

Implicit Reward as the Bridge: A Unified View of SFT and DPO Connections

Bo Wang^{1*} Qinyuan Cheng^{1*} Runyu Peng^{1*} Rong Bao¹ Peiji Li¹
 Qipeng Guo² Linyang Li² Zhiyuan Zeng¹ Yunhua Zhou² Xipeng Qiu^{1†}
¹School of Computer Science, Fudan University ²Shanghai Artificial Intelligence Laboratory
 {bwang22, chengqy21, rypeng22, rbao22}@m.fudan.edu.cn
 {linyanglei19, xpqiui}@fudan.edu.cn
 {guoqipeng, zhouyunhua}@pjlab.org.cn

Abstract

Post-training processes are essential phases in grounding pre-trained language models to real-world tasks, with learning from demonstrations or preference signals playing a crucial role in this adaptation. We present a unified theoretical framework bridging Supervised Fine-Tuning (SFT) and preference learning in Large Language Model (LLM) post-training. Through rigorous mathematical derivation, we demonstrate that both SFT and preference learning methods like Direct Preference Optimization (DPO) operate within the same optimal policy-reward subspace, with SFT representing a special case of implicit reward learning. Our analysis reveals a critical limitation in conventional SFT: the KL divergence term in distribution matching becomes constant with respect to the policy during optimization, failing to constrain model updates. To address this, we propose a simple yet effective learning rate reduction approach that yields significant performance improvements (up to **25%** relative gain and **6%** absolute win rate increase in instruction following tasks). Additionally, we derive alternative SFT objectives from various f-divergence functions that preserve the KL term during optimization, further enhancing post-DPO model performance. Finally, we extend the theoretical relationship between LLM logits and Q-functions from preference learning to the SFT context, providing mathematical derivations and experimental validation.

1 Introduction

Post-training represents a critical phase in grounding Large Language Models (LLMs) in real-world applications. After accumulating general prior knowledge from numerous pre-training corpora, post-training aims to leverage the potential of LLMs for different needs, such as following natural language instructions [1, 2, 3, 4, 5]. Two principal methodological approaches dominate the post-training landscape. The first approach learns from expert demonstrations [6, 7], commonly known as imitation learning, which in the context of LLMs is typically referred to as Supervised Fine-Tuning (SFT). The second approach focuses on learning from environmental signals, primarily through Reinforcement Learning methods [8, 9, 10].

Within the post-training landscape, preference signals have emerged as particularly valuable forms of feedback, attracting substantial research attention [11, 1]. Preference learning typically follows a two-stage process (hereafter referred to as **sequential training**): an initial stage of SFT followed by preference optimization methods like Direct Preference Optimization (DPO) [12]. However, the relationship between these critical stages remains predominantly understood through empirical

*Equal Contribution.

†Corresponding author

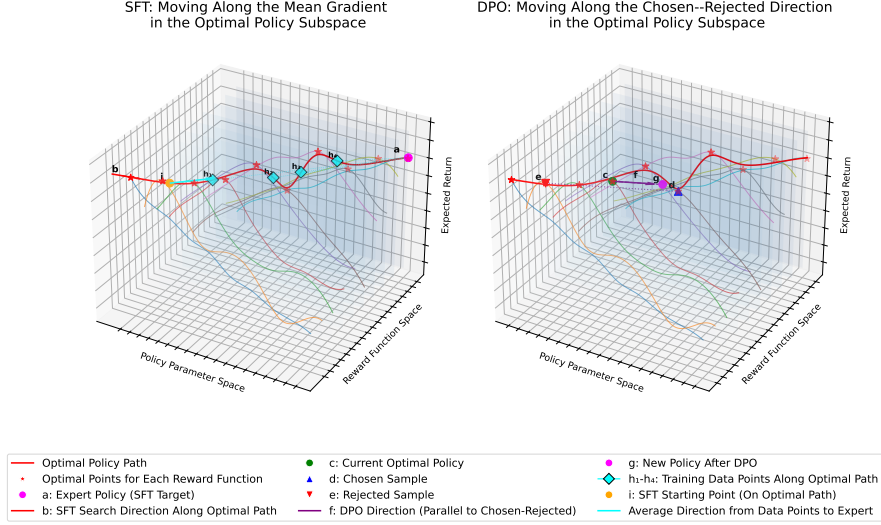


Figure 1: Schematic diagram: SFT and DPO are optimizing implicit rewards in the optimal policy subspace. The x -dimension refers to the space of all possible policy models. The y -dimension represents the space of all possible reward functions. Given a specific reward function, each possible policy will have its own expected return, forming a curve in the policy-return plane. The optimal policies corresponding to different reward functions constitute a subspace. **Left:** The SFT process searches within this subspace and moves along the average direction indicated by the demonstrations. **Right:** Likewise, DPO operates within this same subspace, but navigates along the direction vector formed by the chosen and rejected examples.

observations rather than theoretical foundations, and SFT is often treated merely as a preparatory warm-up step [13]. Despite the widespread adoption of this sequential paradigm, a significant gap persists in theoretical perspectives regarding how these two approaches fundamentally relate to one another. While previous research [14] has extensively explored various aspects of LLM learning dynamics, the theoretical connections between SFT objectives and preference learning frameworks have received insufficient attention, limiting our understanding of their combined effectiveness in the post-training process.

To mitigate this gap, we prove that implicit reward learning can be utilized as a unified view connecting SFT and preference-learning processes. Previous work [12] established that preference learning in the second stage can operate through implicit rewards. In our research, we revisit the distribution matching objective and apply necessary adjustments for post-training. We provide a comprehensive mathematical proof following earlier works [15, 16]. Our proof demonstrates that the conventional SFT objective represents a special case of learning implicit rewards. Figure 4 illustrates our theoretical conclusion. The optimal policy for each possible reward function forms a policy-reward subspace, with both SFT and DPO operating within this subspace.

This theoretical framework of implicit rewards yields several novel insights. By framing SFT as a training target derived from distribution matching, we uncover a key insight: the crucial KL term in the objective functions merely serves as a zero-order component. Since this term remains constant with respect to π , it imposes no constraints on model updates following differentiation. We propose a simple yet efficient heuristic to mitigate this issue by reducing the learning rate. Furthermore, we identify alternative training objectives by choosing different f -divergence derivative functions for distribution matching that preserve the KL term during optimization and show their effectiveness empirically. Finally, we demonstrate that LLM logits can function as a Q-function corresponding to implicit rewards during the SFT process. This extends the theoretical framework in [17], which primarily established this relationship in the DPO setting, while our work reveals similar mathematical structures within the SFT process. Our empirical training results align strongly with these theoretical predictions in the instruction-following tasks.

Our main contributions are as follows:

1. We revisit the distribution matching objective and mathematically prove that SFT also learns an implicit reward function identical to that of DPO. This provides a unified theoretical view that clarifies the relationship between SFT and preference learning.
2. Within this theoretical framework, we provide a simple yet effective approach by reducing the learning rate during the SFT phase to mitigate the absence of the KL term. The KL term typically ensures that the policy does not deviate excessively from the base model, promoting stable and efficient learning. This significantly improves results, with relative improvements up to 25% (absolute win rate increases of up to 6%, from 15.6% to 21.5%).
3. We also propose several alternative SFT objectives derived from other f -divergence functions for LLMs. We demonstrate further improvements in model performance after DPO training (hereafter referred to as **post-DPO**), which yields up to 4% absolute win rate improvements.
4. We mathematically extend the relationship between LLM logits and Q-functions from the DPO context to the SFT process, supporting this extension with indirect experimental evidence. This formulation enables us to efficiently estimate state values under the model’s implicit reward and provides deeper insights into the role of SFT in the alignment process.

2 Related Work

2.1 Inverse Reinforcement Learning

First formalized by [18], Inverse Reinforcement Learning assumes that the expert represents the optimal policy under a certain reward function. We provide a detailed introduction in Appendix A.

Instead of using the Bellman equation like [15, 16], we apply closed-form solutions and avoid inner-loop optimization. In comparison with [19], our work utilizes a more general f -divergence formulation [20] and does not introduce negative samples. Unlike [21], we primarily examine the relationship between SFT and DPO in the sequential training context of large language models.

2.2 Implicit Reward in LLMs

The typical Reinforcement Learning from Human Feedback (RLHF) methods generally involve substantial computational budgets (e.g., PPO [22] uses four models in the training process), and DPO [12] was proposed to reduce the computational overhead of the RL process. It proved that the maximization problem has a closed-form solution, allowing us to perform reward modeling directly on the implicit reward. Many related works like IPO [23], KTO [24], SimPO [25], and R-DPO [26] have continued development along this path. [17] has also established the relationship between Q-functions and LLM-logits during the DPO process. [27] treats logits as Q-values and trains a Q-head on robotic tasks. We demonstrate that analogous structures exist in the SFT process.

2.3 Post-Training Theory Analysis for LLMs

There are also several analyses examining relationships in post-training phases. [28] analyzes the isomorphic relationship between implicit rewards and reward models. It connects implicit rewards with the generation-verification gap. [14] pays more attention to the learning dynamics during different training phases. In our work, we focus more on the relationship between training objectives during the post-training process and use implicit rewards as a medium to understand them.

3 Unified View between SFT and DPO

Token-Level MDP in LLM In language models, the Markov Decision Process (MDP) applies to token-level decisions. The state $s \in S$ represents the context, actions $a \in A$ are possible next tokens, and the policy $P(a|s)$ gives token probabilities. State transitions $T(s'|s, a)$ are deterministic, with $s' = s \oplus a$ (token appended to context). The model receives reward $r(s, a)$ for each choice, continuing until reaching a terminal state. γ is the discount factor. This framework formalizes how language models make sequential decisions during text generation.

Definition 1 (Occupancy Measure [15]). *The occupancy measure of a policy π is defined as:*

$$\rho(s) = (1 - \gamma) \sum_{i=0}^{\infty} \gamma^i P(s_i = s | \pi),$$

where $P(s_i = s \mid \pi)$ denotes the probability of visiting state s at time step i under policy π .

Definition 2 (State-Action Distribution [15]). *The state-action distribution of a policy π is given by:*

$$\mu(s, a) = \pi(a \mid s) \rho(s),$$

where $\mu(s, a)$ represents the stationary distribution over state-action pairs induced by π .

3.1 Distribution Matching in Post-training

A well-established training objective in imitation learning is **distribution matching** [29, 15]. It focuses on minimizing the f -divergence between the expert’s state-action distribution μ_E and that of the policy model μ_π , while incorporating an entropy regularization term to promote exploration. However, entropy is not entirely suitable in the post-training scenario for LLMs. The use of probabilistic averaging over the whole vocabulary could potentially damage the natural language priors established during the pretraining of the base model. Therefore, we modify the regularization term from entropy to the Kullback-Leibler (KL) divergence between the base model π_{ref} and the policy model π .

$$\min_{\pi} D_f(\mu_\pi \parallel \mu_E) + \underbrace{\beta D_{KL}(\pi \parallel \pi_{ref})}_{-\beta \mathcal{H}(\pi) \text{ in traditional setting}}, \quad (1)$$

where $D_f(\cdot \parallel \cdot)$ denotes the f -divergence, and $\mathcal{H}(\pi)$ is the entropy of the policy. β is the coefficient of the regularization term and often serves as a hyperparameter.

A similar object was also introduced in [28]. We approach this concept from a different theoretical perspective and provide additional clarification here:

- 1) **From the Cross-Entropy Term Perspective:** The KL divergence can be split into the entropy term and the cross-entropy term: $\mathcal{H}(\pi, \pi_{ref}) - \mathcal{H}(\pi)$. As the base model has converged during the pretraining phase and has been exposed to extensive natural language data, the cross-entropy term should not be large. This implies that the policy model should not deviate from the domain of natural language when maximizing exploration.
- 2) **From a KL Divergence Perspective:** The base model obtained from the pretraining phase already possesses sufficient quality and contains additional knowledge. Therefore, when minimizing the divergence for distribution matching, we need to preserve the intrinsic properties of the base model.

3.2 Imitation Learning as Implicit Reward Discovery

Following the derivation process of non-adversarial imitation learning [15], the training objective can be expressed as an equivalent min-max problem. We have the following key result:

Theorem 1 (Equivalent Objective for Distribution Matching). *Learning a policy that minimizes the f -divergence between expert and policy state-action distributions is equivalent to **first learning an optimal policy under an arbitrary reward function, then optimizing a function of that reward function**:*

$$-\min_r [\mathbb{E}_{\mu_E}[f^*(-r)]] + \underbrace{\max_{\pi} \mathbb{E}_{\mu_\pi}[r] - \beta D_{KL}(\pi \parallel \pi_{ref})}_{\text{Has closed-form solution}}, \quad (2)$$

where f^* is the convex conjugate function corresponding to the chosen f -divergence, μ_π is the state-action distribution of the policy being learned, and μ_E is the expert’s state-action distribution. r is the independent variable of f^* and is commonly interpreted as the reward function.

We provide a detailed proof in Appendix B.1. Here, the reward function r is not yet related to implicit rewards but rather represents an arbitrary function. The commonly used SFT loss still differs from distribution matching approaches in fundamental ways. However, **this formulation establishes a connection between finding a policy model and identifying a suitable reward function**. Although the equivalent objective is formulated as a bi-level optimization problem, the latter part, which we denote as $J(\pi) = \mathbb{E}_{\mu_\pi}[r] - \beta D_{KL}(\pi \parallel \pi_{ref})$, has a closed-form solution as demonstrated by [17]. We leverage this established result in our approach.

$$\pi^* = \arg \max_{\pi} J(\pi), \quad J(\pi^*) = V^*(s_0) \quad (3)$$

where $\pi^*(a \mid s)$ is the optimal policy. V^* is the value function of the optimal policy.

Lemma 1 (Relationship between Reward and Policy[17], **Implicit Reward**). *The relationship between reward and corresponding optimal policy is :*

$$r(x, y) = \beta \log \frac{\pi^*(y | x)}{\pi_{\text{ref}}(y | x)} + V^*(s_0) - V^*(s_t), \quad (4)$$

where $r(x, y)$ represents the reward for the LLM’s input-output pair (x, y) .

As the reward r in the training objective eq. (2) can be initialized arbitrarily, we can have the following assumption.

Assumption 1 (Initial Reward Simplification). *Without loss of generality, the initial reward r can be treated as $V_\pi(s_0)$ with $V_\pi(s_t) = 0$ for all $t > 0$. Under this assumption, the initial policy π is the optimal policy with respect to the initial reward.*

Now we can substitute the latter part of eq. (2) with its closed-form solution. The correspondence between divergence measures and their conjugate functions has been established in previous work [20]. We also list them in Appendix C. We select the total variation distance as our divergence measure, for which the corresponding conjugate function is simply the identity function. Additionally, the relationship between the reward and policy π satisfies eq. (4). This allows us to directly obtain the final objective, which takes the familiar form of SFT:

$$\max_{\pi} \mathbb{E}_{\mu_E} [\underbrace{\beta \log \pi(y|x)}_{\text{MLE}} - \underbrace{\log \pi_{\text{ref}}(y|x) - V_\pi(s_t)}_{\text{Serve as Constants}}], \quad (5)$$

where s_t represents the terminal state. The expected return after this state becomes a constant.

Conclusion 1: Commonly used SFT is a special case of finding implicit reward, same as DPO. We now derive that the commonly used SFT loss constitutes a *special case* of reward discovery through imitation learning when total variation is selected as the f -divergence measure. Appendix C presents alternative training targets derived from different f -divergence functions. Since the derivation process above is reversible, we conclude that the **SFT process searches along the optimal policy-reward subspace, attempting to model the reward implicitly embedded in expert demonstrations**. At the start point of the SFT process, the policy is the optimal policy under Assumption 1. During the optimization process, the relationship between the model and reward continues to satisfy eq. (4), resulting in searching the optimal subspace of policy-reward. This implicit reward structure aligns perfectly with that in DPO, offering a harmonious theoretical view that unifies both approaches.

Conclusion 2: KL term absent in commonly used SFT. The difference in eq. (5) between the reference model and policy model takes a zero-order form, which acts as a constant when performing stochastic gradient descent. However, it constrains the update step size of the policy model and plays an important role in most RL algorithms. **The absence of this term leads to a substantial distance between the post-DPO model’s training starting point and the base model.** We propose a simple but effective method to mitigate this limitation. With smaller learning rates to reduce the optimization step size, we show significant performance improvements in the instruction following domain in Section 4.2. Furthermore, by selecting different f -divergences, we can derive objectives similar to SFT while preserving the KL term. Most of these involve logarithmic and exponential operations that may lead to numerical instability. We select three representative divergences and present their comparative results in Section 4.3.

3.3 Model Maintains Intrinsic Expected Return Estimation During SFT

In the DPO process, [17] noted that the logits of LLMs can be interpreted as a Q-function under mild assumptions. We extend this conclusion based on similar structures in eq (2).

Theorem 2 (Intrinsic Expected Return). *During the SFT process, the logits l_a of a language model correspond to the Q-function $Q(s, a)$ of the learned implicit reward:*

$$l_a = Q_{\hat{r}}(s, a) + C(s) = \hat{r}(s, a) + \gamma \mathbb{E}_{s_{t+1} \sim P(\cdot|s,a)} [V_{\hat{r}}(s_{t+1})], \quad (6)$$

where \hat{r} is the model’s implicit reward function satisfying eq. (4), γ is the discount factor, and $V_{\hat{r}}(s_{t+1})$ is the value function of the next state. $C(s)$ is a function conditioned only on the state.

We provide a detailed proof in Appendix B.3. Our findings indicate that not only in the DPO process but also in SFT, the model’s logits can be interpreted as a Q-function characterizing the model’s estimation of expected returns. These returns are calculated based on **the implicit reward learned by the model itself**. The function $C(s)$ represents the gap between the true Q value and the logits, but it does not affect the relative ranking among different actions since it depends only on the current state and acts as a constant across actions. The value function can be calculated using log-sum-exp according to Appendix B, and we hypothesize that:

Assumption 2 (Value-Dominance Assumption). *For most two states s_1 and s_2 , the difference between $C(s_1)$ and $C(s_2)$ is smaller than the difference between $V(s_1)$ and $V(s_2)$.*

This conclusion allows us to use the log-sum-exp of logits from the LLM as a Value function, instead of performing Monte Carlo sampling when utilizing other divergence formats.

4 Empirical Study

In this section, we provide empirical analysis on the instruction-following task. Our detailed discussion is presented as follows.

- **Small learning rate during SFT process can yield significant benefits for post-DPO models.** We prove that reducing the learning rate to decrease the single-step optimization stride for SFT during sequential training improves results, as demonstrated in Section 4.2.
- **Alternative f -divergences that preserve KL terms also lead to better results.** Training targets derived from other f -divergence do not suffer from losing the KL term. We select Pearson χ^2 and Squared Hellinger, which avoid numerical stability issues associated with logarithmic and exponential functions, to demonstrate these improvements in Section 4.3.
- **LLM logits exhibit value function properties, evaluating state quality similarly.** By leveraging the characteristic of value functions to reflect expected state quality, we demonstrate that different models maintain similar judgments across states in Section 4.4.
- **SFT mitigates initial reward randomness and quickly aligns implicit rewards to reasonable values.** We explain the role of SFT in post-training as correcting the initial reward Assumption 1. We demonstrate that $V(s_0)$ converges rapidly during the SFT process and present the corresponding empirical results in Section 4.5.

4.1 Basic Experiment Setting

Model and Dataset Selection General instruction following is a fundamental capability of large language models required for most downstream tasks. Following the setting of SimPO [25], we select Llama3-8B [2] and Mistral-7B [5] as our base models. UltraChat-200K [30] is a commonly used SFT dataset. For general instruction-following tasks, models typically complete SFT training on UltraChat-200K before performing DPO on Ultra-feedback [31] to obtain the final model. We use these two datasets in our experiments.

Hyperparameters, Device, Baselines, and Evaluation Benchmarks The most commonly used learning rates during post-training are $2e-5$ for SFT and $5e-7$ for DPO. For all of our experiments, we train with a batch size of 128 on $8 \times H100$ GPUs using the OpenRLHF [32] framework. For the DPO training process, β is an important hyperparameter, and we set $\beta = 0.01$. We evaluate our models using AlpacaEval2 [33], Arena Hard [34], and MT-bench [35]. Since these benchmarks can be influenced by many implementation details, such as the vLLM [36] version, we maintain consistent implementation versions with SimPO. We use the same decoding parameters as SimPO during downstream evaluation. As evaluating all three benchmarks simultaneously would incur significant API costs, in some experiments, we used AlpacaEval2 as the representative benchmark.

4.2 Small Learning Rate SFT Leads to Better post-DPO Results

As mentioned in Section 3.2, the KL term, i.e., $\log \pi_{\text{ref}}$, that constrains the SFT learning process is a zero-order term and provides no gradient contribution to policy optimization after differentiation. Considering the importance of step size constraints in traditional RL, we infer that the learning rate for SFT should be reduced to decrease the effective update magnitude. Compared with the commonly

Table 1: Downstream results for the smaller learning rate setting. The reference-only result is reported by SimPO[25]. We reproduce the SFT, DPO, and SimPO results using publicly available checkpoints. The models trained with a smaller learning rate for SFT and subsequent models fine-tuned from this SFT checkpoint are marked in blue.

Method	Llama-3-Base (8B)					Mistral-Base (7B)				
	AlpacaEval 2		Arena-Hard	MT-Bench		AlpacaEval 2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4
<i>Reference Only. Not Compared Directly</i>										
RRHF[37]	11.6	10.2	5.8	5.4	6.7	12.1	10.1	6.3	5.8	7.0
SLiC-HF[38]	10.9	8.9	7.3	5.8	7.4	12.3	13.7	6.0	6.3	7.6
CPO[39]	9.8	8.9	6.9	5.4	6.8	10.8	8.1	5.8	6.0	7.4
IPO[23]	11.8	9.4	7.5	5.5	7.2	14.4	14.2	17.8	6.5	7.4
KTO[24]	13.1	9.1	5.6	5.4	7.0	14.2	12.4	12.5	6.3	7.8
ORPO[40]	14.7	12.2	7.0	5.8	7.3	12.2	10.6	10.8	6.1	7.6
R-DPO[26]	17.4	12.8	8.0	5.9	7.4	17.6	14.4	17.2	6.6	7.5
<i>Learning Rate Optimization For Supervised Fine-tuning</i>										
SFT	5.8	3.7	2.7	5.9	6.8	5.7	3.6	1.6	5.4	6.1
+ DPO[12]	17.3	14.2	19.7	6.8	7.3	15.6	12.5	11.7	6.3	6.4
+ SimPO[25]	23.5	21.3	30.3	7.0	7.3	24.1	22.9	22.6	6.5	6.8
SFT (smaller lr)	6.3 _{+0.5}	4.3 _{+0.6}	3.3 _{+0.6}	5.9	6.3	6.7 _{+1.0}	3.7 _{+0.1}	3.1 _{+1.5}	5.7 _{+0.3}	6.4 _{+0.3}
+ DPO[12]	19.4 _{+2.1}	16.2 _{+2.0}	21.1 _{+1.4}	7.0 _{+0.2}	7.4 _{+0.1}	21.5 _{+5.9}	16.7 _{+4.2}	21.6 _{+9.9}	6.5 _{+0.2}	7.0 _{+0.6}
+ SimPO[25]	28.5 _{+5.0}	25.0 _{+3.7}	34.3 _{+4.0}	6.6	7.3	27.3 _{+3.2}	24.0 _{+1.1}	14.9	6.3	6.8

used 2×10^{-5} , we implement smaller learning rates of 5×10^{-6} for Llama3-Base and 1×10^{-6} for Mistral during the SFT process. For the RL process following SFT, we select DPO and SimPO algorithms while maintaining the same hyperparameters as in the 2×10^{-5} configuration. We utilize the publicly released checkpoints from the original SimPO implementation and evaluate them in the same testing environment to establish our baseline results.

Main Results Results are presented in Table 1. Our reproduced baseline results outperform the results reported in the original SimPO paper. We maintain identical settings to ensure fair comparison. It can be observed that reducing the learning rate leads to moderate improvements for SFT checkpoints and significant enhancements after applying alignment algorithms. The SimPO results show relative improvements of **20%** (absolute improvement of 5%) for Llama3-8B and **25%** (absolute improvement of 6%) for Mistral after applying DPO. As we maintain identical hyperparameters in the DPO training process, the performance improvements primarily derive from adjusting the learning rate during the SFT phase, which confirms our hypothesis.

4.3 Other Forms of Imitation Loss Behave Better in Sequential Training

Beyond the Total Variance divergence, other f -divergence functions shown in Appendix C yield training targets where the KL term is not limited to zero-order approximations. However, many alternatives involve logarithmic or exponential calculations, or even composite log-exp operations (as in Jensen-Shannon divergence), which can lead to numerical instability. We select two additional f -divergence formulations, which are Pearson χ^2 and Squared Hellinger. Their derived training targets are presented in Table 2.

For Pearson χ^2 , there exists a squared probability difference term that acts as a KL constraint. For Squared Hellinger, a coefficient term related to probability differences is multiplied before the classic gradient term after applying the chain rule, which modulates the update step size. We compare these results with our previously obtained Total Variance results.

Main result The results are presented in Table 3. It can be observed that both the Pearson χ^2 and Squared Hellinger lead to weaker SFT but better results after DPO, regardless of whether we use Mistral or Llama. We can reach an interesting conclusion that a better SFT checkpoint doesn't necessarily lead to better DPO results. These improvements from KL-regularized SFT validate our theory, showing the importance of the KL term during post-training. The training loss curves for these three SFT approaches are shown in Figure 2c.

Table 2: We list three optional f -divergences and their corresponding training targets. We use ΔV as an abbreviation for $V_\pi(s_0) - V_\pi(s_t)$ for simplicity of notation. See Appendix C for the complete table.

Name	$D_f(P\ Q)$	Conjugate $f^*(t)$	Training Target $\max_\pi \mathbb{E}_{\mu_E}[\cdot]$
Total variation	$\frac{1}{2} \int p(x) - q(x) dx$	t	$\beta \log \pi(y x) - \log \pi_{\text{ref}}(y x) - V_\pi(s_t)$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)} dx$	$\frac{1}{4}t^2 + t$	$-\frac{1}{4}(\log \frac{\pi(y x)}{\pi_{\text{ref}}(y x)} + \Delta V)^2 + \log \frac{\pi(y x)}{\pi_{\text{ref}}(y x)} + \Delta V$
Squared Hellinger	$\int (\sqrt{p(x)} - \sqrt{q(x)})^2 dx$	$\frac{t}{1-t}$	$1 - \frac{1}{1 + \log \frac{\pi(y x)}{\pi_{\text{ref}}(y x)} + \Delta V}$

Table 3: Downstream results for different training targets and their corresponding post-DPO checkpoints. SFT refers to the commonly used training target derived from total variation. Pearson-SFT refers to the imitation objective derived from Pearson- χ^2 divergence. SH-SFT refers to the objective derived from Squared Hellinger divergence.

Method	Llama-3-Base (8B)					Mistral-Base (7B)				
	AlpacaEval 2		Arena-Hard	MT-Bench		AlpacaEval 2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4
<i>Traditional SFT</i>										
SFT	6.3	4.3	3.3	5.9	6.3	6.7	3.7	3.1	5.7	6.4
+ DPO	19.4	16.2	21.1	7.0	7.4	21.5	16.7	21.6	6.5	7.0
<i>Other Format of SFT</i>										
Pearson-SFT	5.1	3.7	3.3	5.8	6.1	6.2	3.4	3.5	6	6.5
+ DPO	20.1 _{+0.7}	17.7 _{+1.5}	24.7 _{+3.6}	7.1 _{+0.1}	7.2	23.1 _{+1.6}	19.0 _{+2.3}	21.8 _{+0.2}	6.8 _{+0.3}	6.7
SH-SFT	4.8	3.7	17.3	5.7	6	6.5	3.5	3.1	5.7	6.1
+ DPO	19.6 _{+0.2}	17.3 _{+1.1}	19.9	6.9	7.2	23.6 _{+2.1}	20.9 _{+4.2}	22 _{+0.4}	6.8 _{+0.3}	6

4.4 Value and Reward in SFT

With the theoretical conclusions presented in Section 3.3, we estimate the value function using the logits of LLMs. Traditional value estimation typically involves Monte Carlo sampling. Moreover, values are conventionally calculated using ground truth rewards, which in our context are implicit and not directly accessible. We aim to provide empirical evidence that LLM logits exhibit properties of value functions: their scores can be used to evaluate state quality.

More precisely, we demonstrate that for LLMs trained on the same domain, the evaluations for different states maintain similar rankings across models. We divide the UltraChat-200k dataset into 4 splits and perform SFT on Llama-3-base to obtain 4 different checkpoints. We also select Llama-3-instruct as a representative model that shares the same prior but was trained on different datasets, and Zephyr [41] as a model with a different prior but trained on similar datasets. For clearly defined steps, we choose MATH-500 [42] as the validation set, sample one trajectory for each question, and split the reasoning path into steps. We extract the logits from the model’s output after inputting the final token of each step, calculate the log-sum-exp, and rank these values within each individual model. If the logits of an LLM possess the property of evaluating state quality, they should share similar ranks across models. We calculate the Kendall rank correlation coefficient (KLCC).

Main Results The results are shown in Fig. 2a. We find that the value rankings of LLMs are positively correlated across all experimental settings. The correlation approaches 1 for the four dataset shards and remains positive even when the post-training dataset or model prior changes. An interesting observation is that the ranking correlation between Zephyr and Llama3-instruct is significantly higher than that between Zephyr and our sharded models, despite Zephyr being trained on UltraChat rather than the same dataset. The positive correlation indirectly validates that the logits function as a value, thus confirming our Assumption 2.

Another observation for implicit reward We find that stronger alignment between the model’s implicit reward and downstream reward correlates with better model performance. During the SFT training process, we calculate the DPO loss on previously annotated pairwise data of AlpacaEval 2 used as an evaluation set. As the benchmark serves as an environment, lower DPO loss indicates greater consistency between implicit rewards and the environment. The results are shown in Figure

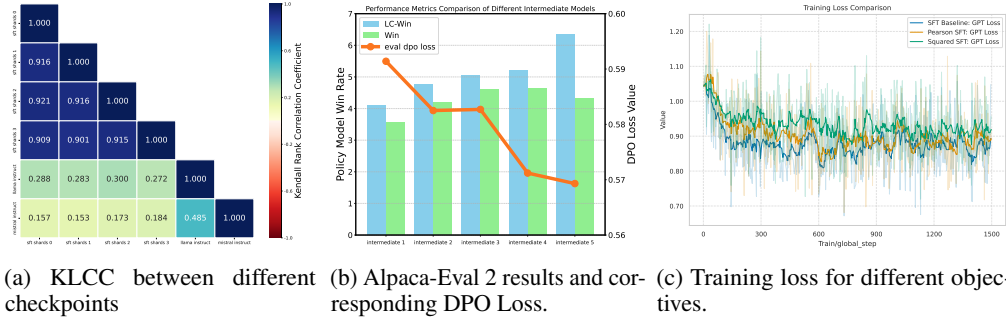


Figure 2: **Left:** We obtain rankings of different models on identical step-wise instruction-response pairs and calculate KLCC to measure agreement between these rankings. **Middle:** Lower values of DPO Loss indicate better alignment between implicit reward and environment reward. And the corresponding AlpacaEval2 result. **Right:** Training loss for objectives derived from: total variation, Pearson χ^2 , and squared Hellinger.

2b. We observe that alignment between rewards and the environment positively correlates with model performance, which aligns with intuitive expectations.

4.5 Reward Stabilization through SFT

SFT in our theory is attempting to bring the implicit reward into an appropriate range for fine-grained modifications. Assumption 1 assumes that at the starting point of the implicit reward search process, the model represents the optimal policy under some unknown reward function. This reward function may significantly differ from the reward function in real downstream tasks. We plot the training curve of the log-sum-exp of the first logits after the prompt and create some early-exit SFT checkpoints to perform DPO on them, as shown in Figure 3. It can be observed that the $V(s_0) = \log \Sigma(\cdot)$ increases rapidly and converges quickly. The downstream task performance exhibits highly consistent trends. We conclude that SFT has already completed its task of bringing the implicit reward to a reasonable range by 150 steps, and subsequent steps focus on more refined modeling.

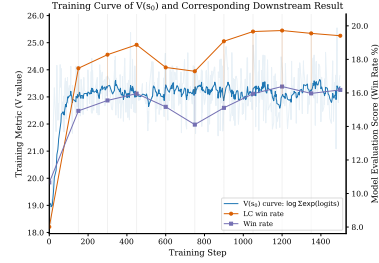


Figure 3: The trend of $V(s_0)$ using the logits of the first response token. Performance on AlpacaEval 2 for post-DPO on SFT checkpoints at corresponding training steps.

5 Broader Impacts, Limitations and Future Work

Broader Impacts: Philosophical Dimension As we mentioned, models learn implicit rewards during SFT, which may lead to further discussions about whether LLMs can be considered entities with preset environmental awareness. This opens philosophical inquiries into the nature of consciousness and the extent to which artificial systems might exhibit consciousness-like properties.

Limitation: We did not experimentally explore additional divergence functions The commonly used KL divergence and JS divergence involve logarithmic and exponential calculations, which can lead to numerical instability. We have attempted various implementations and performed small-value clipping on the data. Although this prevented NaN errors, the loss would still reach extremely large values, such as $6e7$. We did not design specialized operators to implement these methods, resulting in unknown effectiveness for these KL divergence approaches.

Future Work: SFT and DPO multi-object learning Since both the SFT process and DPO process model the implicit reward, a natural idea is to formulate them into multi-objective learning rather than sequential training. Appendix D details some failed attempts at implementing this multi-objective approach, hoping these findings can contribute to future research in this direction.

6 Conclusion

In conclusion, we establish implicit reward learning as a unifying view connecting SFT and preference learning in LLM post-training. We demonstrate that conventional SFT is a special case of implicit reward learning using total variation divergence, limited by an absent KL term. Our approach of reducing learning rates significantly improves model performance, while alternative f -divergence objectives preserving the KL term show additional gains. We extend DPO’s logits-to-Q-function mapping to SFT and confirm SFT’s crucial role in stabilizing random implicit rewards, advancing both theoretical understanding and practical strategies for more effective post-training.

References

- [1] Long Ouyang, Jeffrey 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 F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 1
- [2] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. 1, 6
- [3] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024. 1
- [4] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojuan Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian,

- Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024. 1
- [5] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. 1, 6
- [6] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: less is more for reasoning. *CoRR*, abs/2502.03387, 2025. 1
- [7] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: less is more for alignment. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 1
- [8] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. 1
- [9] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. 1
- [10] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122, 2024. 1
- [11] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *CoRR*, abs/1706.03741, 2017. 1
- [12] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *CoRR*, abs/2305.18290, 2023. 1, 2, 3, 7
- [13] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. 2
- [14] Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. *CoRR*, abs/2407.10490, 2024. 2, 3, 19

- [15] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *CoRR*, abs/2106.12142, 2021. 2, 3, 4, 15
- [16] Markus Wulfmeier, Michael Bloesch, Nino Vieillard, Arun Ahuja, Jorg Bornschein, Sandy H. Huang, Artem Sokolov, Matt Barnes, Guillaume Desjardins, Alex Bewley, Sarah Bechtle, Jost Tobias Springenberg, Nikola Momchev, Olivier Bachem, Matthieu Geist, and Martin A. Riedmiller. Imitating language via scalable inverse reinforcement learning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. 2, 3, 15, 16
- [17] Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^* : Your language model is secretly a q-function. *CoRR*, abs/2404.12358, 2024. 2, 3, 4, 5, 15, 16
- [18] Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, Stanford University, Stanford, CA, USA, June 29 - July 2, 2000, pages 663–670. Morgan Kaufmann, 2000. 3, 15
- [19] Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Ruoyu Sun, and Zhi-Quan Luo. Entropic distribution matching in supervised fine-tuning of llms: Less overfitting and better diversity. *CoRR*, abs/2408.16673, 2024. 3, 15
- [20] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *CoRR*, abs/1606.00709, 2016. 3, 5, 15
- [21] Chris Cundy and Stefano Ermon. Sequencematch: Imitation learning for autoregressive sequence modelling with backtracking. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 3, 15
- [22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. 3
- [23] Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR, 2024. 3, 7
- [24] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. 3, 7
- [25] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. 3, 6, 7
- [26] Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 4998–5017. Association for Computational Linguistics, 2024. 3, 7
- [27] Joey Hong, Anca D. Dragan, and Sergey Levine. Q-SFT: q-learning for language models via supervised fine-tuning. *CoRR*, abs/2411.05193, 2024. 3
- [28] Gokul Swamy, Sanjiban Choudhury, Wen Sun, Zhiwei Steven Wu, and J. Andrew Bagnell. All roads lead to likelihood: The value of reinforcement learning in fine-tuning. *CoRR*, abs/2503.01067, 2025. 3, 4

- [29] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 1433–1438. AAAI Press, 2008. 4, 15
- [30] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3029–3051. Association for Computational Linguistics, 2023. 6
- [31] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *CoRR*, abs/2310.01377, 2023. 6
- [32] Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024. 6
- [33] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *CoRR*, abs/2404.04475, 2024. 6
- [34] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *CoRR*, abs/2404.04475, 2024. 6
- [35] 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. Judging llm-as-a-judge with mt-bench and chatbot arena. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 6
- [36] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. 6
- [37] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. RRHF: rank responses to align language models with human feedback. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 7
- [38] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. Slic-hf: Sequence likelihood calibration with human feedback. *CoRR*, abs/2305.10425, 2023. 7
- [39] Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. 7
- [40] Jiwoo Hong, Noah Lee, and James Thorne. ORPO: monolithic preference optimization without reference model. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 11170–11189. Association for Computational Linguistics, 2024. 7
- [41] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023. 8

- [42] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. 8
- [43] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4565–4573, 2016. 15
- [44] Seyed Kamyar Seyed Ghasemipour, Richard S. Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 1259–1277. PMLR, 2019. 15

A Related Work

A.1 Inverse Reinforcement Learning and f -Divergence

First formalized by [18], Inverse Reinforcement Learning assumes that the expert represents the optimal policy under a certain reward function. It tries to recover this reward function and train a policy model that maximizes it. This approach was significantly advanced by [29] with the principle of maximum entropy to ensure proper exploration. Some commonly used IRL methods include game-theoretic approaches, most notably GAIL [43]. It trains in an adversarial manner using explicit reward recovery and learns policies from the recovered reward function. [20] mathematically provides a unified view using f -divergence. [44] also leverages different forms of f -divergence derived for imitation learning. It involves iterative optimization and primarily explores applications in classical RL scenarios.

Learning an explicit reward function often involves additional parameters and high-variance adversarial optimization. [15] avoids explicit reward learning by learning a Q-function parameterized by both reward and policy. [16] followed this work and introduced non-adversarial imitation learning for large language models. [19] formulates imitation learning using reverse KL divergence with an entropy regularizer and avoids explicitly training a reward function, which helps reduce overfitting. [21] involves different formulations of f -divergence in solving the out-of-distribution (OOD) problem in the generation process and enables backtracking in sequence generation.

We modify the original training objective of maximum-entropy distribution matching to make it more suitable for the post-training process. This allows us to apply closed-form solutions and avoid inner-loop optimization instead of using the Bellman equation like [15, 16]. In comparison with [19], our work utilizes a more general f -divergence formulation [20] and does not introduce negative samples.

B Theoretical Proofs

In this section, we provide detailed proofs of our theoretical results.

Lemma 2 (Fixed-Point Solution for Maximum-Entropy RL [17]). *The optimal policy $\pi^*(a_t | s_t)$ and the corresponding optimal value function $V^*(s_t)$ in the maximum-entropy framework satisfy the following fixed-point equations:*

$$\pi^*(a_t | s_t) = \exp \left(\frac{Q^*(s_t, a_t) - V^*(s_t)}{\beta} \right), \quad (7)$$

$$V^*(s_t) = \beta \log \int_{\mathcal{A}} \exp \left(\frac{Q^*(s_t, a_t)}{\beta} \right) da_t, \quad (8)$$

$$J(\pi^*) = V^*(s_0), \quad (9)$$

where $\pi^*(a | s)$ is the optimal policy. Q^* and V^* are the quality function and value function of the optimal policy.

B.1 Non-adversarial Imitation: Finding Reward

Theorem 3 (Equivalent Objective for Distribution Matching). *Learning a policy that minimizes the f -divergence between expert and policy state-action distributions is equivalent to **first learning an optimal policy under an arbitrary reward function, then optimizing a function of that reward function**:*

$$\min_{\pi} D_f(\mu_{\pi} \| \mu_E) + \beta D_{\text{KL}}(\pi \| \pi_{\text{ref}}), \quad (10)$$

$$= -\min_r \mathbb{E}_{\mu_E}[f^*(-r)] + \underbrace{\max_{\pi} \mathbb{E}_{\mu_{\pi}}[r] - \beta D_{\text{KL}}(\pi \| \pi_{\text{ref}})}_{\text{Has closed-form solution}}, \quad (11)$$

where f^* is the convex conjugate function corresponding to the chosen f -divergence, μ_{π} is the state-action distribution of the policy being learned, and μ_E is the expert's state-action distribution.

Proof. Following the derivation process of non-adversarial distribution matching [16], the training objective 1 can be rewritten into a min-max problem with a conjugate function:

$$\text{Given that } f^*(g) = \max_{x \in \text{dom}(f)} \{xg - f(x)\}, \quad f(x) = \max_{g \in \text{dom}(f^*)} \{xg - f^*(g)\}, \quad (12)$$

$$(13)$$

$$\min_{\pi} D_f(\mu_{\pi} \parallel \mu_E) + \beta D_{\text{KL}}(\pi \parallel \pi_{\text{ref}}), \quad (14)$$

$$= \min_{\pi} \int \mu_E f\left(\frac{\mu_{\pi}}{\mu_E}\right) dsda + \beta D_{\text{KL}}(\pi \parallel \pi_{\text{ref}}), \quad (15)$$

$$= \min_{\pi} \int \mu_E \left(\max_{g: S \times A \rightarrow \text{dom}(f^*)} \left\{ \frac{\mu_{\pi}}{\mu_E} g - f^*(g) \right\} \right) dsda + \beta D_{\text{KL}}(\pi \parallel \pi_{\text{ref}}), \quad (16)$$

$$= \min_{\pi} \max_{g: S \times A \rightarrow \text{dom}_{f^*}} -\mathbb{E}_{\mu_E}[f^*(g)] + \mathbb{E}_{\mu_{\pi}}[g] + \beta D_{\text{KL}}(\pi \parallel \pi_{\text{ref}}). \quad (17)$$

□

By substituting g with $-r$, we can rewrite the min-max formulation:

$$- \max_{\pi} \min_{r: S \times A \rightarrow -\text{dom}_{f^*}} \mathbb{E}_{\mu_E}[f^*(-r)] + \mathbb{E}_{\mu_{\pi}}[r] - \beta D_{\text{KL}}(\pi \parallel \pi_{\text{ref}}). \quad (18)$$

The target is also a saddle point problem, as the conjugate function exhibits convexity and the KL term can be split into an entropy term that is concave and a cross-entropy term that is linear with respect to π . Therefore, the order of min-max operations can be exchanged:

$$- \min_{r: S \times A \rightarrow -\text{dom}_{f^*}} \max_{\pi} \mathbb{E}_{\mu_E}[f^*(-r)] + \mathbb{E}_{\mu_{\pi}}[r] - \beta D_{\text{KL}}(\pi \parallel \pi_{\text{ref}}). \quad (19)$$

The maximization is only performed on the latter part of the expression and has a closed-form solution according to previous work [17]:

$$- \min_r \mathbb{E}_{\mu_E}[f^*(-r)] + \underbrace{\max_{\pi} \mathbb{E}_{\mu_{\pi}}[r] - \beta D_{\text{KL}}(\pi \parallel \pi_{\text{ref}})}_{\text{Has closed-form solution}}, \quad (20)$$

B.2 SFT is a Special Case

The final objective is

$$- \min_r [\mathbb{E}_{\mu_E}[f^*(-r)] + \underbrace{\max_{\pi} \mathbb{E}_{\mu_{\pi}}[r] - \beta D_{\text{KL}}(\pi \parallel \pi_{\text{ref}})}_{\text{Has closed-form solution}}], \quad (21)$$

$$= - \min_r \mathbb{E}_{\mu_E}[f^*(-r)] + V_{\pi}(s_0). \quad (22)$$

Throughout the training process, the reward maintains the equation (4) and the policy π is always the optimal solution of the latter parts. When choosing the Total Variation distance, the conjugate function is $f^*(t) = t$, and the training objective is just the MLE term shown below:

$$- \min_r \mathbb{E}_{\mu_E}[f^*(-r)] + V_{\pi}(s_0), \quad (23)$$

$$= \max_r \mathbb{E}_{\mu_E}[-f^*(-r)] - V_{\pi}(s_0), \quad (24)$$

$$= \max_r \mathbb{E}_{\mu_E}[r] - V_{\pi}(s_0), \quad (25)$$

$$= \max_r \mathbb{E}_{\mu_E}[\log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} + V_{\pi}(s_0) - V_{\pi}(s_t)] - V_{\pi}(s_0), \quad (26)$$

$$= \max_{\pi} \mathbb{E}_{\mu_E}[\underbrace{\beta \log \pi(y|x)}_{\text{MLE}} - \underbrace{\log \pi_{\text{ref}}(y|x)}_{\text{constant}} - V_{\pi}(s_t)]. \quad (27)$$

B.3 SFT Maintains Intrinsic Expected Return Estimation

Theorem 4 (Intrinsic Expected Return). *During the SFT process, the logits l_a of a language model correspond to the Q -function $Q(s, a)$ of the learned implicit reward:*

$$l_a = Q_{\hat{r}}(s, a) + C(s) = \hat{r}(s, a) + \gamma \mathbb{E}_{s_{t+1} \sim P(\cdot|s, a)} [V_{\hat{r}}(s_{t+1})], \quad (28)$$

where \hat{r} is the model’s implicit reward function satisfying eq. (4), γ is the discount factor, and $V_{\hat{r}}(s_{t+1})$ is the value function of the next state. $C(s)$ is a function conditioned only on the state.

Proof. The language model represents token probabilities through a softmax operation over logits l :

$$p(a_i|s) = \frac{e^{l_i/\tau}}{\sum_j e^{l_j/\tau}}, \quad (29)$$

where τ is the temperature parameter, typically set to 1 during training.

As previously discussed, the model represents the optimal policy under some implicit reward function. The probability distribution for this optimal policy satisfies:

$$p(a_i|s) = \frac{e^{Q(s, a_i)/\beta}}{\sum_j e^{Q(s, a_j)/\beta}}. \quad (30)$$

Equating these expressions, we have:

$$\frac{e^{l_i/\tau}}{\sum_j e^{l_j/\tau}} = \frac{e^{Q(s, a_i)/\beta}}{\sum_j e^{Q(s, a_j)/\beta}}, \quad (31)$$

$$e^{l_i/\tau} = e^{Q(s, a_i)/\beta} \cdot \frac{\sum_j e^{l_j/\tau}}{\sum_j e^{Q(s, a_j)/\beta}}, \quad (32)$$

$$e^{l_i/\tau} = k \cdot e^{Q(s, a_i)/\beta}, \quad (33)$$

$$l_i = \frac{\tau}{\beta} Q(s, a_i) + C(s), \quad (34)$$

where β is the KL divergence coefficient. The relationship shows that logits l_i have a linear mapping to the Q -values, with $C(s)$ depending only on the state since it incorporates terms summed over all possible actions. We hypothesize that $C(s)$ remains numerically similar across different states, and we provide supporting evidence for this in the experimental section. \square

C Different f-Divergence Leads to Different Loss Format

In this section, we present the detailed formulation of loss functions derived from different f -divergences.

Name	$D_f(P\ Q)$	Conjugate $f^*(t)$	Training Target
			$\max_{\pi} \mathbb{E}_{\mu_E}[\cdot]$
Total variation	$\frac{1}{2} \int p(x) - q(x) \, dx$	t	$\beta \log \pi(y x) - \log \pi_{\text{ref}}(y x) - V_{\pi}(s_t)$
Kullback-Leibler			
(KL)	$\int p(x) \log \frac{p(x)}{q(x)} \, dx$	$\exp(t) - 1$	$[-\pi_{\text{ref}}(y x) + \pi(y x)] \cdot \exp(-\Delta V - 1) - V_{\pi}(s_0)$
Reverse KL	$\int q(x) \log \frac{q(x)}{p(x)} \, dx$	$-1 - \log(-t)$	$1 + \log(\log \frac{\pi(y x)}{\pi_{\text{ref}}(y x)} + \Delta V) - V_{\pi}(s_0)$
Pearson χ^2	$\int \frac{(q(x) - p(x))^2}{p(x)} \, dx$	$\frac{1}{4}t^2 + t$	$-\frac{1}{4}(\log \frac{\pi(y x)}{\pi_{\text{ref}}(y x)} + \Delta V)^2 + \log \frac{\pi(y x)}{\pi_{\text{ref}}(y x)} + \Delta V$
Squared Hellinger	$\int (\sqrt{p(x)} - \sqrt{q(x)})^2 \, dx$	$\frac{t}{1-t}$	$1 - \frac{1}{1 + \log \frac{\pi(y x)}{\pi_{\text{ref}}(y x)} + \Delta V}$
Jensen-Shannon	$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, dx$	$-\log(2 - \exp(t))$	$\log(2 - \frac{\pi_{\text{ref}}(y x)}{\pi(y x)}) \cdot \exp(-\Delta V)$
	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, dx - \log(4)$	$-\log(1 - \exp(t))$	$\log(1 - \frac{\pi_{\text{ref}}(y x)}{\pi(y x)}) \cdot \exp(-\Delta V)$

Table 4: For different f -divergences, we list their corresponding conjugate functions f^* and the derived training targets. We use the symbol ΔV as an abbreviation for $V_{\pi}(s_0) - V_{\pi}(s_t)$ for simplicity of notation.

D Multi-Object

Since both the SFT process and DPO process model the implicit reward, we investigate whether they can be formulated as multi-objective learning rather than sequential training. We approach this using Lagrangian methods. Specifically, we formulate the multi-objective learning process using the Lagrangian multiplier method. We assume that during the SFT process, the accuracy of the DPO loss should also be kept as low as possible. We formulate this with the following objective:

$$\min_{\pi} \text{SFT loss, s.t., (accuracy of (chosen, reject) - target acc)} < \delta, \quad (35)$$

$$\min_{\pi} \text{SFT loss} + \lambda \cdot \text{DPO loss}. \quad (36)$$

Where δ is the hyperparameter. Following the process of PPO, the parameter λ is adjusted to $\frac{\lambda}{2}$, when $\text{acc} - \text{target acc} < \delta$ and is set to $2 \cdot \lambda$ otherwise.

We train the model directly on the UltraFeedback dataset, and some training metrics during one training are shown below.



Figure 4: Training Metric Visualization for Multi-object Learning

We also experiment with various hyperparameters, including different growth and decay coefficients for λ , but all of these configurations lead to poor downstream results. We think that our failure appears consistent with the theory presented in [14]. As the SFT loss coefficient increases, it tends to increase probability mass in negative regions, thus diminishing the effectiveness of the squeezing effect. We validate in other experiments through interleaved experiments, where we split the SFT and DPO data into four segments and train them in an interleaved manner. The results are shown below.

Table 5: Results for interleaved SFT and DPO training. The table shows the performance after each training stage in terms of Win Rate and LC Win Rate percentages.

Training Stage	AlpacaEval 2	
	Win (%)	LC-Win (%)
<i>Interleaved Training</i>		
SFT 1	5.28	4.86
DPO 1	7.03	8.11
SFT 2	5.06	4.63
DPO 2	11.27	11.94
SFT 3	8.94	4.83
DPO 3	12.94	13.30
SFT 4	8.44	5.13
DPO 4	8.24	5.02

It can be observed that after each DPO stage, subsequent SFT training experiences a significant performance drop, indicating that SFT and DPO objectives are inherently conflicting in our experimental setting.

However, our failure does not necessarily mean this approach is fundamentally infeasible. We have not yet explored its effectiveness in other domains, and we leave this investigation to future work. We hypothesize that in tasks where expert trajectories more closely align with SFT’s assumption that the training data represents optimal behavior, the results may be more promising.

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: We finish this in Section [1](#)

Guidelines:

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

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: We finished this part in Section [5](#)

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [\[Yes\]](#)

Justification: We provide all of our assumptions in Section 3 and our corresponding proof in shown in Section B

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes]

Justification: We provided our training detail and other implementation detail in Section 4

Guidelines:

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

5. Open access to data and code

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

Answer: [Yes]

Justification: The dataset we used are public accessed and we also give out the detail in Section 4

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes]

Justification: All of them are listed in Section 4

Guidelines:

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

7. Experiment statistical significance

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

Answer: [Yes]

Justification: Most of the benchmarks we use already take statistical significance into account. We also illustrate statistical significance in figures such as 2a.

Guidelines:

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

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

8. Experiments compute resources

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

Answer: [Yes]

Justification: We mentioned that in Section 4.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: Yes.

Guidelines:

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

10. Broader impacts

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

Answer: [Yes]

Justification: We discussed in Section 5

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

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

11. Safeguards

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

Answer: [NA]

Justification: We have no such risks.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: Yes.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

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

Answer: [NA]

Justification: We do not release new assets.

Guidelines:

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

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

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

Answer: [NA]

Justification: We do not involve this.

Guidelines:

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

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

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

Answer: [NA]

Justification: We do not involve this.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: We mentioned them in Section 4

Guidelines:

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