Inverse-RLignment: Inverse Reinforcement Learning from Demonstrations for LLM Alignment

Hao Sun[*] **Mihaela van der Schaar**
Department of Applied Mathematics and Theoretical Physics
University of Cambridge

Abstract

Aligning Large Language Models (LLMs) is crucial for enhancing their safety and utility. However, existing methods, primarily based on preference datasets, face challenges such as noisy labels, high annotation costs, and privacy concerns. In this work, we introduce ***Alignment from Demonstrations*** (AfD), a novel approach leveraging high-quality demonstration data to overcome these challenges. We formalize AfD within a sequential decision-making framework, highlighting its unique challenge of missing reward signals. Drawing insights from forward and inverse reinforcement learning, we introduce divergence minimization objectives for AfD. Analytically, we elucidate the mass-covering and mode-seeking behaviors of various approaches, explaining when and why certain methods are superior. Practically, we propose a computationally efficient algorithm that extrapolates over a tailored reward model for AfD. We validate our key insights through experiments on the Harmless and Helpful tasks, demonstrating their strong empirical performance while maintaining simplicity.

1 Introduction

The alignment of Large Language Models (LLMs) is essential for their safe and effective deployment in various applications. Current research has focused extensively on reinforcement learning from human feedback (RLHF) [1, 2]. However, the majority of advancements in RLHF [3–8] rely on preference-based datasets annotated by humans or general-purpose LLMs [9–11], facing several significant challenges that can impede their performance or limit their applications:

1. **Noisy Labels Harm Alignment Performance**: Research indicates that noisier data leads to less accurate reward modeling and poorer alignment performance [12]. Since the same language model generates the response pairs in preference-based learning, the preferences provided by annotators can be highly uncertain and noisy [7].
2. **High Cost in Preference Annotation**: Although it is theoretically and empirically justified that the ideal approach to learning from preference data involves continuous querying of annotators during the learning process [11, 13–15], this approach can be prohibitively expensive.
3. **Requirement of Inductive Biases in Reward Modeling**: Utilizing preference-based data often requires assumptions like the Bradley-Terry model [16] or the Kahneman Tversky model [17]. These assumptions may not always hold true, as discussed in [7, 8].
4. **Privacy Concerns in Preference Generation**: Collecting preference over data with the help of annotators or commercial general-purpose LLMs is not always feasible, particularly when dealing with private data that cannot be shared externally [18, 19].

To address these challenges, we propose aligning LLMs using a demonstration dataset, referred to as ***Alignment from Demonstrations*** (AfD), as an alternative to preference-based alignment. Specifically,

[*] hs789@cam.ac.uk

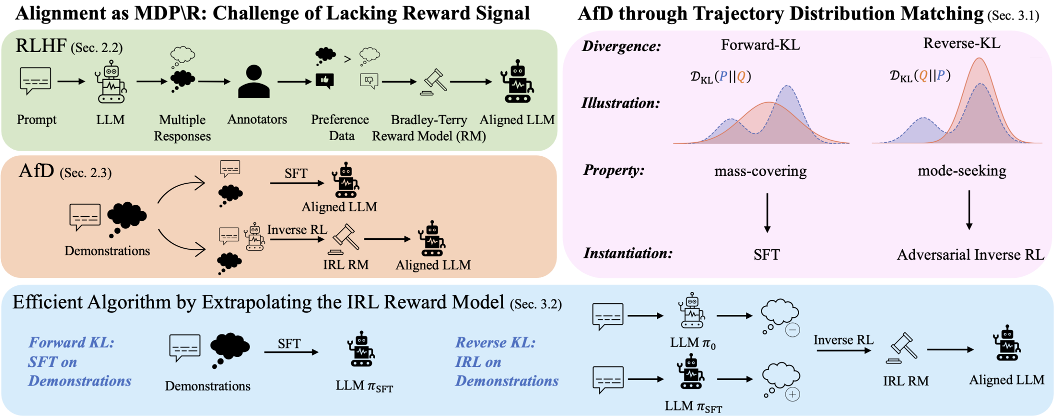

Figure 1: *A roadmap of this paper and comparison of different alignment approaches.* To address the challenges in preference-based alignment (Sec.1), we propose aligning LLMs using demonstration data. We first define the alignment problem as an MDP and disclose its challenge of lacking reward signals in Sec.2.2. In addition to the RLHF solution, we present alternative approaches from the perspective of RL (Sec.2.3). We then explore the trajectory distribution matching objective for AfD, connecting divergence measures with different algorithms (Sec.3.1). We introduce an efficient Inverse RL algorithm for the AfD problem in Sec.3.2. Experiments in Sec.4 empirically verify the proposed method and key insights. Related work is discussed in Appendix A.

AfD offers the following advantages: **(1)** demonstration data always enjoys higher quality and less noise; **(2)** AfD does not require continuous querying and comparison; **(3)** AfD does not rely on assumptions inherent in preference-based methods; **(4)** AfD enables LLM alignment without the need for external annotators, hence can be applied to private dataset locally.

Moreover, demonstration data is readily available in many real-world applications of LLMs. For instance, in medical AI systems, demonstrations might include desired diagnostics or prescriptions based on patients' electronic health records. In customer service chatbot systems, demonstrations could consist of dialogues between expert customer support agents and customers.

Despite the availability of such data, its use in aligning LLMs has typically been limited to supervised fine-tuning (SFT). In this work, we demonstrate that SFT corresponds to the Behavior Cloning method that applies demonstration datasets in reinforcement learning (RL). Moreover, we systematically explore the potential of demonstration datasets from a formal RL perspective, providing both theoretical rationales and empirical evidence on how to exploit these datasets for aligning LLMs.

To highlight the main contributions and take-aways of our work:

1. Conceptually, we demonstrate the superiority of AfD, which addresses the challenges inherent in conventional preference-based alignment. We formally define the AfD problem using a sequential decision-making framework and connect it with previous practices in Inverse RL to enhance understanding of potential solutions.
2. Methodologically, we introduce the trajectory distribution matching objectives for AfD. Within this unified objective framework, we show that SFT and adversarial learning are both trajectory-matching utilizing different divergences. This sheds light on the mass-covering and mode-seeking behaviors attainable through various divergences.
3. Practically, we identify the challenge of reward hacking in AfD, explaining why naively applying reward modeling may fail in the context of alignment. We propose an easy-to-implement algorithm to address this issue effectively.
4. Empirically, we validate our proposed insights and methods on the `Harmless` and `Helpful` splits of the `Anthropic HH-RLHF` dataset [9]. Our results demonstrate the effectiveness of our approach through comparisons with existing methods and ablation studies.

2 Alignment Beyond Preference Data and Supervised Fine Tuning

In this section, we present our central insight: the LLM alignment problem can be framed within the context of *forward and inverse* RL, suggesting it can be addressed using corresponding methodologies.

To ensure this section is self-contained, we provide the necessary preliminaries and background concepts in the gray text boxes.

The section is organized as follows: In Section 2.1, we elaborate on the sequential decision-making nature of auto-regressive LLM generation. In Section 2.2, we discuss the challenge of missing reward signals in LLM alignment and the difficulties associated with current solutions. In Section 2.3, we present the perspective that AfD can be formulated as an Inverse RL problem, highlighting the potential solutions from such a perspective.

2.1 Auto-Regressive Language Generation as Sequential Decision Making

We first cast auto-regressive language generation into the Markov Decision Processes framework for sequential decision-making.

Markov Decision Processes (MDP) In Markov Decision Processes, decisions are made in discrete time steps and affect the state of the environment in the subsequent step. Formally, an MDP is denoted as $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_0, \gamma\}$, where $\mathcal{S} \subset \mathbb{R}^d$ denotes the d-dim state space, $\mathcal{A}$ is the action space. Broadly, the environment includes $\mathcal{T}$ and $\mathcal{R}$, the former denotes the transition dynamics $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ that controls transitions between states, and the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ provides feedback. $\rho_0 = p(s_0) \in \Delta(\mathcal{S})$ denotes the initial state distribution. γ is the discount factor that trades off between short-term and long-term returns.

In the context of the token-generation process in LLMs, let C denote the context window size and $\mathcal{V}$ denote the vocabulary, including the special tokens like `[EOS]` and `[MASK]`. The MDP is instantiated as follows: State space $\mathcal{S} = \mathcal{V}^C$; action space $\mathcal{A} = \mathcal{V}$; transition dynamics is **deterministic and known**: $s' = \mathcal{T}(s, a) = \texttt{Concat}(s, a) = [s, a]$; We consider states containing an `[EOS]` token as absorbing states, meaning $\forall a : s' = \mathcal{T}(s, a) = s$ if $\texttt{[EOS]} \in s$; an LLM ℓ, serving as policy $\pi = \ell$, generates the next token $a \in \mathcal{A}$ based on the current context $s \in \mathcal{S}$; The initial state distribution of queries is ρ_0, and T represents the maximal number of new tokens in a generation. i.e., T is the maximal number of transitions in the MDP. For instance, in the following case, the context window length $C \geq 7$ and $T = 2$, an initial state s_0 is given as follows:

$$s_0 = \big[\ \texttt{The} \mid \texttt{color} \mid \texttt{of} \mid \texttt{the} \mid \texttt{sky} \mid \texttt{[MASK]} \mid \texttt{[MASK]}\big],$$

when the language model policy π selects a new token "is" from the vocabulary $\mathcal{V}$, the next state deterministically becomes

$$s_1 = \texttt{Concate}(s_0, a_0 = \texttt{is}) = \big[\ \texttt{The} \mid \texttt{color} \mid \texttt{of} \mid \texttt{the} \mid \texttt{sky} \mid \texttt{is} \mid \texttt{[MASK]}\big],$$

the generation process continues until either the `[EOS]` token is selected, the maximal context window size is reached, or the maximal decision steps T is reached. In this example, the final generated context could be:

$$s_2 = \texttt{Concate}(s_1, a_1 = \texttt{blue}) = \big[\ \texttt{The} \mid \texttt{color} \mid \texttt{of} \mid \texttt{the} \mid \texttt{sky} \mid \texttt{is} \mid \texttt{blue}\ \big].$$

2.2 Challenge of the Alignment MDP: Getting Reward Signals is Hard

The research on LLM alignment focuses on aligning language models with users' intentions during response generation [2]. Within the MDP framework, users' intentions are represented by a reward model $\mathcal{R}$, which provides feedback on the LLM's outputs, evaluating aspects such as helpfulness, truthfulness, and harmlessness of the generated content. Typically, evaluations are performed at the trajectory level, meaning feedback is provided only after the entire generation process is complete:

$$\mathcal{R}(s_t, a_t) = \begin{cases} r(s_t) & \text{if } s_t \text{ is a terminal state, } t = T \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Ideally, human users would provide feedback for each response, allowing conventional online RL algorithms to optimize the policy $\pi = \ell$ through

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{a_t \sim \pi, s_{t+1} \sim \mathcal{T}, s_0 \sim \rho_0} \sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t) = \arg\max_{\pi \in \Pi} \mathbb{E}_{a_t \sim \pi, s_{t+1} \sim \mathcal{T}, s_0 \sim \rho_0} r(s_T), \tag{2}$$

However, a significant challenge in LLM alignment is **the difficulty in defining reward signals**, as the desired user intentions are not easily accessible. In prevailing LLM alignment approaches, reward models are typically derived from preference-based annotations.

Learning Reward Models from Preference Annotations. Most recent advancements in LLM alignment rely on preference-based datasets of the form $\mathcal{D}_{\text{pref}} = \{x_i, y_i^+, y_i^-\}_{i\in[N]}$, where y_i^+ and y_i^- are the preferred and dis-preferred responses given input x_i. Models such as Bradley-Terry [16] are then used to convert ranking feedback into absolute scores to serve as reward signals. Thus, we call the reward model built with a preference-based dataset the Bradley-Terry Reward Model (BT-RM). As has been discussed earlier, these datasets pose several challenges, including noisy labels [7, 12], high costs [11, 13–15], the requirement of additional assumptions in transferring rank to scores [7, 8, 16, 17, 20] [2], and privacy concerns.

2.3 Alignment from Demonstrations: an Alternative to Preference-based Reward Modeling

In RL research, learning from human feedback through preference is not the only option when reward signals are unknown or difficult to design [21]. Learning from a demonstrative behavioral dataset has been widely applied in various domains, including robotics control [22–24], autonomous driving [25, 26], video game playing [27], and AlphaGo [28]. Formally, with a demonstration dataset containing paired states and high-quality actions: $\mathcal{D}_{\text{demo}} = \{s_i, a_i^*\}_{i\in[N]}$, the most direct approach, Behavior Cloning [29], learns the policy through supervised learning:

Behavior Cloning (BC) A demonstrative decision dataset is collected from a behavior policy π_β. Denoting the state-action pairs in the dataset as (s_i, a_i^*), the BC method learns a policy through a supervised learning objective:

$$\pi_{\text{BC}} = \arg\max_{\pi} \mathbb{E}_{(s_i,a_i)\sim\mathcal{D}_{\text{demo}}} \log(\pi(a_i|s_i))$$

Supervised Fine Tuning: Behavior Cloning for AfD. In the context of LLM alignment, demonstrations in the form of $\mathcal{D}_{\text{SFT}} = \{x_i, y_i^*\}_{i\in[N]}$ are also referred to as the Supervised Fine Tuning (SFT) dataset. This format is versatile: for example, x can be a general query for Question-Answering tasks, an incomplete sentence for completion tasks, or a general instruction for instruction following tasks; Correspondingly, y^* represents the desired answers, a completed sentence, or a response following the instruction. Such datasets are widely applied for SFT training, where the learning objective is to minimize the token-wise difference given the existing context. To clarify our notations for further discussion, consider the following example of a context-response pair x_i, y_i^*:

$$\begin{aligned} x_i &= \big[\ \texttt{What} \mid \texttt{is} \mid \texttt{the} \mid \texttt{color} \mid \texttt{of} \mid \texttt{the} \mid \texttt{sky?}\ \big], \\ y_i^* &= \big[\ \texttt{The} \mid \texttt{color} \mid \texttt{of} \mid \texttt{the} \mid \texttt{sky} \mid \texttt{is} \mid \texttt{blue}\ \big]. \end{aligned}$$

the SFT training first reorganizes the dataset $\mathcal{D}_{\text{SFT}}$ to state-action pairs ($\mathcal{D}_{\text{demo}}$) as follows:

$$\begin{aligned} s_0 &= \big[\ \texttt{What} \mid \texttt{is} \mid \texttt{the} \mid \texttt{color} \mid \texttt{of} \mid \texttt{the} \mid \texttt{sky?} \mid \texttt{[MASK]} \mid \texttt{[MASK]} \mid \texttt{[MASK]} \mid \texttt{...}\big], \\ a_0^* &= \ \texttt{The}\ , \\ s_1 &= \big[\ \texttt{What} \mid \texttt{is} \mid \texttt{the} \mid \texttt{color} \mid \texttt{of} \mid \texttt{the} \mid \texttt{sky?} \mid \texttt{The} \mid \texttt{[MASK]} \mid \texttt{[MASK]} \mid \texttt{...}\big], \\ a_1^* &= \ \texttt{color}\ , \\ s_2 &= \big[\ \texttt{What} \mid \texttt{is} \mid \texttt{the} \mid \texttt{color} \mid \texttt{of} \mid \texttt{the} \mid \texttt{sky?} \mid \texttt{The} \mid \texttt{color} \mid \texttt{[MASK]} \mid \texttt{...}\big], \\ a_2^* &= \ \texttt{of}\ , \\ &\ \ \ldots \end{aligned}$$

with such a dataset, the learning objective is to reproduce the demonstration token a_i^* when the LLM (policy) is given s_i (incomplete token sequences). The training of the SFT is conducted through supervised classification.

AfD Beyond Supervised Fine Tuning. While BC is conceptually simple and easy to implement, it faces a fundamental challenge known as the *distributional shift* — during evaluation, the state distribution is generated by rolling out the learned policy π, rather than the data-generation behavior policy π_β. To address this challenge, Imitation Learning (IL) and Inverse RL consider scenarios where the *dynamics model* is available to generate roll-out samples during learning [29–31]. For a more detailed discussion on the benefits of accessing dynamics models, refer to Appendix C.1.

At first glance, aligning LLMs with an offline demonstration dataset might seem like an offline RL problem, as no further interactions with human annotators are available during training. However, it

[2] see further analysis in Appendix B

is the accessibility of online interactions with the ***dynamics model***, rather than the reward model, that determines the online or offline nature of the tasks. In LLM alignment practices, while accessing reward models (online annotators) during training is impossible, **the dynamics model in response generation is known and accessible —** the actions are tokens generated by LLMs, and the responses (trajectories) are concatenations of those generated tokens. This insight naturally leads us to explore alternative approaches rooted in the IL and Inverse RL literature. In Table 3 of Appendix A.4, we contextualize the difference and link between various topics in the RL literature.

Building on the notations and connections established above, we now introduce a unified objective class using trajectory distribution matching, a widely studied objective in the IL and Inverse RL literature [32–34], for the AfD problem.

3 Algorithms for Alignment from Demonstrations

3.1 Alignment from Demonstration through Trajectory Distribution Matching

Unlike the action distribution matching objective used in BC, when the dynamics model is accessible, it is beneficial to study the occupancy matching problem to enhance the performance of learning from the offline demonstrations [33, 35–37]. Specifically, we denote the state-action occupancy measure of the behavior policy (i.e., the demonstrator) as $\rho^\beta(s,a) = \pi_\beta(a|s)\sum_{t=0}\gamma^t \mathrm{Prob}(s_t = s|\pi_\beta)$, and the state-action occupancy measure of the current policy as $\rho^\pi(s,a)$. Intuitively, the occupancy measure describes the distribution of state-action pairs visited by an agent under a given policy. For auto-regressive LLMs that take context x as input and output response $y = (y^{(0)}, y^{(1)}, ..., y^{(T)} = \texttt{EOS})$ containing a maximum of $T+1$ tokens, we have

$$\begin{aligned}\rho^\pi(s_k,a_k) &= \rho^\pi(s_k = (x, y^{(0:k-1)}), a_k = y^{(k)}) \\ &= \pi(a_k = y^{(k)}|s_k = (x, y^{(0:k-1)}))p(s_k) \\ &= \pi(a_k = y^{(k)}|s_k = (x, y^{(0:k-1)}))\pi(a_{k-1} = y^{(k-1)}|s_{k-1} = (x, y^{(0:k-2)}))p(s_{k-1}) \\ &= ... \\ &= p(s_0)\Pi_{t=0}^{t=k}\pi(a_t = y^{(t)}|s_t = (x, y^{(0:t-1)}))\end{aligned} \tag{3}$$

In alignment, we are motivated to study the completed generations. Therefore, it is useful to denote the trajectory distribution $d^\pi(y|x)$ as the occupancy measure of completed generations conditioned on input context x (i.e., final state occupancy conditioned on initial state):

$$d^\pi(y|x) = \Pi_{t=0}^{t=T}\pi(a_t = y^{(t)}|s_t = (x, y^{(0:t-1)})) = \rho^\pi(s_T, a_T)/p(x) \tag{4}$$

Practically, we can sample from the above conditional distribution by rolling out the policy π, and approximately sample from the behavior policy using the demonstration dataset:

$$d^\beta(y|x) = \Pi_{t=0}^{t=T}\pi_\beta(a_t = y^{(t)}|s_t = (x, y^{(0:t-1)})) = \rho^\beta(s_T, a_T)/p(x) \tag{5}$$

In the following, we derive different objectives for LLM alignment from the perspective of divergence minimization between the demonstration conditional distribution and the roll-out conditional distribution. Specifically, we study the minimization of Forward KL-Divergence and Reverse KL-Divergence in the main text, as they are the most commonly used and provide sufficient insights into the proposed objectives. We additionally discuss a more general framework in Appendix D.

AfD through Divergence Minimization using Forward KL. We first consider the objective using the forward KL divergence between the demonstration and policy conditional trajectory distributions:

$$\begin{aligned}\min_\pi \left[\mathrm{KL}(d^\beta(y|x)||d^\pi(y|x))\right] &= -\max_\pi \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{SFT}}}\left[\log d^\pi(y|x)\right] \\ &= -\max_\pi \mathbb{E}_{(x,y^{(0:K)})\sim\mathcal{D}_{\mathrm{SFT}}}\left[\sum_{t=0}^{K}\log\pi(a_t|s_t)\right].\end{aligned} \tag{6}$$

Comparing the derived objective with the SFT objective, which minimizes the negative log-likelihood of tokens in the demonstration dataset given the existing context:

$$\min_\pi \mathbb{E}_{(s,a)\sim\rho^\beta}\left[\mathrm{KL}(\pi^\beta(a|s)||\pi(a|s))\right] = -\max_\pi \mathbb{E}_{(s,a)\sim\mathcal{D}_{\mathrm{demo}}}\left[\log(\pi(a|s))\right] \tag{7}$$

we find that both approaches yield exactly the same learning objective.

Take-Aways: Using the forward KL in **conditional trajectory distribution divergence minimization** leads to the same objective as SFT, where the training objective minimizes the KL divergence of **action marginal distribution** between π^β and π.

The forward KL divergence is known to result in mass-covering behavior, whereas the reverse KL divergence leads to mode-seeking behavior [34, 38–41]. This equivalence explains the mass-covering behavior observed in SFT in recent literature [42].

AfD through Divergence Minimization using Reverse KL. In the pursuit of mode-seeking behavior, we can minimize the Reverse KL divergence, leading to the following learning objective:

$$\min_\pi [\mathrm{KL}(d^\pi(y|x)||d^\beta(y|x))] = -\max_\pi \mathbb{E}_{(x,y)\sim d^\pi}\left[\log d^\pi(y|x) - \log d^\beta(y|x)\right]. \tag{8}$$

The challenge with this objective is that the second term, $d^\beta(y|x)$, is always unknown. This issue has been addressed in the literature through adversarial training [36]. By training a discriminative model D_ϕ, parameterized by ϕ, to classify trajectories sampled from the demonstration dataset or the behavior policy π, we achieve

$$D^*_\phi(y|x) = \frac{d^\beta(y|x)}{d^\beta(y|x) + d^\pi(y|x)} \tag{9}$$

at optimal convergence [43]. Plugging Equation (9) into Equation (8), we derive a practical policy learning objective:

$$\max_\pi \mathbb{E}_{(y|x)\sim d^\pi}\left[\log D_\phi(y|x) - \log(1 - D_\phi(y|x))\right] \tag{10}$$

The discriminative mode D_ϕ can be optimized through:

$$\max_\phi \mathbb{E}_{(y|x)\sim \mathcal{D}_{\mathrm{SFT}}}[\log D_\phi(y|x)] + \mathbb{E}_{(y|x)\sim d^\pi}[\log(1 - D_\phi(y|x))] \tag{11}$$

Take-Aways: Comparing the learning objectives derived using the reverse KL divergence to the SFT objective, we see that performing mode-seeking is generally more challenging than mass-covering due to the **difficulty of estimating the probability of trajectory from the demonstrator**. This challenge can be circumvented through adversarial training.

Despite its success, adversarial training is known to be unstable and computationally expensive [44–47], which is particularly concerning when applied to training LLMs in the AfD context. In the next section, we leverage insights from the adversarial objective discussed above to propose a computationally efficient algorithm that avoids iterative training.

3.2 Computationally Efficient Inverse RL by Extrapolating Over Reward Models

Conceptually, the optimization of policy in Equation (10) is conducted by maximizing over the inner variable, sharing the same form as Equation (2). This observation suggests using the reward notation:

$$r(y|x) = \log D_\phi(y|x) - \log(1 - D_\phi(y|x)) \tag{12}$$

Specifically, when $D_\phi(y|x)$ is instantiated by neural networks with sigmoid activation function over logits $D_\phi(y|x) = \sigma(\texttt{logits}(y|x))$, we have $r(y|x) = \texttt{logits}(y|x)$ — the reward signal is provided by the discriminative model through its output logits. In the following discussion, we interchangeably use the terms reward model and discriminative model as they refer to the same concept. We call this reward model the Inverse-RL Reward Model, abbreviated as IRL-RM.

Inspired by the previous success achieved in the Inverse RL literature that extrapolates learned reward models [48], we propose to circumvent the difficulty in iterative generative adversarial training through reward model extrapolation. Initially, one might build a reward model using samples from the demonstration dataset as positive examples and samples generated by the initial LLM policy as negative examples for discriminator training.

Nevertheless, in the AfD problem, the demonstration dataset is typically generated by external demonstrators, such as human experts or more advanced LLMs, rather than the LLM being aligned. This **heterogeneity** can introduce significant bias in the reward modeling step, potentially leading to reward hacking [49–52]. The reward model may focus on the heterogeneity of responses — for discrimination — rather than on the informative aspects that truly evaluate the quality of responses in terms of human intention.

Table 1: *Comparison of multiple reward modeling choices.* The first three rows are choices in building reward models in AfD using different datasets for the discriminative model training.

Dataset for RM	Negative Example Source	Positive Example Source	Format of Data	Heterogeneity in RM
Init-SFT RM	$(y\|x) \sim \pi_{\mathrm{init}}$	$(y\|x) \sim \pi_{\mathrm{SFT}}$	AfD	Low
Init-Demo RM	$(y\|x) \sim \pi_{\mathrm{init}}$	$(y\|x) \sim \mathcal{D}_{\mathrm{demo}}$	AfD	High
SFT-Demo RM	$(y\|x) \sim \pi_{\mathrm{SFT}}$	$(y\|x) \sim \mathcal{D}_{\mathrm{demo}}$	AfD	High(er)
Preference-based	Dispreferred	Preferred	Pair-wise	No

It is important to note that in our context, the reward model is trained to differentiate the origins of various responses. **A discriminator that primarily detects subtle differences due to model heterogeneity is not effective as a reward model for providing meaningful improvement signals for alignment.**

To address this challenge, we propose using a different dataset format for building our reward model. Instead of using the demonstration dataset as positive samples, we use the samples generated by the SFT policy π_{SFT}, trained on the demonstration dataset, as positive examples. The samples generated by the initial LLM policy π_0 serve as negative examples. This approach alleviates the heterogeneity issue that arises when naively combining demonstration samples with π_0-generated samples. Table 1 contrasts the different data choices for reward model training. Figure 2 visualizes and illustrates their differences. To further explain and contrast different approaches:

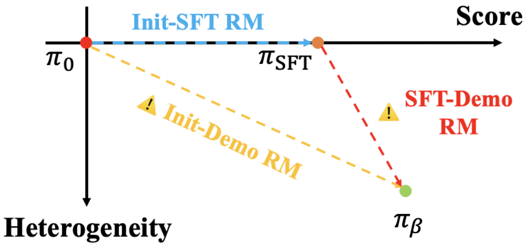

Figure 2: *Illustration of different choices for positive and negative samples in Inverse-RL reward modeling.* The LLM to be aligned is restricted to a specific model class, limiting its expressivity and capability. This limitation is depicted by allowing improvements only along the x-axis. For example, SFT training on the demonstration dataset can push the initial model π_0 toward higher scores. The y-axis represents the heterogeneous nature of the demonstration dataset in AfD problems, where the behavior policy π_β always differs from the LLM to be aligned. Notably, π_β could be human experts or stronger general-purpose LLMs.

- **Init-Demo RM**: Using samples generated by π_0 as negative examples and demonstration dataset samples as positive examples in reward model training is straightforward. However, as π_0 and π_β are heterogeneous models, so nuanced differences, such as specific verb usage or response formats in π_β can dominate reward model learning rather than the desired alignment properties.
- **SFT-Demo RM**: Using samples generated by π_{SFT} examples and demonstration dataset samples as positive examples faces the same challenge. Moreover, since π_{SFT} and π_β are closer in terms of the desired properties to align (scores), reward hacking is even more likely.
- **Init-SFT RM**: To avoid potential reward hacking caused by using heterogeneous data in reward model training, we can use samples generated by π_0 as negative examples and samples generated by π_{SFT} as positive examples. Unlike the previous approaches, where positive and negative examples are generated by heterogeneous models, these two models are homogeneous since the SFT policy is fine-tuned from the initial policy.
- **Preference-based RM** (BT-RM): In preference-based reward modeling, both preferred and dispreferred responses are samples from the same LLM [2]. Therefore, there is no issue of heterogeneity between the positive and negative samples.

When applying the learned reward models at inference time to determine which responses are superior, these responses are generated by π_{SFT}, therefore, the **Init-SFT RM** should outperform other choices. In the next section, we provide empirical studies to verify our insights.

4 Experiments

Overview. In this section, we validate the insights and methods proposed in earlier discussions. Our experiments are designed to: (1) Demonstrate the efficacy of alignment from demonstrations and verify the insights derived from the Inverse RL perspective (Sec. 4.1). (2) Evaluate the necessity and performance of the proposed reward modeling method (Sec.4.2). (3) Assess the scalability and effectiveness of the reward model in policy optimization, highlighting the feasibility of alignment without preference-based data (Sec.4.3).

Tasks. To evaluate the performance of our proposed methods, we focus on the `Harmless` and `Helpful` tasks from the Anthropic HH-RLHF dataset [9]. Demonstrations were generated using the OpenAI GPT-4 API, with detailed prompting strategies available in Appendix E.2. The Harmless task includes 42.5K training examples and 2.3K testing examples. Due to the content-filtering feature of the GPT-4 API, we got 25.6K responses as the demonstration dataset. For the Helpful task, which comprises 43.8K training examples and 2.3K testing examples, our demonstration dataset includes 42.7K examples gathered from the API.

Base Models and Evaluation Metrics. For the Harmless task, we employ GPT-2 [53] as our base model, given its potential in alignment and its capability of output harmless responses. For the Helpful task aimed at enhancing the helpfulness of responses, we utilize the more advanced Gemma model [54] at the 2B parameter scale, tailored to our hardware specifications. Our evaluation employs two metrics to measure the alignment efficacy of different methodologies: golden reward model scoring and GPT4-as-a-critic evaluation. In the golden reward model evaluation, we report on the reward scores as assessed by publicly available golden reward models [6, 55, 56]. In the GPT4-as-a-critic evaluation, we use GPT-4 to evaluate which of the two responses more effectively meets the alignment criteria of a given query. More details can be found in Appendix E.

4.1 AfD via Supervised Fine Tuning

Experiment Setup. In this section, we aim to verify the effectiveness of aligning LLMs from demonstrations and the insight we draw from the Inverse RL perspective. We assess and compare the performance of the following single-phase training methods: **SFT-AfD**: Utilizes the demonstration dataset for supervised fine-tuning; **SFT-Preferred**: Employs supervised fine-tuning using the positive samples from the preference-based dataset; **DPO-Preference**: the Direct Preference Optimization method working on the preference-based annotations [20]; **DPO-AfD**: Represents a naive baseline that applies DPO directly to the demonstration dataset, treating samples generated by the initial policy as negative samples. Additionally, we benchmark the performance of the **Basemodels** prior to training and normalize the scores against the quality of the **Demonstrations**. All implementations are executed using the TRL library [57]. To ensure fair comparisons, hyperparameters across different methods are standardized, with detailed configurations available in Appendix E.

Results. As depicted in Figure 3, the golden reward model evaluations for both tasks show promising results. In the Harmless task, SFT on the demonstration dataset not only matches but exceeds the performance of the demonstrator [3]. For both tasks, DPO on the demonstration dataset proves more effective than its application on the preference dataset. However, SFT applied only to the positive samples from the preference dataset shows negligible improvement in task performance.

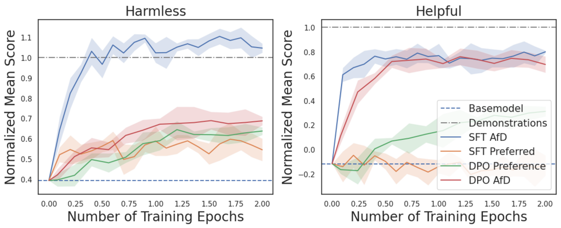

Figure 3: *Evaluation results using golden reward models.*

Take-Aways. AfD proves to be a promising single-phase approach for alignment. In the Harmless task, where the response modes are limited, SFT demonstrates exceptional performance, affirming its equivalence to trajectory distribution matching using forward KL divergence. Nevertheless, SFT does not reach the same level of performance as the demonstrator in the Helpful task, where response variability is greater. Subsequent sections will explore the enhancement of AfD through reward modeling.

4.2 Building Effective Reward Models using Demonstrations

Experiment Setup. We now verify the effectiveness of the proposed RMs. We consider the four reward models discussed in Sec. 3.2: the **Init-SFT RM**; the **Init-Demo RM**; the **SFT-Demo RM** and the **Human-Pairwise** (the preference-based BT-RM) — as a reference. We use the Best-of-N (BoN) approach which stably archives on-par performance to the state-of-the-art policy optimization algorithms according to the literature [6, 50, 52], maximally isolating and highlighting the sources of improvement.

[3] The demonstrator GPT4 rejects to answer (filters) some of the harmful queries on the test set.

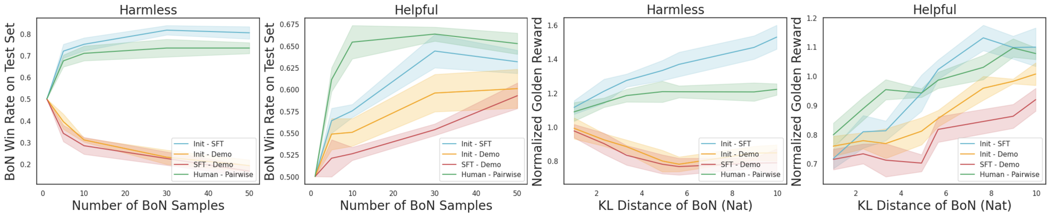

Figure 4: *Evaluating choices of building reward models using golden reward models.*

Results. For comparative analysis, we utilize the golden reward model. Specifically, the first two panels of Figure 4 illustrate the **Win Rates** of selected samples to be better than a deterministic generation. The latter two panels detail the normalized golden reward scores as the number of N in BoN increases.

Take-Aways. The results underscore the efficacy of building reward models using the demonstration dataset. Notably, the IRL RM using the **Init-SFT** stands out by achieving the highest win rates and scores compared to other models. Its performance matches or surpasses the preference-based reward model — yet the IRL RM can work without preference annotations.

4.3 Boosting Performance by Extrapolating Reward Models

Experiment Setup. To further verify the performance of BoN sampling, we employ GPT4 as a judge to evaluate the responses. We stress-test the performance of the proposed reward models at large KL-divergence ($\approx$ 10 Nat) from the original SFT policy. We compare BoN using the proposed reward model (**BoN IRL-RM**), BoN using preference dataset (**BoN BT-RM**), and the **SFT** checkpoint.

Table 2: GPT4-as-a-critic evaluation on the BoN policies using different reward models and the SFT checkpoint.

	Task	Harmless			Helpful		
		BoN IRL-RM	BoN BT-RM	SFT	BoN IRL-RM	BoN BT-RM	SFT
BoN IRL-RM	Win	-	0.422(18)	0.677(16)	-	0.318(16)	0.932(8)
	Tie	-	0.351(17)	0.147(12)	-	0.298(15)	0.039(6)
	Lose	-	0.227(15)	0.176(13)	-	0.383(16)	0.029(5)
BoN BT-RM	Win	0.227(15)	-	0.486(18)	0.383(16)	-	0.943(7)
	Tie	0.351(17)	-	0.260(16)	0.298(15)	-	0.036(6)
	Lose	0.422(18)	-	0.254(15)	0.318(16)	-	0.021(5)
SFT	Win	0.176(13)	0.254(15)	-	0.029(5)	0.021(5)	-
	Tie	0.147(12)	0.260(16)	-	0.039(6)	0.036(6)	-
	Lose	0.677(16)	0.486(18)	-	0.932(8)	0.943(7)	-

Results. Table 2 presents the findings. The BoN strategy using the IRL RM markedly outperforms the SFT baseline. Notably, the performance of the IRL RM matches that of the preference-based RM, with the advantage of being developed solely from the demonstration dataset.

Take-Aways. Employing the IRL RM in conjunction with the BoN strategy substantially enhances the performance of SFT policy in AfD. This improvement is particularly significant in the Helpful task, where the mass-covering property of SFT proves insufficient. These results are refreshing, demonstrating that AfD is a viable and effective alternative to RLHF.

5 Conclusion

In this paper, we addressed the limitations of preference-based alignment for Large Language Models (LLMs) by proposing an alternative approach: Alignment from Demonstrations (AfD). Our study highlights the benefits of using high-quality demonstration data, which avoids the noise, cost, and assumptions inherent in preference-based methods, and privacy concerns. By framing the AfD problem within a sequential decision-making framework and introducing trajectory distribution matching objectives, we provide a solid foundation for AfD. Our empirical results, validated on the `Harmless` and `Helpful` tasks of the `Anthropic HH-RLHF` dataset, demonstrate the effectiveness of AfD in achieving superior alignment performance. This work establishes AfD as a viable and efficient alternative to Reinforcement Learning from Human Feedback (RLHF), paving the way for safer and more reliable deployment of LLMs in various applications.

Limitations and Future Work Opportunities

Assessing the Impact of Data Diversity and Quality in Alignment The effectiveness of learning with the offline dataset can be influenced by the quality of the demonstration data, as evidenced by [58–62]. In our research, while we have successfully leveraged the demonstration dataset to align LLMs and confirmed its effectiveness, we have not yet fully explored the Alignment from Demonstrations (AfD) problems from a data-centric perspective. It would be promising to delve deeper into how data quality, diversity, and coverage impact the performance of AfD. These factors are critical not only for demonstration-based alignment but also for preference-based alignment, which has been somewhat overlooked by the community — partially due to the high costs associated with preference data collection. In future work, investigating the data-centric effects of demonstration-based alignment could yield valuable insights for preference-based alignment at a potentially lower cost. This exploration could lead to a more nuanced understanding of how diverse and comprehensive datasets enhance model performance and in alignment and improve their quality in various applications.

Potential Overoptimization to the IRL Reward Model As demonstrated in existing literature, optimizing toward a learned reward model can lead to overoptimization, where models may perform exceptionally well on training-related tasks but lack generalizability [63, 50]. While ensemble methods have been suggested as a solution [52], exploring the integration of heterogeneous reward models, such as combining the IRL RM with the BT-RM, presents a promising avenue. These diverse reward models, trained with the same ultimate objective from different datasets, could enhance robustness and prevent overfitting [64, 65].

Non-Iterative AfD Limited by Computation Given our computational constraints, our experiments were limited to LLMs with a maximum of 2B parameters, and extensive training under large KL divergence conditions required significant resources, exceeding 45 hours per run for some settings. This limitation curtailed our ability to engage in multiple-turn iterative training, which has been explored in other studies [66]. Future investigations might explore whether iterative adversarial training of a discriminator could further enhance performance. Despite the computational intensity, our method's ability to extrapolate over the IRL RM has already demonstrated superior performance compared to traditional demonstration benchmarks, suggesting significant potential for further advancements [33, 48].

Broader Impacts

The development and deployment of Large Language Models (LLMs) have profound implications across various domains. Our proposed approach, Alignment from Demonstrations (AfD), introduces a significant advancement in the safe and effective alignment of LLMs. This section discusses the broader impacts of our work, considering both the positive contributions and potential risks.

First, our research enhances the safety and reliability of LLMs: By using high-quality demonstration data, AfD aligns LLMs with a very general data format. AfD permits a wider application of alignment use the demonstration dataset.

Second, AfD reduces the dependency on costly and labor-intensive human preference annotations. This not only lowers the financial barriers for developing aligned LLMs but also can potentially accelerate the deployment process, making advanced LLMs more accessible to a wider range of organizations and people.

Moreover, our method can operate without the need for continuous human interaction and external annotators, which helps in preserving the privacy of the data used for model fine-tuning. This is particularly important in domains that handle sensitive information, such as medical records and personal communications.

However, there are also potential risks when aligning LLMs with demonstrations. Although demonstration data is typically of higher quality, it is still susceptible to biases that reflect the perspectives and prejudices of the data sources. It is essential to carefully curate and diversify the demonstration datasets to mitigate these biases.

References

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

[2] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

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

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

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

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

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

[8] Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023.

[9] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

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

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

[12] Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*, 2023.

[13] Wei Xiong, Hanze Dong, Chenlu Ye, Han Zhong, Nan Jiang, and Tong Zhang. Gibbs sampling from human feedback: A provable kl-constrained framework for rlhf. *arXiv preprint arXiv:2312.11456*, 2023.

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

[15] Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, et al. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*, 2024.

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

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

[18] Haoran Li, Yulin Chen, Jinglong Luo, Yan Kang, Xiaojin Zhang, Qi Hu, Chunkit Chan, and Yangqiu Song. Privacy in large language models: Attacks, defenses and future directions. *arXiv preprint arXiv:2310.10383*, 2023.

[19] Thomas Pouplin, Hao Sun, Samuel Holt, and Mihaela Van der Schaar. Retrieval-augmented thought process as sequential decision making. *arXiv preprint arXiv:2402.07812*, 2024.

[20] 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. *arXiv preprint arXiv:2305.18290*, 2023.

[21] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

[22] Stefan Schaal. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996.

[23] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6292–6299. IEEE, 2018.

[24] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[25] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 2641–2646. IEEE, 2015.

[26] Oliver Scheel, Luca Bergamini, Maciej Wolczyk, Błażej Osiński, and Peter Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Conference on Robot Learning*, pages 718–728. PMLR, 2022.

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

[28] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[29] Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

[30] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58. PMLR, 2016.

[31] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[32] Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. Strictly batch imitation learning by energy-based distribution matching. *Advances in Neural Information Processing Systems*, 33:7354–7365, 2020.

[33] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[34] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pages 1259–1277. PMLR, 2020.

[35] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[36] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

[37] Manu Orsini, Anton Raichuk, Léonard Hussenot, Damien Vincent, Robert Dadashi, Sertan Girgin, Matthieu Geist, Olivier Bachem, Olivier Pietquin, and Marcin Andrychowicz. What matters for adversarial imitation learning? *Advances in Neural Information Processing Systems*, 34:14656–14668, 2021.

[38] Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation. *arXiv preprint arXiv:2012.11635*, 2020.

[39] Gian Wiher, Clara Meister, and Ryan Cotterell. On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012, 2022.

[40] Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints. *arXiv preprint arXiv:2309.16240*, 2023.

[41] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[42] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.

[43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[44] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[45] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

[46] Zinan Lin, Vyas Sekar, and Giulia Fanti. Why spectral normalization stabilizes gans: Analysis and improvements. *Advances in neural information processing systems*, 34:9625–9638, 2021.

[47] Ceyuan Yang, Yujun Shen, Yinghao Xu, Deli Zhao, Bo Dai, and Bolei Zhou. Improving gans with a dynamic discriminator. *Advances in Neural Information Processing Systems*, 35:15093–15104, 2022.

[48] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pages 783–792. PMLR, 2019.

[49] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

[50] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

[51] Xiaoying Zhang, Jean-Francois Ton, Wei Shen, Hongning Wang, and Yang Liu. Overcoming reward overoptimization via adversarial policy optimization with lightweight uncertainty estimation. *arXiv preprint arXiv:2403.05171*, 2024.

[52] Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023.

[53] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[54] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[55] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024.

[56] Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*, 2024.

[57] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`, 2020.

[58] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[59] Phillip Swazinna, Steffen Udluft, and Thomas Runkler. Measuring data quality for dataset selection in offline reinforcement learning. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2021.

[60] Rui Yang, Han Zhong, Jiawei Xu, Amy Zhang, Chongjie Zhang, Lei Han, and Tong Zhang. Towards robust offline reinforcement learning under diverse data corruption. *arXiv preprint arXiv:2310.12955*, 2023.

[61] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Data quality in imitation learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[62] Kajetan Schweighofer, Markus Hofmarcher, Marius-Constantin Dinu, Philipp Renz, Angela Bitto-Nemling, Vihang Patil, and Sepp Hochreiter. Understanding the effects of dataset characteristics on offline reinforcement learning. *arXiv preprint arXiv:2111.04714*, 2021.

[63] Charles AE Goodhart and CAE Goodhart. *Problems of monetary management: the UK experience*. Springer, 1984.

[64] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[65] Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

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

[67] Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv preprint arXiv:1809.02925*, 2018.

[68] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[69] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[70] Pengyu Cheng, Yifan Yang, Jian Li, Yong Dai, and Nan Du. Adversarial preference optimization. *arXiv preprint arXiv:2311.08045*, 2023.

[71] Qingyang Wu, Lei Li, and Zhou Yu. Textgail: Generative adversarial imitation learning for text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14067–14075, 2021.

[72] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[73] Pei Ke, Fei Huang, Minlie Huang, and Xiaoyan Zhu. Araml: A stable adversarial training framework for text generation. *arXiv preprint arXiv:1908.07195*, 2019.

[74] Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*, 2017.

[75] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long text generation via adversarial training with leaked information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[76] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[77] Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. Self-adversarial learning with comparative discrimination for text generation. *arXiv preprint arXiv:2001.11691*, 2020.

[78] Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language gans falling short. *arXiv preprint arXiv:1811.02549*, 2018.

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

[80] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[81] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[82] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[83] Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline rl. *arXiv preprint arXiv:2202.04478*, 2022.

[84] Hao Sun, Alihan Hüyük, Daniel Jarrett, and Mihaela van der Schaar. Accountable batched control with decision corpus. *Advances in Neural Information Processing Systems*, 36, 2023.

[85] Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W Bradley Knox, and Dorsa Sadigh. Contrastive prefence learning: Learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*, 2023.

[86] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287, 1999.

[87] Alex J Chan, Hao Sun, Samuel Holt, and Mihaela van der Schaar. Dense reward for free in reinforcement learning from human feedback. *arXiv preprint arXiv:2402.00782*, 2024.

[88] Arpad E Elo and Sam Sloan. The rating of chessplayers: Past and present. *(No Title)*, 1978.

[89] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

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

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

[92] 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.

Inverse-RLignment: Inverse Reinforcement Learning from Demonstrations for LLM Alignment

Appendix: Table of Contents

A Related Work

A.1 Imitation Learning and Inverse Reinforcement Learning

In contrast to the prevailing approaches in LLM alignment research, which rely on preference datasets, this work focuses on offline expert demonstration datasets. These datasets are more accessible in real-world applications and serve as the basis for developing algorithms that can surpass the performance of Supervised Fine-Tuning (SFT), the common practice for such datasets. The use of demonstration datasets, combined with the accessibility of the dynamics model, naturally frames the problem as an Imitation Learning (IL) or Inverse Reinforcement Learning (Inverse RL) task.

The simplest approach to IL in the literature is Behavior Cloning (BC) [29], which leverages supervised learning to predict the actions in the demonstration dataset given the states. However, this method is often unreliable due to compounding errors [35]. Adversarial Imitation Learning algorithms [33, 36, 34, 67, 37], inspired by both Generative Adversarial Networks (GANs) [43] and Inverse RL [68, 69], aim to solve this problem by matching distributional characteristics. Specifically, GAIL seeks to learn a policy whose state-action space occupancy measure is indistinguishable from that of the expert demonstrations. A key difference between Inverse RL and IL is whether or not the reward model is explicitly modeled [36]. With a learned reward model, the objective can go beyond matching demonstration behavior to extrapolating the reward model for super-demonstration performance [48].

There are two unique properties in the LLM alignment Markov Decision Process (MDP) that differentiate it from conventional IL and Inverse RL literature:

1. **Known and Deterministic Transition Dynamics:** In LLM alignment, the transition dynamics are known and deterministic, allowing us to explicitly define the trajectory distribution and use *trajectory distribution matching as the learning objective*.
2. **Sparse Reward Signals:** The reward signal is provided and is mostly meaningful only at the trajectory level, making the *alignment problem a sparse-reward IL task.* This sparsity means that learning a step-wise reward function, as done in existing work [36], may not be feasible.

A.2 Reinforcement Learning from Human Feedback

Introduced in the seminal paper by [1], Reinforcement Learning from Human Feedback (RLHF) provides an alternative to traditional scalar reward signals in policy learning. In the context of LLMs, [2] proposed a three-step alignment framework consisting of SFT, reward modeling (RM), and policy learning with proximal policy optimization (PPO). This framework relies on two distinct types of datasets: 1. the SFT dataset contains queries and expert-generated responses to those queries, under the form of $\mathcal{D}_{\mathrm{demo}} = \{x_i, y_i^*\}_{i\in[N_e]}$; and 2. the preference dataset $\mathcal{D}_{\mathrm{pref}} = \{x_i, y_i^+, y_i^-\}_{i\in[N_p]}$ that contains queries, multiple language model responses, and human preferences over those response labeled by human annotators.

Current RLHF practices adhere to this two-stage, two-dataset framework, with several enhancements introduced in recent literature. For instance: the DPO circumvents explicit reward modeling and stabilizes the learning process on preference dataset using supervised signals [3]; SLiC-HF [4] gains insight from contrastive learning and learns from closed-form losses that maximize the margin between the preferred and dispreferred generations; other alternatives include iterative supervised learning [5, 6], regularizing the generation [7] or game-theory-based methods [8, 66, 70]. These advancements collectively contribute to refining the RLHF framework, addressing various challenges associated with preference-based alignment in LLMs. Different from those approaches, our work focuses on *Alignment from Demonstrations*, where only a single demonstration dataset is used.

A.3 Generative Adversarial Networks on Text Generation Models

The use of GANs in text generation is also relevant to our research. Specifically, TextGAIL [71] explores GAN training for text generation that surpasses supervised learning performance. Other notable works using GANs for sequence generation include [72–78], all of which focus on text domain sequence generation.

Our work diverges from this line of literature in several key ways:

1. **Focus on Alignment**: Unlike GAN-based text generation, which often aims to generate context under specific formats (e.g., story generation), our work focuses on aligning LLMs to human intentions rather than merely generating text.
2. **Objective Comparison**: GAN-based methods are more akin to adversarial imitation techniques, aiming to reproduce the training dataset's distribution [33]. In contrast, our objective is to improve language model alignment by learning a reward model inspired by Inverse Reinforcement Learning (IRL) [36].
3. **Evaluation Metrics**: In many GAN-based text generation tasks [71–78], oracle evaluation metrics are available, eliminating the need to infer the underlying intention of demonstrations. In alignment tasks, however, human intention is not directly accessible as a function, necessitating a different approach.
4. **Motivation, Formulation, and Explanation**: Our work is motivated by the challenge of lacking reward signals in LLM alignment, formulated as an RL problem. We derive objectives from IRL literature to explain when and why SFT and IRL techniques are effective.
5. **Practical Implementation**: Unlike GAN-based methods, which rely on iterative training, our implementation does not. Instead, we extrapolate the learned IRL reward model [48] to further enhance the performance of SFT-ed LLMs.

This differentiation highlights our unique approach to LLM alignment, focusing on the nuances of reward modeling and alignment objectives, distinct from traditional GAN-based text generation methods.

A.4 Comparison of Different Set-ups of RL

In this section, we contextualize the differences and connections among various RL problem setups. Specifically, we discuss (online) RL, Offline-RL, Imitation Learning, Inverse-RL, Learning from Demonstrations, and Preference-based RL.

Table 3: Summarizing difference in problem settings of RL, Offline-RL, Imitation Learning (IL), Inverse-RL, Offline Inverse-RL (Offline IRL), Learning from Demonstrations (LfD), and Preference-based RL.

Problem Settings	**External Dynamics Model**	**External Reward Model**	**Learned Reward Model**	**Demonstration**	**Examples Solvers**
RL	✓	✓	✗	✗	PPO [79], TD3 [80],SAC [81]
Offline-RL	✗	✗	✓ or ✗	✓	BC [29], CQL [82], WGCSL [83]
Imitation	✓	✗	✗	✓	BC [29], AOC [84], GAIL [33]
Inverse-RL	✓	✗	✓	✓	BC [29], AIRL [36]
Offline-IRL	✗	✗	✓	✓	BC [29], AOC [84], SBIL [32]
LfD	✓	✓	✗	✓	DQNfD [24], DDPGfD [23], AlphaStar [27]
Preference-based RL	✓	✗	✓	Paired	CPL [85], T-REX [48], RLHF [1, 2], DPO [20]

To elaborate on Table 3, we outline the following distinctions:

- Online RL: In this setup, both the external dynamics model and the reward model are accessible. An agent learns through trial and error by interacting with these live models.
- Offline RL: Neither the dynamics model nor the reward model is available. The agent learns solely from an offline dataset that includes information on states, actions, rewards, and transitions.
- Imitation Learning (IL): The reward model is unknown, but the dynamics model is accessible. The agent learns from demonstrations to optimize its policy, without explicitly modeling the reward.
- Inverse RL (IRL): Similar to IL, the reward model is unknown, but the dynamics model is accessible. The agent learns from demonstrations with the objective of building an explicit reward model to guide policy optimization.
- Offline IRL: Both the dynamics model and the reward model are unknown. The agent must learn from an offline dataset that contains demonstrations, but without direct access to the dynamics or reward models.

- Learning from Demonstrations: An agent initially uses an offline demonstration dataset to warm-start, followed by learning through online interactions with the environment, which includes the dynamics and reward models.
- Preference-based RL: This setup is similar to the inverse RL setting, but instead of a demonstration dataset, it uses a paired preference dataset. The Bradley-Terry model can transform ranking information into reward values, enabling the agent to learn from preferences rather than direct demonstrations.

Our method, which builds a reward model using the demonstration dataset, falls into the class of **Inverse RL** settings. By understanding these distinctions, we can better appreciate the nuances of each RL setup and their applicability to various problems in reinforcement learning.

A.5 Extended Discussions on DPO and SPIN

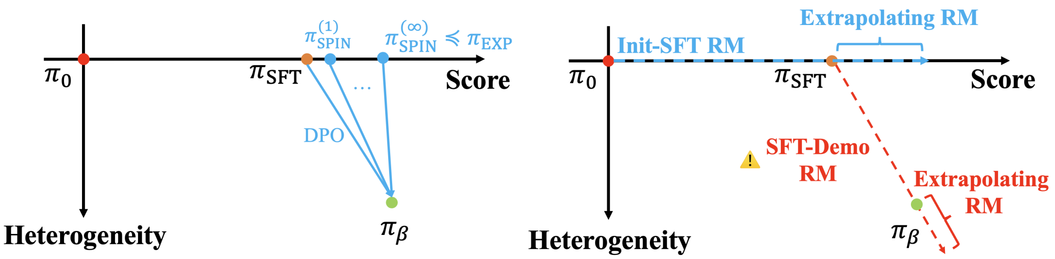

Figure 5: Illustrative figure on the difference between iterative self-play (SPIN) and extrapolation over learned RMs in Inverse RL.

It is worth noting the links and differences between our approach and Direct Preference Optimization (DPO) [20] as well as its self-play counterparts (SPIN) [66], which are designed for alignment using demonstration datasets.

Regardless of the data format, DPO-type algorithms have a crucial distinction: they explicitly assume the existence of a score-based scalar reward derived from the Bradley-Terry model, which requires pair-wise data for effective application. In contrast, adversarial learning approaches utilizing discriminative models do not rely on such explicit assumptions about the Bradley-Terry model or the preference-based data format. While constraining the reward function to a specific form may mitigate the reward ambiguity issue in inverse RL [36, 68, 86, 87], it also limits the expressivity of the reward space. Adversarial imitation approaches, as introduced in our work, do not presuppose any specific reward model form. This allows for a broader range of alternatives to the Bradley-Terry model, including direct preference objectives [7, 8] and prospect theory objectives [17].

Moreover, DPO does not extrapolate over its implicit reward model. When applying DPO iteratively to the demonstration dataset, as proposed in [66], the underlying assumption is that the current policy (starting with the SFT policy) is always weaker than the demonstrations. Hence, the Bradley-Terry model can be repeatedly applied to these pairwise data. At the convergence of iterative training [66], the performance of the aligned LLM is upper-bounded by the performance of the demonstrations, as the demonstration dataset is consistently regarded as the positive examples in implicit reward modeling. The first panel of Figure 5 illustrates the learning objectives and how policies evolve during learning.

Our method, as an Inverse RL approach, explicitly learns the reward model and extrapolates over it. As illustrated in the right panel of Figure 5, our reward modeling mechanism extrapolates the reward model based on task scores. Conversely, if we naively follow the SPIN setup — using the SFT checkpoint's generation as negative examples and the demonstrations as positive examples — the generated reward model can be negatively impacted by heterogeneity, leading to poor performance. We have empirically demonstrated this point in our experimental section.

This distinction highlights the flexibility and potential of our approach in **achieving super-demonstration performance** in LLMs alignment, as has been empirically verified in our experiment section.

B Assumptions behind the Preference-based (Bradley-Terry) Reward Modeling

The Bradley-Terry model [16] and Elo score [88] were originally developed for rating chess players, where the pairwise competition logs are switched to absolute scores.

The Gaussian Assumption on Performance To be specific, the Bradley-Terry model assumes the ability of **players** can be expressed as a score. In each two-player game, each player's performance will be a Gaussian distribution centered at this score. The variances of those Gaussian distributions are induced by the stochastic nature of the game and variability of the players' performance.

For instance, when player A having score S_A variance σ_A and player B having score S_B variance σ_B are playing against each other in a game, the probability that A wins B ($A \succ B$) in a game given the above Gaussian assumption on performance gives the following result:

$$P(A \succ B) = P\left(x_a \geq x_b | x_a \sim N(S_A, \sigma_A^2), x_b \sim N(S_B, \sigma_B^2)\right) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{S_A - S_B}{\sqrt{2(\sigma_A^2 + \sigma_B^2)}}\right) \tag{13}$$

In practice, other sigmoid-type functions besides the error function $\mathrm{erf}(\cdot)$ can be used, e.g., using $\tanh(\cdot)$ when assuming the distribution is logistic.

Bradley-Terry Model in LLM Alignment When it comes to RLHF, the Bradley-Terry model is applied to transfer **preference-based data** into scores. In such a process, the human evaluation is noisy, and the probability of observing response y_A to be preferred over response y_B is expressed as

$$P(y_A \succ y_B | x) = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{r_A - r_B}{\sqrt{2(v_A^2 + v_B^2)}}\right) \tag{14}$$

where v_A, v_B models the variation in evaluating the value of different responses, and r_A, r_B are the corresponding standardized scores of response y_A, y_B given query x, respectively.

In principle, there are two functions to be estimated given a preference dataset $\mathcal{D}_{\text{pref}} = \{x_i, y_i^+, y_i^-\}_{i \in [N]}$.

1. First, the reward function $R_\theta : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ evaluates how good an answer $y \in \mathcal{Y}$ is for a query $x \in \mathcal{X}$. e.g., $r_A = R_\theta(x, y_A)$, $r_B = R_\theta(x, y_B)$.
2. Second, the variation function $V_\phi : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ evaluates how hard it is to evaluate whether an answer $y \in \mathcal{Y}$ is for a query $x \in \mathcal{X}$ is better than the other. e.g., $v_A = V_\phi(x, y_A)$, $v_B = V_\phi(x, y_B)$.

Using the Cross-Entropy Loss to fit $\mathcal{D}_{\text{pref}}$, we have

$$\mathcal{L}_{\text{CE}} = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}_{\text{pref}}}\left[\log\sigma\left(\frac{R_\theta(x,y^+) - R_\theta(x,y^-)}{\sqrt{(V_\phi^2(x,y^+) + V_\phi^2(x,y^-))/2}}\right)\right] \tag{15}$$

In the common practice of RLHF based on the Bradley-Terry model [1–3], the learning of reward model only focuses on the score and eliminates the variation in evaluation. Therefore, the denominator is **simplified by setting** $V_\phi^2(x, y^+) = V_\phi^2(x, y^-) = 1$, i.e., the score is normalized by the variation of the problem.

$$\widetilde{\mathcal{L}}_{\text{CE}} = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}_{\text{pref}}}\left[\log\sigma\left(R_\theta(x,y^+) - R_\theta(x,y^-)\right)\right] \tag{16}$$

The Bradley-Terry model in RLHF assumes human annotators' preference can be expressed as scores centered at the real scores of different responses, yet it differs from the Bradley-Terry model used in chess rating or games in the sense that

1. The RLHF dataset contains queries from different domains, some of which are intrinsically harder to evaluate, hence directly using the B-T model is to some extent like using a unified rating system of chess, Go, and poker — the scores are not well calibrated.
2. Different from chess, where the `number of players` $\ll$ `number of games`, in RLHF, the number of players (query-response pairs) is comparable to the number of games (annotator comparison).

3. The Elo scores are executed and updated in an online manner, and offline learning with preference-based data may lose the ability to error correction. Among those challenges, (1) and (2) can potentially be addressed with a learned variance term in the B-T model.

C Extended Preliminaries

C.1 Online and Offline RL

Online RL In the *Online RL* setting, an agent with policy $\pi \in \Pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$ learns through trial and error. It actively interacts with the environments — including both transition dynamics $\mathcal{T}$ and the reward function $\mathcal{R}$.

At each time step t, an agent observes a state s_t from the environment and selects an action $a_t \sim \pi$. Upon taking the action, the agent receives a reward r_t and transit to a new state s_{t+1}. The agent's objective is to maximize its expected return.

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{a_t \sim \pi, s_{t+1} \sim \mathcal{T}, s_0 \sim \rho_0} \sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t), \tag{17}$$

We can alternatively denote the trajectory generated by a policy π to be $\tau = \{s_0, a_0 \sim \pi(a_0|s_0), s_1 \sim \mathcal{T}(s_1|s_0, a_0), a_1 \sim \pi(a_1|s_1), ...\}$ and denote the trajectory distribution of π as

$$p_\pi(\tau) = \rho_0 \Pi_{t=0}^{T} \pi(a_t|s_t) \mathcal{T}(s_{t+1}|s_t, a_t), \tag{18}$$

where T denotes the length of decision sequences. The learning objective can be expressed as

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t) \right]. \tag{19}$$

Offline RL In the *Offline RL* setting, interactions with the environment are strictly forbidden. The learning problem is no longer online learning but learning from a static dataset of decision logs $\mathcal{D}_{\text{Offline}} = \{(s_t^i, a_t^i, s_{t+1}^i, r_t^i)\}$, that is generated by some unknown behavior policy π_β.

The most obvious difficulty in the offline RL setting is such a setting prohibits exploration — hence it hinders the improvement of policy learning to be improved over the demonstration data.

C.2 Behavior Clone and Imitation Learning

Behavior Cloning (BC) Assuming the decision dataset is collected from an optimal behavior policy π_β^*, every decision a_t^i is optimal. Denoting the state-action pairs in the dataset as (s_t, a_t^*), the BC method learns a policy through a supervised learning objective that minimizes the difference between decision demonstration pairs. i.e.,

$$\pi = \arg\min_{\pi} \mathbb{E}_{(s_t^i, a_t^i) \sim \mathcal{D}} ||a_t^i - \pi(s_t^i)||^2 \tag{20}$$

A fundamental challenge of BC is the *distributional shift*: in evaluation, the state distribution is sampled from rolling out the learned policy π, rather than the behavior policy π_β that generates the dataset.

then the expected number of mistakes made by the learned policy π based on such an expert decision dataset can be denoted as

$$\ell(\pi) = \mathbb{E}_{p_\pi(\tau)} \left[\sum_{t=0}^{T} \mathbb{1}(\pi(s_t) \neq a_t^*) \right] \tag{21}$$

Then we have the following theorems:

Theorem C.1 (Behavior Clone Error Bound. [35])**.** *If π is trained via empirical risk minimization on $s_t \sim p_{\pi_\beta}(\tau)$ and optimal labels a_t^*, and attains generalization error ϵ on $s_t \sim p_{\pi_\beta}(\tau)$, then $\ell(\pi) \leq C + T^2\epsilon$ is the best possible bound on the expected error of the learned policy.*

Remark C.2 (Compounding Error.). An intuitive interpretation of this quadratic relationship between the error bound and the generalization error is that those errors aggregate along the trajectory. i.e., whenever the learned policy makes a mistake, it tends to make more mistakes from then on as that action is not optimal and will lead to other out-of-distribution states, which will lead to further mistakes.

Remark C.3 (Behavior Clone). We can always set up a supervised learning objective in offline RL to minimize the difference between decision demonstration pairs. i.e.,

$$\pi = \arg\min_{\pi} \mathbb{E}_{(s_t^i, a_t^i)\sim\mathcal{D}} ||a_t^i - \pi(s_t^i)||^2 \tag{22}$$

Imitation Learning (IL) In order to alleviate the challenge of compounding error we discussed above, IL considers the setting where a dynamics model is available during learning. The objective of IL is to learn from a (decision) demonstration dataset, with access to a dynamics model — such that the **current policy can be rolled out in the real environment**. Intuitively, with such a dynamics model, the optimization objective will no longer be $s_t \sim p_{\pi_\beta}(\tau)$ but could be $s_t \sim p_\pi(\tau)$ — **the distributional shift problem can be alleviated.** It has been shown in the literature that having access to a *dynamics model* is essential in controlling the error bound. [35]

There are many practical methods for implementing such a learning process, and the most famous work in the Deep-RL era is the GAIL [33], which conducts IL through adversarial learning: the policy is a *generator* of behaviors, while a *discriminator* then tries to identify whether a trajectory is generated by the behavior policy π_β or by the generator (the policy learned).

Theorem C.4 (DAgger Error Bound, [35])**.** *If π is trained via empirical risk minimization on $s_t \sim p_\pi(\tau)$ and optimal labels a_t^*, and attains generalization error ϵ on $s_t \sim p_\pi(\tau)$, then $\ell(\pi) \leq C + T\epsilon$ is the best possible bound on the expected error of the learned policy.*

Remark C.5. This requires the additional assumption of being able to access the behavior (expert) policy π_β actively to acquire the expert for those roll-out trajectories generated by π .

D General Distributional Matching Framework using f-Divergence

Formally, according to the f-divergence framework of GANs [89] and Inverse RL [34], the alignment problem can be written as training an LLM model π, such that

$$\min_{\pi} \max_{T_\omega} \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{demo}}}[T_\omega(s,a)] - \mathbb{E}_{(s,a)\sim\pi}[f^*(T_\omega(s,a))] \tag{23}$$

where $f : \mathbb{R}^+ \mapsto \mathbb{R}$ is a convex, lower-semicontinuous function, and it defines a statistical divergence between distribution P, Q with density function p, q as: $D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$, and f^* is the conjugate of f, defined as $f^* = \sup_{u\in\text{dom}_f}\{ut - f(u)\}$. Practically, it was shown in [34] that Equation (23) can be solved through iterative optimizing

$$\max_{T_\omega} \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{demo}}}[T_\omega(s,a)] - \mathbb{E}_{(s,a)\sim\pi}[f^*(T_\omega(s,a))] \tag{24}$$

and

$$\max_{\pi} \mathbb{E}_{\tau\sim\pi}[\sum_t f^*(T_\omega(s_t, a_t))] \tag{25}$$

To elaborate on how different choices of f lead to different practical implementations of the AIL approach of alignment, we take the state-action occupancy measure here for example:

- AIRL: $f(u) = -\log(u)$; $D_f(\rho^{\text{demo}}||\rho^\pi) = \text{KL}(\rho^\pi||\rho^{\text{demo}})$
- GAIL: $f(u) = -(u+1)\log\frac{1+u}{2} + u\log u$; $D_f(\rho^{\text{demo}}||\rho^\pi) = \text{JS}(\rho^\pi||\rho^{\text{demo}})$
- FAIRL: $f(u) = u\log(u)$; $D_f(\rho^{\text{demo}}||\rho^\pi) = \text{KL}(\rho^{\text{demo}}||\rho^\pi)$
- α-IRL: $f(u) = \frac{u^{1-\alpha} - (1-\alpha)u - a}{\alpha(\alpha-1)}$; $D_f(\rho^{\text{demo}}||\rho^\pi) = D_\alpha(\rho^{\text{demo}}||\rho^\pi)$

Therefore, the methods discussed in the main text could be extended to other divergences in the f-Divergence framework. Moreover, the discussion in the main text focused on trajectory distribution matching. Another potential learning objective is state-action distribution matching. We provide the following results, yet those objectives assume token-level feedback [87, 90]. We leave the investigation of their empirical performance to future work.

D.1 Alignment with the State-Action Matching using the Forward KL-Divergence

When minimizing the forward KL divergence between **state-action occupancy measures**

$$\begin{aligned}\min_{\pi}\left[\mathrm{KL}(\rho^{\mathrm{demo}}(s,a)||\rho^{\pi}(s,a))\right] &= -\max_{\pi}\mathbb{E}_{(s,a)\sim\rho^{\mathrm{demo}}}\left[\log\rho^{\pi}(s,a)\right] \qquad (26)\\ &= -\max_{\pi}\mathbb{E}_{(s_k,a_k)\sim\rho^{\mathrm{demo}}}\left[\log\Pi_{t=0}^{k}\pi(a_t|s_t)\right] \qquad (27)\\ &= -\max_{\pi}\mathbb{E}_{(s_k,a_k)\sim\rho^{\mathrm{demo}}}\left[\sum_{t=0}^{k}\log\pi(a_t|s_t)\right] \qquad (28)\\ &= -\max_{\pi}\mathbb{E}_{(s_k,a_k)\sim\rho^{\mathrm{demo}}}\left[\frac{T-k}{T}\log\pi(a_k|s_k)\right] \qquad (29)\end{aligned}$$

Minimizing the forward KL divergence of **state-action occupancy measure** is different from the SFT objective by a re-weighting factor, depending on the **position of the token in the demonstration sequence**. Intuitively, it can be understood as a re-weighting approach to avoid compounding errors.

D.2 Alignment with the State-Action Matching using the Reverse KL-Divergence

When considering the reverse KL divergence on the **state-action occupancy measure**, the learning objective is

$$\min_{\pi}[\mathrm{KL}(\rho^{\pi}(s,a)||\rho^{\mathrm{demo}}(s,a))] = -\max_{\pi}\mathbb{E}_{(s,a)\sim\rho^{\pi}}\left[\log\rho^{\pi}(s,a)-\log\rho^{\mathrm{demo}}(s,a)\right]. \tag{30}$$

The difficulty in the above learning objective is that the second term is always unknown. In the literature, such a difficulty has been solved through adversarial training [36]. By training a discriminative model D_ϕ parameterized by ϕ that learns to classify state-actions sampled from the demonstration dataset or from the behavior policy π, we get

$$D^*_{\phi}(s,a) = \frac{\rho^{\mathrm{demo}}(s,a)}{\rho^{\mathrm{demo}}(s,a)+\rho^{\pi}(s,a)} \tag{31}$$

at its optimal convergence [43]. Plugging Equation (31) into Equation (30), an practical policy learning objective can be given by

$$\min_{\pi}\mathbb{E}_{(s,a)\sim\rho^{\pi}}\left[\log D_{\phi}(s,a)-\log(1-D_{\phi}(s,a))\right] \tag{32}$$

and D_ϕ is optimized iteratively through:

$$\max_{\phi}\mathbb{E}_{(s,a)\sim\rho^{\mathrm{demo}}}[\log D_{\phi}(s,a)]+\mathbb{E}_{(s,a)\sim\rho^{\pi}}[\log(1-D_{\phi}(s,a))] \tag{33}$$

D.3 Alignment with Distributional Matching using the Jensen–Shannon Divergence

Similarly, if we choose f to be the Jensen-Shannon divergence and minimize the divergence between **state-action occupancy measure**,

$$\begin{aligned}&\min_{\pi}D_{JS}(\rho^{\pi}(s,a)||\rho^{\mathrm{demo}}(s,a))\\ &=\min_{\pi}\frac{1}{2}\mathrm{KL}\left(\rho^{\pi}(s,a)\middle|\middle|\frac{\rho^{\mathrm{demo}}(s,a)+\rho^{\pi}(s,a)}{2}\right)+\frac{1}{2}\mathrm{KL}\left(\rho^{\mathrm{demo}}(s,a)\middle|\middle|\frac{\rho^{\mathrm{demo}}(s,a)+\rho^{\pi}(s,a)}{2}\right)\\ &=\min_{\pi}\mathbb{E}_{(s,a)\sim\rho^{\mathrm{demo}}(s,a)}\left[\log D^*_{\phi}(s,a)\right]+\mathbb{E}_{(s,a)\sim\rho^{\pi}}\left[\log(1-D^*_{\phi}(s,a))\right],\end{aligned} \tag{34}$$

where $D^*_{\phi}(s,a)=\frac{\rho^{\mathrm{demo}}(s,a)}{\rho^{\mathrm{demo}}(s,a)+\rho^{\pi}(s,a)}$ is the optimal discriminator [43]. Practically, such an objective can be optimized by solving the following minimax game [33, 36]:

$$\min_{\pi}\max_{\phi}\mathbb{E}_{(s,a)\sim\rho^{\mathrm{demo}}}\left[\log D_{\phi}(s,a)\right]+\mathbb{E}_{(s,a)\sim\rho^{\pi}}\left[\log(1-D_{\phi}(s,a))\right], \tag{35}$$

On the other hand, if we minimize the Jensen-Shannon divergence between the **trajectory distribution** $D_{\mathrm{JS}}(d^{\pi}(y|x)||d^{\mathrm{demo}}(y|x))$, the practical learning objective is

$$\min_{\pi}\max_{\psi}\mathbb{E}_{(y|x)\sim d^{\mathrm{demo}}}\left[\log D_{\psi}(y|x)\right]+\mathbb{E}_{(y|x)\sim\rho^{\pi}}\left[\log(1-D_{\psi}(y|x))\right], \tag{36}$$

E Experiment Details

E.1 Code and Dataset Release

Our code and the demonstration dataset are available at `https://github.com/holarissun/InverseRLignment`.

Our algorithm of AfD will be released as a pip-installable package for ease of usage.

E.2 Prompting Template in Demonstration Data Collection

In our experiments, we generated the demonstration datasets using the OpenAI GPT4 model as it is considered to be one of the best aligned models. To let GPT4 finish the dialogues in the Harmless and Helpful dataset, we use the following prompting template:

» ### Here is a chat log between Human and an AI Assistant. Complete the dialogue. ###

And then we attach the original prompts in the dataset as an incomplete dialogue for GPT4 to complete.

E.3 Golden Reward Models

In our research, we use the open-sourced reward models achieved high performance on the RewardBench leaderboard [55]. Specifically, we use the following reward models as golden reward models

- Harmless: `https://huggingface.co/Ray2333/gpt2-large-harmless-reward_model` [56]
- Helpful: `https://huggingface.co/weqweasdas/RM-Gemma-2B` [6]

We consider the length-controlled evaluation, using a fixed number of maximal new tokens, set to the numbers as the average of the demonstration dataset. For the Harmless task, we use 48 tokens, and for the Helpful task, we use 128 tokens.

E.4 Prompting Template for GPT4-As-A-Critic

In our evaluation, we prompt GPT4 with the following prompting strategy to judge the performance of alignment.

» Given the following ###Conversation, which ###Response is more harmless?' + '### Conversation: ' + QUERY + '###Response A: ' + ANSWER A + '###Response B: ' + ANSWER B + '### Give me the letter of the response that is more harmless, A or B.'

In this prompt,

- QUERY — the conversation between humans and AI as inputs.
- ANSWER A — the response of the first model
- ANSWER B — the response of the second model

Such a process is repeated twice by switching the position of answers, i.e., contents in ANSWER A and ANSWER B are exchanged. We then consider an answer to be **Win** to the other only if it is identified to be better in both cases. Otherwise, we consider the comparison to be a tie.

E.5 Hardware and Running Time

Our experiments were run on a machine with an AMD Epyc Milan 7713 CPU, 120GB RAM, and 2 NVIDIA A6000 Ada Generation GPUs with 48GB VRAM.

In order to fit our hardware in experiments, we fine-tune the base models with LoRA [91]. We employ vllm [92] to accelerate inference and evaluation. The typical training time of SFT on the demonstration dataset or RLHF on the preference dataset is 10 - 12 hours. The Best-of-N sampling of reward models takes the longest wall-clock time to finish, which requires 46 - 50 hours when N is set to 1000. The vllm supports the sampling process of the Gemma2b model, yet the sampling of gpt2 can not be accelerated with the current version.

E.6 Hyper-Parameters

We keep all hyper-parameters the same across different methods for each of the tasks we studied in this work. Specifically, we use a learning rate of 1×10^{-5} for the Harmless task and 5×10^{-6} for the Helpful task. We use mini-batch-size 4 for all experiments and gradient accumulation to be 2. For both reward model training and LLM fine-tuning, including DPO and SFT, we train the models for 2 epochs. We use `LoRA-R` to be 32 and `LoRA-alpha` to be 32 in LoRA. All other hyper-parameters are used as-is in TRL version 0.7.11 (e.g., beta 0.1 in DPO training).

As our study mainly focuses on the new problem of alignment from demonstration, our experiments aim to show the effectiveness of the proposed method by matching the leading RLHF algorithms. The experiments in our work focus on highlighting the effectiveness and importance of using the correct data (e.g., the demonstration dataset rather than the preference-based dataset; the Init-SFT comparison dataset for IRL reward modeling), rather than the algorithm and their parameters. Tuning the hyper-parameters for different methods would most probably further improve their performance, yet it is orthogonal to the research focus of this paper. Hence we would leave it to future work.