathcal{P}^*}(\rho_0) : \rho_0 \rightarrow [\pi^*(\rho_0), \pi^*(s_1^\theta), \pi^*(s_2^\theta), \ldots] \tag{15}$$

Initially biased, this estimation gradually becomes unbiased as the model aligns with human preference over time. As $T_\theta(s_n^\theta, \rho_0)$ is consistently closer to 1 than $T_\theta(s_{n-1}^*, \rho_0)$, PPO provides an closer approximation than SFT to the actual circumstances of model in Transition Optimization:

$$T_\theta(\hat{s_n^*}, s_{n-1}) \rightarrow T^*(\hat{s_n^*}, s_{n-1}) \tag{16}$$

which sets $T_\theta(s_n^\theta, \rho_0) = 1$ and $\hat{s_n^*} = \pi^*(s_{n-1}^\theta)$. However, estimating $\pi^*(s_{n-1}^\theta)$ is at the expense of preference-labeling, reward modeling and online sampling.

DPO theoretically achieves the best estimation across all scenarios, even without reward modeling. However, obtaining pairwise preference data online is costly, as it requires real-time negative sampling from model and preference labeling by human. Thus, mainstream implementations often rely on off-policy negative samples out-of-distribution from the optimized model, which may yield unstable and sub-optimal results due to biased preference estimation and inferior transition optimization.

Method		Preference Estimation		Transition Optimization
		$\hat{s_n^*}$ in $\hat{\mathcal{P}^*}$	$\hat{s_n^\theta}$ in $\hat{\mathcal{P}_\theta}$	
Truly		s_n^*	s_n^θ	$T_\theta(s_n^*, \rho_0) \rightarrow T^*(s_n^*, \rho_0)$
SFT		s_n^*	s_n^*	$T_\theta(s_n^*, s_{n-1}^*) \rightarrow T^*(s_n^*, s_{n-1}^*)$
PPO		s_n^θ	s_n^θ	$T_\theta(\hat{s_n^*}, s_{n-1}^\theta) \rightarrow T^*(\hat{s_n^*}, s_{n-1}^\theta)$
DPO	online	s_n^*	s_n^θ	$T_\theta(s_n^*, \rho_0) \rightarrow T^*(s_n^*, \rho_0)$
	offline	s_n^*	$s_n^{\theta^-}$	$\hat{T}_\theta(s_n^*, \rho_0) \rightarrow T^*(s_n^*, \rho_0)$

Table 1: Reformulation of SFT, PPO and DPO

4 Method

While SFT is data and computation-efficient, it has an inferior approximation for both Preference Estimation and Transition Optimization. On the other side, Preference Optimization (represented by PPO and DPO) enjoys better approximation at the expense of preference data construction. We hope to make good use of their strength, using solely target data as SFT but having a similar approximation as Preference Optimization. See pseudo code in Appendix B.3.

4.1 Intuitive Preference Estimation

A key distinction between SFT and Preference Optimization is whether the full distribution of model's preference for each initial instruction is sampled. Preference Optimization samples the policy's entire answer to estimate its preference, ensuring each generation relies on the prior adheres to the model's distribution. But SFT only samples subsequent tokens the intermediate state of the target answer, the used prior may be far away from the model preference, leading to inferior preference estimation for model.

To obtain a prior state estimation $\hat{s_i^\theta}$ closer to model distribution, we introduce a model-based distribution disturbance function δ_θ for the biased prior state:

$$\hat{s_i^\theta} = \delta_\theta(s_i^*) = (1 - \lambda)s_i^* + \lambda\pi_\theta(s_{i-1}^*) \tag{17}$$

which can also be interpreted as a temporal residual connection that passes the residual embedding from one token to the next. Through this approach, model can predict not only the next token from intermediate state of target answer, but also develop an intuitive sense to the entire answer generation solely based on the initial instruction, deriving more unbiased prior and accurate Preference Estimation for model:

$$\hat{\mathcal{P}_\theta}(\rho_0) = [(1 - \lambda)\mathcal{P}_\theta{}^{sft} + \lambda\mathcal{P}_\theta{}^{truly}](\rho_0) \tag{18}$$

With improved Preference Estimation, we achieve a Transition Optimization process closer to the original objective $\forall n \in [0, N], \rho_0 \in \mathcal{D}, s_n^* \in \mathcal{S}_{\rho_0}^*$:

$$\hat{T}_\theta(s_n^*, \rho_0) \rightarrow T^*(s_n^*, \rho_0) \tag{19}$$

where $s_0^* = \rho_0$ and $\hat{T}_\theta(s_n^*, \rho_0) = \prod_{i=0}^{n-1} T_\theta(s_{i+1}^*, \hat{s_i^\theta})$.

This objective can be optimized by the following loss function, which quantifies the disparities of transition between model and human:

$$\mathcal{L}(\mathcal{T}_\theta, \delta_\theta) = \mathbb{E}\left[-\sum_{n=0}^{N} \log \mathcal{T}_\theta(a_i^*, \delta_\theta(s_i^*))\right] \tag{20}$$

where $a_i^* = \pi^*(\delta^*(s_i^*)) = \pi^*(s_i^*)$. See Appendix A.2 for complete derivation.

4.2 Dynamic Relation Propagation

The Intuitive Preference Estimation implicitly performs Dynamic Relation Propagation, during which the generation of future tokens will be influenced by the prediction accuracy of current token.

However, limited by the parallel computing mode, the gradient map could only be built on the same time-step. Thus, the current generated tokens is unable to obtain gradient feedback from the future generated tokens. Therefore, we reformulate the loss function by a differentiable cumulative-summation to get around this limitation:

$$\mathcal{L}_{\text{IFT}} = \mathbb{E}\left[-\sum_{n=0}^{N}\sum_{i=n}^{N} \log \mathcal{T}_\theta(a_i^*, \delta_\theta(s_i^*))\right] \tag{21}$$

This reformulation implicitly satisfies the Bellman Equation for each state, which guarantees the optimization enjoys both of the effectiveness as RLHF and efficiency as SFT:

$$V_\theta(\hat{s_n^\theta}) = \exp\left(-\mathcal{L}\big(\hat{T}_\theta(s_n^*, \rho_0)\big)\right) \tag{22}$$

The derivation is in Appendix A.3. Additionally, a decay factor can be incorporated to ensure effectiveness in long trajectories, as in the typical Bellman Equation.

5 Experiments

We conduct experiments mainly on NLP setting. Considering the absence of an optimal policy of human language generation, we also utilize the Frozen Lake environment for further validation.

5.1 Settings for NLP

Datasets. We select UltraChat-200k (Ding et al., 2023) and UltraFeedback-60k (Cui et al., 2023) as single-target and pair-wise dataset respectively.

Models. We conduct experiments on Mistral-7B-v0.1 (Jiang et al., 2023) and Mistral-7B-sft-beta (Tunstall et al., 2023), with the former as the base model and the latter fine-tuned on UltraChat-200k.

Scenarios. We consider two different training scenarios, one using Preference Optimization exclusively, and the other employing sequential recipe of SFT and Preference Optimization. In the first scenario, alignment is conducted directly from base model Mistral-7B-v0.1 using UltraFeedback. In order to ensure balanced data volume between different method, we randomly sample 60k data from UltraChat as supplementary for SFT and IFT, for only the target data are utilized in these two methods. The second scenario is commonly seen, where SFT and Preference Optimization is employed sequentially. For this scenario, we use Mistral-7B-sft-beta as start-point, which has been fine-tuned with UltraChat using SFT. Then we fine-tune it further with UltraFeedback using Preference Optimization.

Baselines. SFT and DPO (Rafailov et al., 2024) are our main baselines, and we exclude PPO due to computational limitations. We also incorporate three improved versions of DPO: TDPO (Zeng et al., 2024), ORPO (Hong et al., 2024), and SimPO (Meng et al., 2024). TDPO transformers the DPO loss into token-level to make its objective closer to SFT. SimPO adds on a length-normalization term to replace the regularization from reference model. ORPO adds the SFT loss and a DPO-like loss together, achieving alignment directly without SFT and reference model. In addition to reproducing the algorithms mentioned above, we also consider Zephyr-7B-beta (Tunstall et al., 2023) and Mistral-ORPO-alpha (Hong et al., 2024), two open-source checkpoints that utilize sequential and direct recipes respectively. Both of them used start-point models and datasets similar to ours.

Benchmarks. We consider two types of benchmarks. One is from the widely used Open-LLM LeaderBoard, which contains ARC-Challenge(25-shot) (Clark et al., 2018), MMLU(5-shot) (Chung et al., 2024), TruthfulQA(0-shot) (Lin et al., 2021), WinoGrande(5-shot) (Sakaguchi et al., 2021), and GSM8K(5-shot) (Cobbe et al., 2021). The other is LM-based evaluation, including TL;DR (Völske

Method	Reference	Data		ARC	ARC-Gen	MMLU	TruthfulQA	WinoGrande	GSM8K	Avg.
		pairwise	volume							
Mistral-7B	–	–	–	53.07	73.04	59.14	45.29	77.58	38.89	54.79
fine-tuning with UltraFeedback-60k										
+ SFT	✗	✗	120k	56.49	74.00	60.44	55.57	77.90	42.84	58.65
+ DPO	✓	✓	120k	**61.86**	73.54	**61.02**	47.98	76.64	43.89	58.28
+ TDPO	✓	✓	120k	56.06	73.72	60.23	43.94	77.03	41.70	55.79
+ ORPO	✗	✓	120k	56.66	73.98	60.57	51.77	77.19	42.30	57.70
+ SimPO	✗	✓	120k	59.90	73.55	52.61	47.25	78.30	37.53	55.15
+ IFT	✗	✗	120k	56.74	**74.15**	60.49	**57.65**	**78.45**	**44.73**	**59.61**
Mistral-ORPO-α	✗	✓	120k	57.25	73.72	58.74	60.59	73.72	46.78	59.41
fine-tuning with Ultrachat-200k + UltraFeedback-60k sequentially										
+ SFT	✗	✗	200k	57.68	72.87	58.25	45.78	77.19	40.94	55.97
+ SFT + SFT	✗	✗	260k	58.10	72.61	58.40	48.59	76.80	43.06	56.99
+ SFT + DPO	✓	✓	320k	**63.91**	**73.98**	**59.75**	46.39	76.06	41.47	57.52
+ SFT + TDPO	✓	✓	320k	59.13	73.72	58.92	46.63	76.32	**44.58**	57.12
+ SFT + ORPO	✗	✓	320k	58.45	73.21	58.80	50.31	76.45	42.76	57.35
+ SFT + SimPO	✗	✓	320k	60.83	73.63	59.01	49.45	76.95	38.44	56.94
+ SFT + IFT	✗	✗	260k	58.36	73.38	58.45	**52.39**	**78.06**	43.82	**58.22**
Zephyr-7B-β	✓	✓	320k	67.41	72.61	58.74	53.37	74.11	33.89	57.50

Table 2: Evaluation on Open-LLM Leaderboard with chat template. When fine-tuning with the same recipe, IFT achieves the highest average score across all methods. Directly conducting alignment using IFT showcases the best performance in all recipes with the least data and computation.

et al., 2017), Alpaca-Eval, and Alpaca-Eval-2 (Dubois et al., 2024). As for TL;DR, we keep the same setting as (Rafailov et al., 2024), using GPT-4 to judge the win-rate between model's generation and ground truth answer. We utilize chat template for all benchmarks to obtain a more accurate evaluation for chat models.

5.2 Main Results in NLP Tasks

Effectiveness on Sequential Recipe. In this scenario, IFT demonstrates good performance across benchmarks having standard answers or not, see more details in Table 2 and 3. On Open-LLM Leaderboard, IFT showcases the best average capabilities across all tasks, excelling particularly in tasks requiring generation, reasoning and fact-following abilities, such as TruthfulQA and GSM8K. However, IFT has a relatively large gap between DPO in multi-choice tasks like ARC-Challenge and MMLU. When evaluated for conversation and summarization judged by GPT-4, IFT's performance is comparable to that of the chosen baselines. Remarkably, IFT achieves these results using the least amount of data and computational resources among all the methods tested.

Effectiveness of Preference Optimization Alone. IFT not only maintains the performance advantages compared with other baselines in this setting, as seen in the sequential scenario. But also, IFT performs comparably or even superiorly to many method in sequential recipe. While DPO, SimPO and TDPO tend to fail under this setting, ORPO remains competitive in its open-source model. However, when constrained in the same experiment setting, the performance of ORPO becomes worse than IFT. Additionally, the reliance on preference data makes ORPO more costly in terms of negative sampling, preference labeling, and GPU memory consumption. Consequently, IFT stands out as a more efficient and cost-effective alternative in this context.

Multi-Choice vs. Generation. IFT performs better on generation tasks but struggles with multi-choice, whereas DPO exhibits the opposite performance. This may due to differences in evaluation metrics and training objectives (Zheng et al., 2023; Plaut et al., 2024; Tsvilodub et al., 2024). Multi-choice tasks evaluate log-likelihood for entire answers, while generation tasks require token-by-token construction for causality and reasoning. DPO aligns the mapping between instructions and complete answers, while IFT emphasizes token-level causal relationships. As a result, DPO tends to excel in multi-choice tasks, while IFT performs better in token-by-token exploration tasks. In an ARC-Challenge adaptation to generation tasks,

Method	Alpaca-Eval		Alpaca-Eval-2		TL;DR
	win-rate	lc win-rate	win-rate	lc win-rate	win-rate
Mistral-7B	24.72	11.57	1.25	0.35	92.03
fine-tuning with UltraFeedback-60k					
+ SFT	82.56	78.32	7.09	8.67	84.22
+ DPO	74.00	73.12	9.73	8.58	77.25
+ TDPO	65.74	51.41	4.99	3.47	70.82
+ ORPO	85.14	76.60	8.82	12.34	89.24
+ SimPO	83.08	64.30	**24.47**	**20.31**	59.13
+ IFT	**85.18**	**78.78**	9.95	13.27	**92.63**
Mistral-ORPO-α	87.92	–	–	11.33	–
fine-tuning with UltraChat-200k + UltraFeedback-60k sequentially					
+ SFT	86.69	77.96	4.08	6.43	98.11
+ SFT + SFT	86.34	76.98	4.55	7.14	97.79
+ SFT + DPO	**91.62**	**81.54**	10.08	13.72	**99.18**
+ SFT + TDPO	89.80	76.44	9.25	14.15	98.89
+ SFT + ORPO	86.26	79.67	7.40	12.27	97.92
+ SFT + SimPO	88.79	68.88	**19.62**	**23.94**	98.23
+ SFT + IFT	88.37	81.29	10.26	14.34	98.57
Zephyr-7B-β	90.60	–	–	10.99	–

Table 3: Evaluation on LLM-based Benchmarks. IFT secures top two rankings in nearly all tasks, including conversation and summarization. When fine-tuned on limited data from UltraFeedback, IFT demonstrates a significant lead in TL;DR.

IFT demonstrates superiority without changing the benchmark's distribution. Overall, IFT showcases its balanced performance across diverse tasks and achieving the highest average score.

Objective Trade-off between SFT and Preference Optimization. Traditional Preference Optimization methods deliver excellent alignment performance, particularly in enhancing the instruction-following ability of language models, as showed in Table 3. However, fitting the different objectives of SFT and Preference Optimization involves trade-offs (Tunstall et al., 2023). Even slight overfitting on SFT may result in reduced effectiveness of Preference Optimization. This phenomenon is also observed in Table 2, where the models trained by sequential recipe of SFT and other Preference Optimization methods showcase obvious inferior results on Open-LLM Leaderboard even worse than SFT alone. Avoiding this trade-off, ORPO and IFT can achieve better and more stable performance by directly conducting alignment on the base model.

Efficiency and Scaling Potential of IFT. Although IFT achieves comparable or superior performance to other methods, it also boasts high efficiency in many aspects. IFT does not require a reference model, which conserves GPU memory and computational resources. Most importantly, IFT and SFT are the only methods that conduct alignment without preference data, offering significant benefits as follows. Firstly, this characteristic

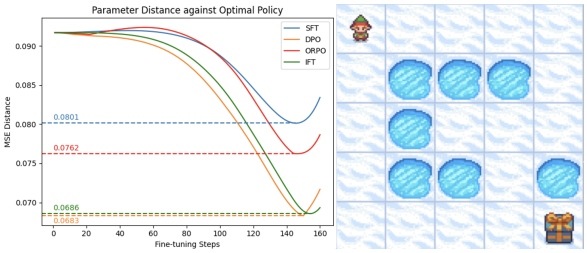

Figure 3: The Frozen Lake Game. Considering the MSE distance between transition matrices of the trained and optimal policy, IFT performs much better than SFT and ORPO, but slightly worse than DPO.

eliminates the need for synchronous storage and computation of pairwise data on the GPU, thereby reducing memory consumption and training duration. Secondly, negative sampling from models and human preference-labeling are no longer necessary, eliminating the highest cost associated with alignment, which has been a discarded but fundamental challenge in research so far. Furthermore, using only the target answer brings the potential for scaling in alignment or even in pre-training.

5.3 Further Validation in Frozen-Lake Environment

As scores on Open-LLM Leaderboard only partially reflect models' performance, and GPT-4 inadequately models human language generation, further comparison to a truly optimal policy is necessary. Given the difficulty of obtaining an optimal policy representing human language, we validate our algorithm in a simplified setting called Frozen Lake (Farama, 2023). In this environment, an agent attempts to find a gift on a nearly frozen lake with several holes, terminating the game upon finding the gift or falling into a hole. The limited number of states and actions in this game allows the optimal policy to be easily derived using classical RL methods.

To simulate parameterized policy alignment, we employ a two-layer fully connected neural network and design the environment with one optimal and one sub-optimal trajectory. The optimal parameterized policy is trained using the previously obtained optimal state-action transition matrix, and various fine-tuning methods from LMs are compared. We evaluate performance by measuring the MSE distance between the transition matrices of the optimal and trained policy. We didn't count in TDPO and SimPO, as their objectives are similar as DPO in Frozen Lake Game.

In this setting, IFT achieves a significantly better policy than SFT and ORPO, although it performs slightly worse than DPO. This is partly because, in terms of comparing how closely the explored grid aligns with the agent's preference, the order is DPO $>$ IFT $>$ ORPO $>$ SFT. Although ORPO also considers the negative trajectories sampled from policy, its direct incorporation of SFT loss with a fusion coefficient deviates its preference estimation, partially diminishing its effectiveness. Additionally, DPO, ORPO and IFT explore more grids than SFT, which helps the agent develop a better understanding of the environment.

6 Related Work

Classical Reinforcement learning (RL) has demonstrated strong performance in various sequential decision-making and optimal control domains, including robotics (Levine et al., 2018), computer games (Vinyals et al., 2019) and others (Guan et al., 2021). There are two main categories of RL algorithms: value-based and policy-based, depending on whether they learn a parameterized policy. Value-based RL aims to fit an value function defined by Bellman Equation, containing methods such as Monte-Carlo (MC) Learning (Lazaric et al., 2007) and Temporal Difference Learning (Sutton, 1988; Seijen and Sutton, 2014). However, value-based methods struggle in continuous or large discrete space for its greedy objective. Thus, policy-based methods were introduced to model the decision-making process using a parameterized policy. As one of its best-known algorithms, Proximal Policy Optimization (PPO) (Schulman et al., 2017) is widely used in various domains, including Natural Language Processing (NLP).

Alignment for LMs has emerged as a crucial task these years, which adjusts the LMs' generation distribution in line with human preferences (Bradley and Terry, 1952; Ziegler et al., 2019; Ouyang et al., 2022; Lee et al., 2023). While PPO remains the primary algorithm for alignment, its high demands for computation and memory hinders its broader use. Consequently, many improved methods have been proposed (Dong et al., 2023; Yuan et al., 2023; Zhao et al., 2023). Among them, DPO (Rafailov et al., 2024) unifies reward modeling and policy optimization by utilizing a loss function derived from PPO, training a single model to serve as both a policy model and a reward model. Without sacrificing performance, DPO decrease the costly consumption of PPO through directly value iteration similar to a preference-based format of MC instead of TD. However, it still relies on an expensive preference-labeling process and requires an SFT-based warm-up stage, which may introduce trade-offs when aligning the objectives of SFT and Preference Optimization.

Improved Versions of DPO come out one after another. Efforts such as (Liu et al., 2023b; Khaki et al., 2024; Yin et al., 2024; Guo et al., 2024b; Bansal et al., 2024; Liu et al., 2024) try to enhance the contrastive learning by utilizing better ranking strategies, more informative data, or more number of negative samples. Except for using offline data, (Liu et al., 2023a; Yuan et al., 2024; Guo et al., 2024a; Calandriello et al., 2024; Chen et al., 2024; Mitra et al., 2024) focus on online sampling and automated label/reward collection, reducing the manual cost required for alignment. Methods like (Ethayarajh et al., 2024; Hong et al., 2024) aim to reduce DPO's dependency on SFT warm-up by transforming its loss functions and data format into a SFT manner. These algorithms handle positive and negative samples using SFT objective and Unlikelihood Training (Welleck et al., 2019), respectively. Recently, (Zeng et al., 2024; Meng et al., 2024) improved the integration of the SFT and DPO by introducing various regularization terms. These terms prevent the policy model from overfitting DPO objective and deviating from SFT objective. However, the actual volume of training data is not decreased in these methods. Also, GPU-memory-consuming pair-wise data is still required, while the need for a reference model and preference-labeling for the entire answer trajectory is only eliminated in limited cases.

7 Conclusion

In this paper, we first interpret SFT and typical Preference Optimization methods into a unified framework using Preference Estimation and Transition Optimization. Through this modeling, we found the biased prior used in SFT is one of the main reasons why SFT performs worse than other Preference Optimization methods. Then, we introduce an efficient and effective method called Intuitive Fine-Tuning (IFT), which achieves alignment directly from the base model using non-preference-labeled data. Finally, experiments on widely used NLP benchmarks and Frozen Lake environment demonstrate the competitive performance of IFT.

8 Limitations

Our validation of IFT is limited to the fine-tuning setting, where data volume is constrained, leaving the scalability of IFT unexplored. Additionally, we primarily use Mistral-7B for baseline testing, and the generalization of IFT to larger and more diverse models requires exploration in the future.

References

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

Hritik Bansal, Ashima Suvarna, Gantavya Bhatt, Nanyun Peng, Kai-Wei Chang, and Aditya Grover. 2024. Comparing bad apples to good oranges: Aligning large language models via joint preference optimization. *arXiv preprint arXiv:2404.00530*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

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

Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, et al. 2024. Human alignment of large language models through online preference optimisation. *arXiv preprint arXiv:2403.08635*.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 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*.

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

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. *arXiv preprint arXiv:2310.01377*.

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*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

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

Farama. 2023. Frozen lake. https://gymnasium.farama.org/environments/toy_text/frozen_lake/. Accessed: 2024-05-19.

Yang Guan, Shengbo Eben Li, Jingliang Duan, Jie Li, Yangang Ren, Qi Sun, and Bo Cheng. 2021. Direct and indirect reinforcement learning. *International Journal of Intelligent Systems*, 36(8):4439–4467.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. 2024a. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Jiexin Wang, Huimin Chen, Bowen Sun, Ruobing Xie, Jie Zhou, Yankai Lin, et al. 2024b. Controllable preference optimization: Toward controllable multi-objective alignment. *arXiv preprint arXiv:2402.19085*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*.

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

Saeed Khaki, JinJin Li, Lan Ma, Liu Yang, and Prathap Ramachandra. 2024. Rs-dpo: A hybrid rejection sampling and direct preference optimization method for alignment of large language models. *arXiv preprint arXiv:2402.10038*.

Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. 2007. Reinforcement learning in continuous action spaces through sequential monte carlo methods. *Advances in neural information processing systems*, 20.

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

Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. 2018. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436.

Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. 2022. Trustworthy ai: From principles to practices. *ACM Comput. Surv.* Just Accepted.

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

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023a. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

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

Xiao Liu, Xixuan Song, Yuxiao Dong, and Jie Tang. 2024. Extensive self-contrast enables feedback-free language model alignment. *arXiv preprint arXiv:2404.00604*.

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

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. *arXiv preprint arXiv:2402.14830*.

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

Benjamin Plaut, Khanh Nguyen, and Tu Trinh. 2024. Softmax probabilities (mostly) predict large language model correctness on multiple-choice q&a. *arXiv preprint arXiv:2402.13213*.

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.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

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

Harm Seijen and Rich Sutton. 2014. True online td (lambda). In *International Conference on Machine Learning*, pages 692–700. PMLR.

Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. 2024. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*.

Polina Tsvilodub, Hening Wang, Sharon Grosch, and Michael Franke. 2024. Predictions from language models for multiple-choice tasks are not robust under variation of scoring methods. *arXiv preprint arXiv:2403.00998*.

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

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

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

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. 2023. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

Yueqin Yin, Zhendong Wang, Yi Gu, Hai Huang, Weizhu Chen, and Mingyuan Zhou. 2024. Relative preference optimization: Enhancing llm alignment through contrasting responses across identical and diverse prompts. *arXiv preprint arXiv:2402.10958*.

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

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

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

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.

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*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

Bowen Zhou and Ning Ding. 2024. Generative ai for complex scenarios: Language models are sequence processors. *International Journal of Artificial Intelligence and Robotics Research*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Theoretical Details

A.1 MDP In LMs

$\mathcal{M} = (S, A, \mathcal{T}, r, \rho_0)$:

- A, the concrete action space, consisting of N_A vocabularies as defined by the tokenizer.
- S, the concrete state space, comprising $N_S = (N_A)^N$ elements related to sequence length N. Each state represents a ordered permutation of vocabularies.
- ρ_0, the initial state of each generation, typically refers to the given instruction;
- $\mathcal{T} \in R^{N_S \times N_A}$, the state-action transition matrix of a given policy, indicating the probability of generating each token given different states;
- r, the reward assigned to a particular state-action pair.

A.2 Loss Function of IFT

The disparities of transition between model and human can be formalized as follows:

$$\mathcal{L}(\hat{T_\theta}; T^*) = \mathbb{E}_{\rho_0 \sim \mathcal{D}} \mathbb{E}_{s_n^* \sim \mathcal{S}_{\rho_0}^*} \left[-\sum_{n=0}^{N} \log \frac{\hat{T}_\theta(s_n^*, \rho_0)}{T^*(s_n^*, \rho_0)} \right] \tag{23}$$

We make the same hypothesis as SFT that the optimization objective of each target intermediate state has a probability equal to 1, so that $\forall n \in [0, N], \rho_0 \in \mathcal{D}, s_n^* \in \mathcal{S}_{\rho_0}^*$:

$$T^*(s_n^*, \rho_0) = 1 = T^*(s_N^*, \rho_0) \tag{24}$$

Thus, the objective of IFT can be represented directly by the following loss function:

$$\mathcal{L}(\hat{T_\theta}) = \mathbb{E}_{\rho_0 \sim \mathcal{D}} \mathbb{E}_{s_i^* \sim \mathcal{S}_{\rho_0}^*} \left[-\sum_{n=0}^{N} \log \mathcal{T}_\theta \left(\pi^* \left(\delta^*(s_i^*) \right), \delta_\theta(s_i^*) \right) \right] \tag{25}$$

As the optimal policy enjoys the optimal transition:

$$s_i^* = [s_{i-1}^*, a_i^*] = [s_{i-1}^*, \pi^*(s_{i-1}^*)] = \Pi^*(s_{i-1}^*) \tag{26}$$

Therefore, the disturbed optimal state keeps similar with the original optimal state:

$$\delta^*(s_i^*) = (1 - \lambda) s_i^* + \lambda \Pi^*(s_{i-1}^*) = s_i^* \tag{27}$$

Then, the final loss function can be presented as:

$$\mathcal{L}(\mathcal{T}_\theta, \delta_\theta) = \mathbb{E}_{\rho_0 \sim \mathcal{D}} \mathbb{E}_{s_i^* \sim \mathcal{S}_{\rho_0}^*} \left[-\sum_{n=0}^{N} \log \mathcal{T}_\theta(a_i^*, \delta_\theta(s_i^*)) \right] \tag{28}$$

A.3 Proof for Bellman Equation

Considering only one sampled state s_n^* constrained by ρ_0 in the datasets, we have:

$$\begin{aligned} &\exp\big(-\mathcal{L}(\hat{T}_\theta(s_n^*, \rho_0))\big) \\ &= \mathcal{T}_\theta(a_n^*, \delta_\theta(s_n^*)) \Big(\sum_{n+1}^{N} \mathcal{T}_\theta(a_i^*, \delta_\theta(s_i^*)) \Big) \\ &= \max_a \Big[\mathcal{T}_\theta(a, s_n^*) \big(r + \gamma V(\hat{s_{n+1}^\theta}) \big) \Big] \\ &= V_\theta(\hat{s_n^\theta}) \end{aligned} \tag{29}$$

where $r = (1-\gamma) V(\hat{s_{n+1}^\theta})$. This reward function implicitly accounts for the influence of the current prediction on future generations.

A.4 Reformulation of Typical Methods

We reformulate the loss function of some methods using the disparities of transition matrices as:

SFT

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{\rho_0 \sim \mathcal{D}} \mathbb{E}_{s_i^* \sim \mathcal{S}_{\rho_0}^*} \left[-\sum_{i=0}^{N} \log \mathcal{T}_\theta(\pi^*(s_i^*), s_i^*) \right] \tag{30}$$

where the human's preference is unbiasedly estimated, but the model's preference is inaccurately represented by s_i^*.

PPO

$$\mathcal{L}_{\text{PPO}} = \mathbb{E}_{\rho_0 \sim \mathcal{D}} \mathbb{E}_{s_i^* \sim \mathcal{S}_{\rho_0}^*} \left[-\sum_{i=0}^{N} \mathcal{R}(\pi_\theta(s_i^\theta), s_i^\theta) \right] \tag{31}$$

where $\mathcal{R} \in (-\infty, 0]$ denotes the degree of closeness between human preferences and the state-action pairs chosen by model. The reward and loss will be zero only if the state-action pairs perfectly align with human preferences. Thus, PPO implicitly models the human policy π^* through reward modeling, which can be formulated as follows:

$$\mathcal{R} = \pi_{\mathcal{R}} \leftarrow \min_\pi \mathcal{L}_{\mathcal{R}} \tag{32}$$

$$\mathcal{L}_{\mathcal{R}} = \mathbb{E}_{\rho_0 \sim \mathcal{D}} \mathbb{E}_{s_i^+ \sim \mathcal{S}_{\rho_0}^+, s_i^- \sim \mathcal{S}_{\rho_0}^-} \left[-\log \sigma \Big(\sum_{i=0}^{N} \log \mathcal{T}_\mathcal{R}(\pi^+(s_i^+)|s_i^+) - \sum_{i=0}^{N} \log \mathcal{T}_\mathcal{R}(\pi^-(s_i^-)|s_i^-) \Big) \right] \tag{33}$$

DPO-Online

$$\mathcal{L}_{\text{DPO}} = \mathbb{E}_{\rho_0 \sim \mathcal{D}} \mathbb{E}_{s_i^* \sim \mathcal{S}_{\rho_0}^*, s_i^\theta \sim \mathcal{S}_{\rho_0}^\theta} \left[-\log \sigma \Big(\sum_{i=0}^{N} \log \mathcal{T}_\theta(\pi^*(s_i^*), s_i^*) - \sum_{i=0}^{N} \log \mathcal{T}_\theta(\pi_\theta(s_i^\theta), s_i^\theta) \Big) \right] \tag{34}$$

Ideally, this loss function increases the probabilities of state-action pairs preferred by humans and decreases the probabilities of those chosen by the model. It unbiasedly estimate both the human's and model's preference.

DPO-Offline

$$\mathcal{L}_{\text{DPO}} = \mathbb{E}_{\rho_0 \sim \mathcal{D}} \mathbb{E}_{s_i^+ \sim \mathcal{S}_{\rho_0}^+, s_i^- \sim \mathcal{S}_{\rho_0}^-} \left[-\log \sigma \Big(\sum_{i=0}^{N} \log \mathcal{T}_\theta(\pi^+(s_i^+), s_i^+) - \sum_{i=0}^{N} \log \mathcal{T}_\theta(\pi^-(s_i^-), s_i^-) \Big) \right] \tag{35}$$

In the offline circumstance, the positive samples can still represent the human preference correctly, as s^+ is usually similar to s^*. However, this is not the case for negative samples.As training progresses, s^- becomes more and more out-of-distributions compared to the model's preferred state s^θ, leading to biased estimations.

IFT

$$\mathcal{L}_{\text{IFT}} = \mathbb{E}_{\rho_0 \sim \mathcal{D}} \mathbb{E}_{s_i^* \sim \mathcal{S}_{\rho_0}^*} \left[-\sum_{n=0}^{N} \sum_{i=n}^{N} \log \mathcal{T}_\theta(a_i^*, \delta_\theta(s_i^*)) \right] \tag{36}$$

$$\delta_\theta(s_i^*) = (1-\lambda) s_i^* + \lambda \pi_\theta(s_{i-1}^*) \tag{37}$$

By using a model-based disturbance function, IFT constructs a residual connection in the temporal dimension, providing a better estimation for the model than SFT. Through this approach, IFT implicitly implements a Relation Propagation in the

Transition Optimization stage, which considers the influence of current predictions on future outcomes. This propagation also reduces the influence of bias introduced by inaccurate estimations in earlier positions.

B Implementation Details

B.1 NLP Settings

For the coefficient β in DPO, TDPO, ORPO and SimPO, we use 0.1, 0.1, 0.25 and 2.0 respectively, as presented in their original papers. For the coefficient γ/β ration in SimPO, we use 0.8 to keep the same setting in its original papers. For IFT, we choose 0.2 for λ and incorporate a decay factor of 0.95 to fitting better with the Bellman Equation. We save checkpoints every 20k steps and select the results from the checkpoint with the best average score to demonstrate the performance of each method.

Name	Value
epoch	3
mini batch size	8
gradient accumulation step	64
warmup ratio	0.1
scheduler	cosine
learning rate	5e-7
optimizer	RMSprop
precision	bfloat16

Table 4: Hyper-Parameters in NLP Setting

We implement our main experiments on four NVIDIA A6000 GPUs. When using 60k single-target data, the entire training process for SFT and IFT takes approximately 20 hours, with each epoch lasting 7 hours. When using 60k pair-wise data, the training process for DPO and ORPO takes around 40 hours and 30 hours respectively, due to the differences in requirements for a reference model.

B.2 Frozen Lake Setting

We keep the similar hyper-parameters as in NLP setting for Frozen Lake game, running this environment on CPUs. Since our designed environment includes an optimal and a sub-optimal trajectory, we select the optimal trajectory as the target for SFT and IFT. For DPO and ORPO, the optimal and sub-optimal trajectories are used as positive and negative samples, respectively.

B.3 Pseudo-code of IFT

Algorithm 1 Intuitive Fine-Tuning

Input:
Initial instruction ρ_0
Ground truth s^* with N tokens: $s^*[1], \ldots, s^*[N]$

Step 1: Inference One Step Ahead
for t in $[1, N]$ **do**
  Predict the probability distribution of the t-th token: $P'_t = \pi_\theta(s^*[0:t-1])$
  Sample the predicted token: $s^\theta[t] = \arg\max P'_t$
end for

Step 2: Intuitive Preference Estimation
Encode s^* and s^θ using the Embedding Layer E
Compute the fused embedding: $e = (1-\lambda)E(s^*) + \lambda E(s^\theta)$
for t in $[1, N]$ **do**
  Predict the probability distribution of the t-th token: $P''_t = (\pi_\theta/E)(e[0:t-1])$
  Compute the token-level loss: $\mathcal{L}_t = \log(P''_t, s^*[t])$
end for

Step 3: Dynamic Relation Propagation
for t in $[1, N]$ **do**
  Compute the cumsum weight similar to Bellman Equation: $w_t = \sum_{i=t}^{N} \alpha^{N-t}\mathcal{L}_i$
end for

Output: Final loss $\mathcal{L}_{\text{IFT}} = w \cdot \mathcal{L}$

Method	Reference	Data		Alpaca-Eval		Alpaca-Eval-2	
		pairwise	volume	win-rate	lc win-rate	win-rate	lc win-rate
Gemma-2B	–	–	–	–	–	–	–
+ SFT	✗	✗	120k	36.53	30.28	0.99	0.57
+ DPO	✓	✓	120k	3.13	1.18	0.13	0.23
+ TDPO	✓	✓	120k	2.14	0.70	0.25	0.10
+ ORPO	✗	✓	120k	36.62	34.23	1.12	0.59
+ SimPO	✗	✓	120k	4.48	2.42	0.13	0.15
+ IFT	✗	✗	120k	36.74	39.33	1.61	1.23

Table 5: Evaluation on LLM-based Benchmarks when fine-tuning with UltraFeedback-60k.

Method	ARC	ARC-Gen	MMLU	TruthfulQA	WinoGrande	GSM8K	Avg.
Gemma-2B	42.75	43.17	35.68	35.25	66.46	16.98	39.42
+ SFT	<u>42.74</u>	**42.66**	**36.23**	<u>51.65</u>	65.90	<u>20.66</u>	43.44
+ DPO	41.38	40.96	35.84	31.82	65.11	19.26	38.68
+ TDPO	41.12	42.06	35.86	33.05	65.19	18.80	38.80
+ ORPO	42.66	42.24	<u>36.10</u>	**52.08**	<u>66.46</u>	20.55	<u>43.57</u>
+ SimPO	42.32	40.78	35.81	29.50	65.59	19.26	38.50
+ IFT	**43.26**	<u>42.41</u>	35.80	51.16	**66.53**	**23.05**	**43.96**

Table 6: Evaluation on Open-LLM Leaderboard when fine-tuning with UltraFeedback-60k.