Inverse Reinforcement Learning Meets Large Language Model Post-Training: Basics, Advances, and Opportunities

Hao Sun *hs789@cam.ac.uk*
Department of Applied Mathematics and Theoretical Physics
University of Cambridge
Cambridge, United Kingdom

Mihaela van der Schaar *mv472@cam.ac.uk*
Department of Applied Mathematics and Theoretical Physics
University of Cambridge
Cambridge, United Kingdom

Abstract

In the era of Large Language Models (LLMs), alignment has emerged as a fundamental yet challenging problem in the pursuit of more reliable, controllable, and capable machine intelligence. The recent success of reasoning models and conversational AI systems has underscored the critical role of reinforcement learning (RL) in enhancing these systems, driving increased research interest at the intersection of RL and LLM alignment. This paper provides a comprehensive review of recent advances in LLM alignment through the lens of inverse reinforcement learning (IRL), emphasizing the distinctions between RL techniques employed in LLM alignment and those in conventional RL tasks. In particular, we highlight the necessity of constructing neural reward models from human data and discuss the formal and practical implications of this paradigm shift. We begin by introducing fundamental concepts in RL to provide a foundation for readers unfamiliar with the field. We then examine recent advances in this research agenda, discussing key challenges and opportunities in conducting IRL for LLM alignment. Beyond methodological considerations, we explore practical aspects, including datasets, benchmarks, evaluation metrics, infrastructure, and computationally efficient training and inference techniques. Finally, we draw insights from the literature on sparse-reward RL to identify open questions and potential research directions. By synthesizing findings from diverse studies, we aim to provide a structured and critical overview of the field, highlight unresolved challenges, and outline promising future directions for improving LLM alignment through RL and IRL techniques.

1 Motivation: Reinforcement Learning in the Era of Large Language Models

1.1 The Success of Large-Scale Data-Driven Models

In the era of large foundation models, great success has been achieved by scaling up training compute, data, and the number of model parameters (Vaswani et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022; Zhang et al., 2024a). And such great success spans in many fields from natural language generation (Achiam et al., 2023; Meta, 2024; Team et al., 2024), understanding (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020), high-resolution image generation (Ramesh et al., 2021; 2022; Podell et al., 2023; Zhang et al., 2023), editing (Hertz et al., 2022; Zhang et al., 2025), audio (Kong et al., 2020; Copet et al., 2023; Wang et al., 2023a) and video generation (Brooks et al., 2024), decision-making and control (Reed et al., 2022; Brohan et al., 2023; Bousmalis et al., 2023; Driess et al., 2023).

Among those large-scale, successful data-driven models, we are particularly interested in the Large Language Models (LLMs), given their high potential of transparency through natural language (Liao & Vaughan, 2023;

Lindsey et al., 2025), the recent progress of applying those models in general-purpose assistant systems (Ouyang et al., 2022), and agentic use-cases to perform deep analysis (OpenAI, 2025).

However, while LLMs can understand and follow users' instructions (Zhou et al., 2023b), quickly adapt to new tasks (Brown et al., 2020), and can have reasoning abilities to finish complex tasks (Wei et al., 2022b; Kojima et al., 2022; Guo et al., 2025), those systems can not always do self-correction by themselves (Huang et al., 2023; Kamoi et al., 2024) and keep the system continue improving.

1.2 The Success of Large Scale Reinforcement Learning

Since the success of Reinforcement Learning (RL) for Atari games and AlphaGo (Mnih et al., 2013; Silver et al., 2016), the ability of RL in achieving super-human performance has been demonstrated in board games (Silver et al., 2017; Schrittwieser et al., 2020), real-time strategy games (Vinyals et al., 2019; Berner et al., 2019), and many other applications ranging from chip design to algorithmic optimization Mirhoseini et al. (2021); Fawzi et al. (2022); Mankowitz et al. (2023). By interacting with the environment, those RL systems can keep improving their abilities to solve the training tasks and finally achieve super-human performance.

While RL can achieve super-human performance and create novel solutions to problems, the transparency of RL systems remains a non-trivial challenge (Qing et al., 2022; Milani et al., 2022). It's challenging for humans to identify, understand, and learn from those creative behaviors (Menick, 2016; Bory, 2019; Zahavy et al., 2023).

1.3 Combining the Success from Both Sides: RL Meets LLM Post-Training

Given the success of RL and LLM in their respective domains, combining the success from both sides becomes promising. From an RL-centered perspective, if we can harness LLMs to achieve superhuman performance, natural language may serve as the ideal interface to leverage RL's creativity to inspire humans; from the LLM-centered standpoint, RL can grant LLMs the ability to continually enhance performance on reward-defined tasks.

LLM Alignment and Post-Training In this paper, we exchangably use *alignment* and *post-training* to denote optimizing pre-trained LLMs aimed at gaining specific capabilities. RL naturally aligns with such a learning paradigm as well as the capability can be quantified as a reward.

RL in Conversational AI In general-purpose dialogue systems, RLHF is proven to be an effective approach to enhance LLMs' abilities through preference annotation, and this is especially useful in tasks where golden evaluation metrics are difficult or impossible to define (Christiano et al., 2017; Bai et al., 2022b; Stiennon et al., 2020; Bai et al., 2022a). Further investigation on alternative approaches explored different aspects of improving such a paradigm (Rafailov et al., 2023; Ethayarajh et al., 2024; Zhao et al., 2023; Liu et al., 2023; Ji et al., 2024; Meng et al., 2024; Yin et al., 2024; Sun et al., 2024c; Azar et al., 2024). The enormous user base and their feedback provide OpenAI with a continuous stream of data to model user preferences and enhance the experience.

RL in Mathematical Reasoning In mathematics, AlphaProof and AlphaGeometry2 won silver medals at the International Mathematical Olympiad (IMO) (AlphaProof and AlphaGeometry teams, 2024; Trinh et al., 2024). Moreover, DeepSeek-R1 (Guo et al., 2025) demonstrated the power of RL in mathematical reasoning and more general reasoning tasks. Through the technique of RL, LLMs can learn the behavior of deep thinking or self-reflection, and then improve their ability in solving tasks by generating more tokens (Xu et al., 2025).

Opportunities and Key Challenges What are the key challenges in scaling up RL to a wider range of LLM tasks and applications? First, lacking reward signals. In most tasks, we do not have rule-based reward signals as in the math or coding tasks. In those cases, efficient reward modeling becomes vitally important, and this will be the focus of the 3rd section of this paper. Second, the demand for computing.

The prohibitively high cost in compute hinders the open-source development of this field. To help alleviating such a challenge, we will introduce a reward model infrastructure to conduct Inverse RL research on LLM alignment efficiently. With such an infrastructure, researchers without GPUs can also efficiently verify their ideas. For the 3rd challenge, although we have lots of algorithms in RL, there is no silver bullet. We need to consider the properties of different LLM alignment tasks. Hence, we are motivated to have this paper, which tries to bridge the gap between Inverse RL and LLM alignment could be helpful for potential future research.

2 Revisiting the Foundations of Reinforcement Learning under an LLM Context

In reinforcement learning, an agent interacts with the external environment to collect feedback and observations. The objective of such a learning process is to maximize the long-term return (Sutton et al., 1998).

2.1 Markov Decision Processes

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.

To solve an MDP problem, the high-level idea is fairly simple — the agent should learn to *discover* and *repeat* successful actions and trajectories. Formally in literature, the *discover* is referred to as *exploration*, and *repeat* is *exploitation* (Sutton et al., 1998).

Although the idea is simple and elegant, the practice and implementation are far from trivial. One fact in the RL literature is that, while some RL algorithms can be better than others in some tasks, there is no single RL algorithm that performs best on every task. Each algorithm comes with its own assumptions, strengths, and limitations. The choice of algorithm should be determined by environmental properties, and sometimes resource constraints. Table 1 shows some examples of different tasks and corresponding successful algorithms, categorized by the structure of their action space $\mathcal{A}$, state space $\mathcal{S}$, reward signal $\mathcal{R}$, algorithmic approaches, and whether their transition dynamics $\mathcal{T}$ are known.

Table 1: Representative RL Tasks and Characteristics

Task	$\mathcal{A}$	$\mathcal{S}$	$\mathcal{R}$	**Algorithms**	$\mathcal{T}$
Atari-Dense	Disc. $\sim 10^1$	Image	Dense	DQN	Unknown
Atari-Explore	Disc. $\sim 10^1$	Image	Sparse	Curiosity-Driven	Unknown
Board Game	Disc. $\sim 10^2$	Disc. $\sim 10^{100}$	Sparse	MCTS, Self-Play	Known
Dota2	Disc. $\sim 10^6$	PO, Mixed	Mixed	(MA)PPO	Unknown
StarCraft	Disc. $\sim 10^{26}$	PO, Mixed	Mixed	BC, AC, League	Unknown
Robotics-GC	Cont. Dim $\sim 10^2$	Cont. Dim $\sim 10^2$	Sparse	Hindsight Exp. Replay	Unknown
Locomotion	Cont. Dim $\sim 10^2$	Cont. Dim $\sim 10^2$	Dense	SAC, TD3, TD7	Unknown
Reasoning	Disc. $\sim 10^6$	$\mathcal{V}^C$	Sparse	GRPO	Known
RLHF	Disc. $\sim 10^6$	$\mathcal{V}^C$	Noisy Sparse	PPO, DPO, REINFORCE	Known

In Atari games, pioneered by the DQN model Mnih et al. (2015), the agent operates in a discrete action space with visual input and mostly dense rewards. However, when modified for sparse-reward settings (e.g., exploration-focused variants), intrinsic motivation and curiosity-driven approaches become necessary Pathak et al. (2017). Board games like Go involve very large discrete state spaces and sparse rewards, where planning and search-based methods such as MCTS and self-play have proven highly effective Silver et al. (2017). Dota 2 and StarCraft exemplify complex, partially observable (PO) hybrid state spaces, multi-agent environments with both sparse and dense reward components. These settings have motivated scalable and distributed algorithms like (Multi-Agent) PPO Berner et al. (2019) and league-based training with off-policy actor-critic methods Vinyals et al. (2019). Multi-goal robotic manipulation tasks typically involve continuous state and action spaces, and sparse goal-conditioned rewards. Hindsight Experience Replay (HER) has shown promise in addressing the challenge of learning from failures in such settings Andrychowicz et al. (2017). In locomotion tasks, methods such as TD3, SAC, and TD7 (Fujimoto et al., 2018; Haarnoja et al., 2018; Fujimoto et al., 2023) are widely used. LLM-based reasoning tasks operate over a large discrete action space (e.g., vocabulary tokens raised to the length of context length) and often rely on sparse or delayed rewards. Recently, GRPO Team (2024) has been proposed to handle these cases more effectively. Finally, Reinforcement Learning from Human Feedback (RLHF) tasks focus on aligning language models with human preferences. These involve noisy, implicit reward signals derived from pairwise or ranked feedback. Algorithms such as PPO, DPO, and REINFORCE are widely used in this domain Christiano et al. (2017); Rafailov et al. (2023); Williams (1992).

Take-away **There is no silver bullet in RL.** The choice of algorithm should be determined by environmental properties (state space, action space, transition dynamics, reward sparsity etc.), and resource constraints.

2.2 Characterizing LLM Generation in an MDP Framework: The Challenge of Missing Reward

Using the MDP framework discussed above, we can formally describe the LLM token generation process. 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 $s_0 \sim \rho_0$, sampled from the initial prompt or user query distribution ρ_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:

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

When it comes to the reward function $\mathcal{R}$, its definition is less clear and non-trivial. In LLM generation, there is no external reward verifier, such as "winning a game" or "achieving a goal". Even with the task of mathematical reasoning, where a rule-based reward model is used to verify whether the answer is correct or not, we do not have a mathematical oracle that tells us the outcome is correct or not, but we have to generate the reward in a data-driven manner.

Finally, the discount factor γ determines the preference over response conciseness. When setting it to 1, it means generating a correct response containing 300k tokens would be equally good to another correct response using 300 tokens (e.g., in the thinking mode when we prioritize the correctness of the final answer of a challenging math question). When setting it to a number smaller than 1, it means we would prefer more concise or shorter responses to finish a given task.

Table 2 summarizes the MDP components of LLM generation, with a highlighted (missing) reward function that has to be generated in a data-driven approach. In the next section, we will revisit the classical methods in the RL literature and draw inspiration from the classics in solving MDP\R problems.

Table 2: LLM generation as an MDP\R

Component	**Interpretation**
$\mathcal{S}$ (State)	Current sentence
$\mathcal{A}$ (Action)	Tokens (or their combinations)
$\mathcal{P}$ (Transition)	Concatenation of tokens
ρ_0 (Initial state distribution)	Prompt / Query distribution
$\mathcal{R}$ (Reward)	*Data-Driven*
γ (Discount factor)	≤ 1 (e.g., no discount vs. brevity preference)

2.3 MDP\R: Markov Decision Processes without Reward Function

In MDPs, the learning objective is to maximize cumulative reward over decision steps. However, in an MDP\R, how to effectively optimize the policies without a reward function? In RL literature, we can learn from a **behavior dataset** in those MDP\R settings.

Motivations and Practices of MDP\R: Learning from Behavior Datasets In many real-world tasks, reward signals are difficult to specify. For example, in early autonomous driving systems such as ALVINN (Pomerleau, 1988), the learning objective is to mimic human driving behavior—a goal that is inherently hard to formalize as a reward function. More generally, in imitation learning setups (Hayes & Demiris, 1994), behavior datasets serve as a direct and expressive means of specifying desired behaviors, without the need for manually crafted reward functions. This difficulty in defining explicit reward signals is also evident in complex robotic skill learning (Peng et al., 2018), where behaviors such as agile locomotion or acrobatic motions are more easily demonstrated than described through rewards. In the context of LLM alignment (Bai et al., 2022b), properties such as helpfulness, harmlessness, and summarization quality are similarly challenging to quantify through reward functions alone (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022a).

Beyond ill-defined reward settings, behavior datasets are also valuable in problems where reward functions are well-defined but sparse. In such scenarios, learning from behavior can substantially aid exploration. A canonical example is the game of Go, where the objective of "winning" is clearly defined, yet extremely difficult to achieve from random play due to the sparsity of the reward. Systems such as AlphaGo, AlphaStar, and OpenAI Five successfully leveraged expert demonstrations or replay data from human players to initialize and guide policy learning (Silver et al., 2016; Vinyals et al., 2019; Berner et al., 2019). Similarly, in robotics control tasks with sparse but well-defined success-based reward signals, incorporating expert demonstrations can significantly enhance sample efficiency and guide exploration. This class of techniques is broadly known as Learning from Demonstrations (LfD) (Nair et al., 2018; Hester et al., 2018).

Methods for MDP\R: Imitation Learning and Inverse Reinforcement Learning In general, approaches to learning from behavior can be broadly categorized into two classes: Imitation Learning (IL) and Inverse Reinforcement Learning (IRL). Both can be interpreted as instances of behavioral distribution matching, where the goal is to align the learned policy's behavior distribution with that of the expert (Ghasemipour et al., 2020; Ke et al., 2021). A common assumption underlying both IL and IRL is the availability of the environment dynamics, allowing for potentially unlimited interactions with the environment through rollouts.

In contrast, their offline counterparts — namely Offline IL and Offline IRL — operate under the constraint that the environment dynamics are unknown and no additional interaction is possible (Jiang et al., 2020; Jarrett et al., 2020; Yu et al., 2023). This offline setting introduces significant challenges, most notably the inability to explore counterfactual behaviors that are not present in the demonstration dataset (Fujimoto et al., 2019). As a result, the learned policy is limited to the support of the existing data, which can hinder its ability to improve beyond what is demonstrated (Zolna et al., 2020).

In light of the challenges discussed above, particularly the issues of distributional shift and compounding errors in the offline setting, we now turn to practical algorithms for IL and IRL that explicitly leverage access

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**	**Behavior Dataset**	**Examples Solvers**
RL	✓	✓	✗	✗	PPO (Schulman et al., 2017), TD3 (Fujimoto et al., 2018), SAC (Haarnoja et al., 2018)
Offline-RL	✗	✗	✓ or ✗	✓	BC (Pomerleau, 1991), CQL (Kumar et al., 2020), WGCSL (Yang et al., 2022)
Imitation	✓	✗	✗	✓	BC (Pomerleau, 1991), AOC (Sun et al., 2023b), GAIL (Ho & Ermon, 2016)
Inverse-RL	✓	✗	✓	✓	BC (Pomerleau, 1991), AIRL (Fu et al., 2017)
Offline-IRL	✗	✗	✓	✓	BC (Pomerleau, 1991), AOC (Sun et al., 2023b), SBIL (Jarrett et al., 2020)
LfD	✓	✓	✗	✓	DQNfD (Hester et al., 2018), DDPGfD (Nair et al., 2018), AlphaStar (Vinyals et al., 2019)
Preference-based RL	✓	✗	✓	Preference	CPL (Hejna et al., 2023), T-REX (Brown et al., 2019), RLHF (Christiano et al., 2017; Ouyang et al., 2022), DPO (Rafailov et al., 2023)

to the environment dynamics. These methods exploit interactions with the environment to mitigate error accumulation and achieve more robust policy learning.

2.4 Practical IL and IRL Algorithms

To make these ideas concrete, we now review practical algorithmic implementations of IL and IRL, focusing on how access to environment dynamics helps address the limitations of purely offline learning. We begin with the most basic form of imitation learning, Behavior Cloning (BC), which requires only demonstration data and no environment interaction, and then discuss more advanced approaches that incorporate rollouts and interactions for distribution matching.

In IL, the objective is to recover the behavior of an expert policy π_β using a parameterized learner policy π. The most straightforward approach of IL is BC (Pomerleau, 1988), which instantiates the imitation through supervised learning.

Behavior Cloning (Pomerleau, 1988) A demonstrative decision dataset is collected from a behavior policy π_β. Denoting the state-action pairs in the dataset as $(s_i, a_i^*) \sim \mathcal{D}$, the BC method learns a policy through a supervised learning objective:

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

Despite its simplicity and minimal requirements on the behavioral data (i.e., only needing state and action pairs but nothing else), its offline nature leads to a fundamental challenge — known as 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. The expected number of mistakes made by the learned policy π based on such an expert decision dataset can be denoted as

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

Then we have the following theorems:

Theorem 2.1 (Behavior Clone Error Bound. (Ross et al., 2011))**.** *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 2.2 (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.

In order to alleviate the challenge of compounding error we discussed above, IL considers the setting where a dynamic 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. (Ross et al., 2011)

Theorem 2.3 (DAgger Error Bound, (Ross et al., 2011))**.** *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 2.4*.* 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 .

Beyond supervised imitation, adversarial imitation learning (AIL) methods such as GAIL (Ho & Ermon, 2016) formulate imitation as a distribution matching problem using adversarial training, drawing direct inspiration from generative adversarial networks (GANs). These methods introduce a discriminator to distinguish expert from learner behavior, and optimize the policy to fool this discriminator.

While specific AIL methods such as GAIL provide effective practical algorithms, a more general understanding can be achieved by viewing them through the lens of f-divergence minimization (Nowozin et al., 2016). This perspective reveals that many AIL variants are in fact instantiations of a unified framework, differing only in the choice of divergence function. The following formulation, proposed by Ghasemipour et al. (2020), provides a general min-max optimization structure underlying these methods:

f-divergence Adversarial Imitation Learning (Ghasemipour et al. (2020) The general adversarial imitation learning problem can be formalized as the following min-max objective:

$$\min_{\pi} \max_{T_\omega} \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{demo}}}[T_\omega(s,a)] - \mathbb{E}_{(s,a)\sim\pi}[f^*(T_\omega(s,a))] \tag{2}$$

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, Equation (2) can be solved through iterative optimizing

$$\max_{T_\omega} \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{demo}}}[T_\omega(s,a)] - \mathbb{E}_{(s,a)\sim\pi}[f^*(T_\omega(s,a))] \tag{3}$$

and

$$\max_{\pi} \mathbb{E}_{\tau\sim\pi}[\sum_t f^*(T_\omega(s_t,a_t))] \tag{4}$$

Using ρ to denote the state-action visitation frequency[1], Table 4 elaborates on how different choices of f lead to different practical implementations of the AIL approach.

Importantly, the difference between those algorithms also highlights the difference between IL and IRL: in IL, the learning objective is to directly recover the expert behavior by imitating it, whereas in IRL, a reward model is learned from the behavior datset, such that maximizing accumulated return predicted by such a learned reward will induce the behavior policy.

[1] Ni et al. (2021) discussed when only state visitation frequency is available.

Table 4: Different f-divergences used in Adversarial IRL methods

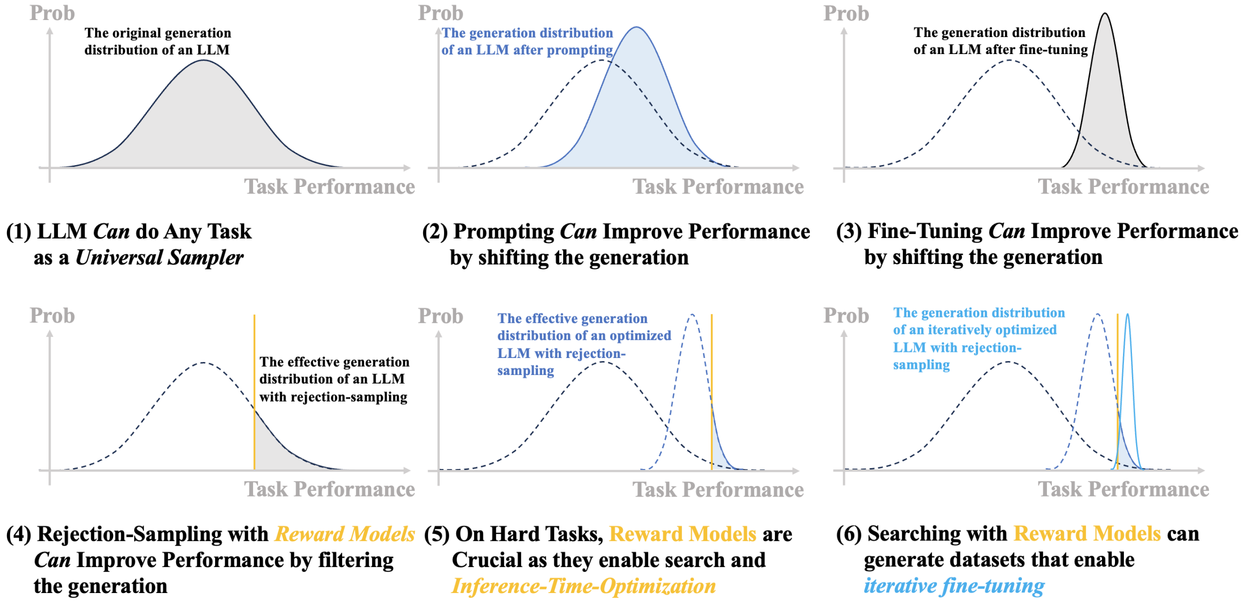
Method	$\boldsymbol{f(u)}$	**Divergence**	$\boldsymbol{D_f(\rho^{\text{demo}}\|\|\rho^{\pi})}$
AIRL (Fu et al., 2017)	$-\log u$	Reverse KL	$\text{KL}(\rho^{\pi}\|\|\rho^{\text{demo}})$
GAIL (Ho & Ermon, 2016)	$-(u+1)\log\frac{1+u}{2}+u\log u$	Jensen-Shannon	$\text{JS}(\rho^{\pi}\|\|\rho^{\text{demo}})$
FAIRL (Ghasemipour et al., 2020)	$u\log u$	Forward KL	$\text{KL}(\rho^{\text{demo}}\|\|\rho^{\pi})$

Take-aways **(1). The access to environmental dynamics is essential.** It enables distributional matching besides the offline BC objective, hence alleviating the distributional shift and compounding error problems. **(2). Reward Models in IRL are not unique.** Different assumptions lead to different reward models. We will demonstrate such a point in Section 4.

3 Optimizing LLMs beyond Imitation: Why do we Need Neural Reward Models

3.1 LLMs as Language Imitators

Given a training corpus, the LLM pre-training and Supervised Fine-Tuning are both performing next token prediction tasks (Radford et al., 2018). Empirically we know that when data, compute, and model scales, those pre-trained models begin to obtain emergent abilities of understanding and comprehending zero-shot complex tasks (Kaplan et al., 2020; Wei et al., 2022a; Kojima et al., 2022). From the perspective of RL from behavior dataset, such pre-training and SFT processes are imitating the behavioral datasets through BC (Srivastava et al., 2022; Sun & van der Schaar, 2024). In panel (1) of Figure 1, we illustrate the LLM generation as a distribution on different task-specific performance. The objective of LLM optimization is to shift the performance distribution to have higher scores on average (i.e., shift the distribution to the right side on the x axis). Since LLMs are conditional generators, asking the same question with different prompting


Prob
The original generation distribution of an LLM
Task Performance
Prob
The generation distribution of an LLM after prompting
Task Performance
Prob
The generation distribution of an LLM after fine-tuning
Task Performance
Prob
The effective generation distribution of an LLM with rejection-sampling
Task Performance
Prob
The effective generation distribution of an optimized LLM with rejection-sampling
Task Performance
Prob
The generation distribution of an iteratively optimized LLM with rejection-sampling
Task Performance

(1) LLM *Can* do Any Task as a *Universal Sampler*

(2) Prompting *Can* Improve Performance by shifting the generation

(3) Fine-Tuning *Can* Improve Performance by shifting the generation

(4) Rejection-Sampling with *Reward Models* *Can* Improve Performance by filtering the generation

(5) On Hard Tasks, Reward Models are Crucial as they enable search and *Inference-Time-Optimization*

(6) Searching with Reward Models can generate datasets that enable *iterative fine-tuning*

Figure 1: *A comparison of different LLM generation optimization approaches.* The first row represents (1) direct generation, (2) prompt optimization, (3) Supervised Fine-Tuning (SFT) on a high-quality dataset. The second row represents methods that leverage reward models (i.e., the IRL approach): (4) reward models can be used to filter out low-quality generations, (5-6) reward models can be combined with prompt optimization or fine-tuning methods to improve the generation quality. **Only reward models enable inference time optimization.**

strategies can lead to a huge difference in task performance. For instance, the chain of thought prompting was known to be very successful in improving the model's ability to solve reasoning tasks (Wei et al., 2022b) — as illustrated by panel (2) of Figure 1. While simple and effective, engineering prompts can be costly since those optimal prompt strategies are always model- and query-dependent (Sun et al., 2023a; Yang et al., 2023). An alternative approach to prompting — when having a high-quality demonstration dataset — is to conduct SFT on those relatively small datasets. LLMs can do decently well on few-shot learning (Brown et al., 2020).

Given that behavior cloning, fine-tuning, and prompting have shown strong empirical performance in aligning LLMs, one may ask: why do we need Inverse RL for alignment? Why do we need explicit reward models? In the following, we will discuss 3 motivations for learning explicit reward models (i.e., optimizing LLMs using IRL).

3.2 Reward Models in Conversational AI: Toward Practical and Scalable Learning from Human Feedback

In real-world deployments of chat models, it is common to collect preference-based feedback from users to improve response quality (Ouyang et al., 2022). While asking users to directly provide demonstrations (i.e., ideal responses) would be desirable in principle, doing so at scale is prohibitively expensive and cognitively demanding.

Practically, users and annotators are often more capable at discriminative tasks, such as choosing the better of two options, than at generative tasks such as writing full responses. This aligns with findings in prior work (Brown et al., 2019), which show that collecting preference data is a more practical and scalable way to build reward models and achieve super-demonstrator performance.

As a result, learning reward models from preference data has become a core technique in large-scale RLHF pipelines, enabling practical supervision without the need for expert-level demonstrations.

> Take-away Demonstration data is not always available. Preference data is more scalable and practical. IRL enables flexible acquisition of data and annotation for learning from behavioral datasets and implicit metrics.

3.3 Reward Models in Mathematical Reasoning: Learning Generalizable Reasoning Skills from Math

The second case where we need a reward model is mathematical reasoning. In those tasks, conducting SFT on demonstrative datasets may improve accuracy on seen question types but often fails to generalize to consistent and effective reasoning patterns. This limitation stems from the inherent difficulty of capturing complex, compositional reasoning behaviors via static datasets and imitation.

In such tasks, reward models offer a flexible mechanism for generalization. Instead of directly mimicking demonstrations, LLMs can be optimized to **explore and discover high-reward reasoning trajectories**, guided by feedback encoded in the reward model. Systems such as DeepSeek-R1 (Guo et al., 2025) demonstrate that using rule-based or learned reward models enables LLMs to exhibit behaviors such as deep thinking, long chains of reasoning, and self-correction capabilities that are difficult to induce through imitation alone.

> Take-away Reward models enable generalization in math reasoning tasks. Data-driven reward functions allow RL algorithms to discover and reinforce generalizable reasoning behaviors such as deep thinking, long chain-of-thought, and self-correction.

3.4 Reward Models for Test-Time Optimization

A unique advantage of reward models lies in their ability to support **inference-time (test-time) optimization**. While prompting and SFT can enhance task-specific performance, these approaches typically operate offline, and the improvements achieved during training cannot be adapted during test-time generation.

In classical reinforcement learning tasks, not every setting requires inference-time optimization. In relatively simple environments such as MuJoCo locomotion or Atari games, inference often consists of a single forward pass through a trained policy network. In contrast, more complex tasks such as Go require test-time optimization, where search-based planning guided by value estimators is critical for achieving superhuman performance.

Similarly, in LLM generation tasks, reward models can be used to enable inference-time optimization. For example, given a trained reward model (illustrated as the golden vertical line in panel (4) of Figure 1), candidate generations can be evaluated and low-quality outputs filtered out during inference. In high-stakes domains such as mathematical reasoning or instruction-following, this enables hybrid strategies that combine prompting, supervised fine-tuning, and reward-model-based optimization to improve test-time performance.

Take-away Reward models enable inference-time (test-time) optimization by scoring and filtering generated outputs. This allows LLMs to adaptively select high-quality responses during deployment, analogous to test-time planning in classical RL tasks.

4 From Real World Evidences to Alignment: Practical IRL via Reward Modeling

Alignment fundamentally concerns ensuring that machine behavior is consistent with the real world and its implicit objectives. The world is rich with observable signals: demonstrations, preferences, behaviors, and choices, which reflect underlying goals and constraints. These signals serve as the evidence upon which alignment should be based.

Achieving such alignment requires learning from real-world data rather than relying solely on manually specified objectives. In this context, a practical and principled approach is to infer reward functions from observed behavior. This allows us to translate real-world evidence into actionable objectives for learning and decision-making.

Crucially, real-world data is often noisy, partial, or biased. Human behavior may be suboptimal, inconsistent, or poorly articulated. Nevertheless, such data remains one of the most informative sources for guiding LLM post-training. By extracting structure and intent from this evidence, we can better align models with actual human goals, even when those goals are not explicitly stated.

This section focuses on how post-training can be operationalized through IRL, particularly by leveraging observed behavior to build effective reward models (RMs).

4.1 Reward Modeling from Preference Feedback

From PPO to DPO: Reinforcement Learning from Human Feedback (RLHF) as IRL Reinforcement Learning from Human Feedback (RLHF) has become a standard paradigm for aligning large language models (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022a;b). The core idea involves learning a reward model from human preference data and then using this model to guide policy optimization.

The training data typically consists of pairwise preferences over model outputs: $\mathcal{D}_{\text{pref}} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$, where x_i is a query and y_i^+, y_i^- denote the preferred and less-preferred responses, respectively. To convert these comparisons into scalar reward signals, models such as Bradley-Terry (Bradley & Terry, 1952) or logistic preference models are employed. These assign relative scores to responses such that $r(y_i^+) > r(y_i^-)$, enabling reward modeling through pairwise loss functions.

RLHF with Bradley-Terry Reward Models (Christiano et al., 2017) In standard RLHF, a reward model $r_\theta : (x, y) \mapsto \mathbb{R}$ is trained to reflect human preferences. Given a dataset of pairwise comparisons $\mathcal{D}_{\text{pref}} = \{(x_i, y_i^+, y_i^-)\}$, where y_i^+ is preferred over y_i^- for query x_i, the reward model is optimized via the Bradley-Terry likelihood:

$$\mathcal{L}_{\text{BT}}(\theta) = \sum_{i=1}^{N} \log \sigma \left(r_\theta(x_i, y_i^+) - r_\theta(x_i, y_i^-) \right),$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic function. This encourages the model to assign higher reward values to preferred responses.

The learned reward model is then frozen and used to supervise policy optimization via Proximal Policy Optimization (PPO) (Schulman et al., 2017). PPO maximizes the expected reward predicted by the learned reward model while constraining the updated policy to remain close to a reference policy using a KL-divergence penalty. We elaborate on policy optimization details in Section 4.4.

In contrast, Direct Preference Optimization (DPO) (Rafailov et al., 2024) sidesteps the explicit reward modeling and trajectory sampling steps altogether. Instead, it directly optimizes the policy to satisfy pairwise preference constraints derived from human feedback, using a KL-regularized classification-style objective over prompt-response pairs. This leads to a simpler and empirically more stable training process compared to PPO-based RLHF pipelines.

Direct Preference Optimization (DPO) (Rafailov et al., 2024) DPO reinterprets preference-based RLHF as a probabilistic inference problem. The key idea is to start from a latent reward function and derive a policy learning objective that avoids explicitly modeling the reward.

Assume a latent reward function $r(x, y)$ governs human preferences via a Bradley-Terry model:

$$P(y^+ \succ y^- \mid x) = \frac{\exp(r(x, y^+))}{\exp(r(x, y^+)) + \exp(r(x, y^-))}.$$

The optimal policy π^* can be derived [a] as:

$$\pi^*(y \mid x) = \frac{\exp(\beta r(x, y))}{Z(x)}, \quad \text{where } Z(x) = \sum_{y'} \exp(\beta r(x, y')).$$

Then the reward difference can be rewritten in terms of the optimal policy:

$$r(x, y^+) - r(x, y^-) = \frac{1}{\beta}\left[\log \pi^*(y^+ \mid x) - \log \pi^*(y^- \mid x)\right].$$

DPO approximates π^* with a learnable policy π_ϕ, and directly maximizes the likelihood of human preferences:

$$\mathcal{L}_{\text{DPO}}(\phi) = \sum_{i=1}^{N} \log \sigma\left(\beta\left[\log \pi_\phi(y_i^+ \mid x_i) - \log \pi_\phi(y_i^- \mid x_i)\right]\right),$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic function.

[a] For derivation details, please refer to (Peters & Schaal, 2007; Wang et al., 2018; Yang et al., 2022; Peng et al., 2019).

This objective avoids reward model training by directly adjusting the policy's log-probabilities to match observed preferences, while implicitly capturing the reward structure through relative likelihoods.

From the perspective of IRL, both RLHF and DPO can be viewed as IRL methods, as they involve inferring preferences or underlying objectives from human feedback. Recent studies (Xu et al., 2024; Ivison et al., 2024) have shown that reward-model-based RLHF can outperform DPO when PPO hyperparameters are properly tuned. However, stabilizing PPO remains non-trivial in practice (Rafailov et al., 2024), and DPO tends to be more robust to overoptimization (Ivison et al., 2024). Recent work has explored hybrid approaches that aim to combine the strengths of both methods (Zhong et al., 2024). Alternatively, one may consider iterative DPO as an online variant of the original DPO, and achieve improved learning efficiency while sustaining high stability (Xiong et al., 2023). Recent advances given in Shi et al. (2025) further provide theoretical insights on the priority of different approaches.

Take-away DPO offers superior training stability and requires less hyperparameter tuning, making it a robust choice for alignment. In contrast, PPO with explicit reward modeling can outperform DPO when carefully tuned. The selection between the two should be guided by task sensitivity and available computational resources.

In the seminal work on RLHF (Christiano et al., 2017), pairwise preference annotations are translated into scalar reward scores using the Bradley-Terry (BT) model (Bradley & Terry, 1952), enabling scalar supervision for aligning LLMs (Stiennon et al., 2020). However, the choice of BT has traditionally been made on heuristic grounds, and alternative preference models such as the Kahneman-Tversky ordinal model (Ethayarajh et al., 2024) and general discrete choice models (Azar et al., 2024) have since been proposed.

Sun et al. (2024b) provides a formal justification for the use of BT models in LLM alignment. Crucially, it distinguishes between *BT parameter estimation* (Bradley & Terry, 1952), which assumes direct access to latent utilities, and *BT regression* (Springall, 1973; Bockenholt, 1988), which instead regresses reward scores from learned input representations. The key insight is that modern reward models operate in the *embedding space* of pre-trained LLMs, making BT regression more appropriate. This perspective also provides a theoretical grounding for recent work that reuses or fine-tunes language model embeddings for reward modeling (Sun et al., 2025a; Shen et al., 2025; Sun et al., 2023a).

Additionally, Sun et al. (2024b) introduces the notion of *order consistency* as a more suitable learning objective in preference-based settings. That is, for tasks such as best-of-N selection, the relative ordering of responses is more important than the absolute value of their scores. While the BT model satisfies this property, it is not uniquely suited for the task. Simpler alternatives, such as binary classification models that treat preferred responses as positives and dispreferred ones as negatives, can also achieve order-consistent objectives, and often perform better in the presence of noisy or ambiguous annotations.

Take-away (1) The theoretical foundation of modern preference-based reward modeling is better captured by Bradley-Terry *regression* (Springall, 1973) than classical BT estimation (Bradley & Terry, 1952). (2) Classification-based objectives can outperform BT models, particularly in the presence of annotation noise, while still preserving the crucial property of order consistency.

Active Learning Building on this foundation, a central challenge in practical reward modeling is the efficient acquisition of preference annotations. Since reward models are trained to preserve the order of responses in embedding space, annotation strategies should prioritize comparisons that are most informative for determining rank. This has motivated the application of active learning in preference data collection.

Recent studies (Muldrew et al., 2024; Mukherjee et al., 2024) have proposed a variety of heuristic-inspired acquisition functions tailored to reward modeling. One commonly used approach is uncertainty sampling, which selects response pairs where the reward model is least confident in its preference. Another is maximum difference sampling, which selects pairs with the highest predicted reward gaps, under the assumption that such examples yield more reliable supervision.

A more principled formulation of active preference learning is presented by Shen et al. (2025); Feng et al. (2025), which draws on tools from Fisher information theory and optimal experiment design. Instead of relying on heuristic acquisition functions, those papers propose to select query pairs that maximize the determinant of Fisher information with respect to the reward model parameters in the embedding space.

Fisher-Information Guided Preference Annotation (Feng et al., 2025; Shen et al., 2025)
Consider the linear BT regression models (on the embedding space), $r(x, y) = w^T\phi(x, y)$. The preference generation process of the i-th pair h_i is

$$h_i \sim \text{Bernoulli}[\sigma[w^T(\phi(x, y_1) - \phi(x, y_2)]]$$

Based on the theory fro generalized linear models, the maximum likelihood estimate $\hat{w}$ is asymptotically Gaussian distributed, with mean w and covariance matrix $\mathcal{I}^{-\infty}$, where $\mathcal{I}$ denotes the Fisher Information (FI) matrix.

$$\mathcal{I} = \sum_{i=1}^{I}(\phi(x_i, y_{i,1}) - \phi(x_i, y_{i,2}))^T(\phi(x_i, y_{i,1}) - \phi(x_i, y_{i,2}))p_i(1 - p_i)$$

where $p_i = \sigma[w^T(\phi(x_i, y_{i,1}) - \phi(x_i, y_{i,1}))]$. Following the classical methods *Bayesian D-Optimality* design (Chaloner & Verdinelli, 1995), preference annotations should prioritize those samples with the highest scores:

$$\mathcal{S}_{\text{D}-\text{Opt}} = |\mathcal{I}|$$

This approach formalizes the goal of active learning as maximizing informativeness under a limited annotation budget. Importantly, their framework highlights an inherent *exploration–exploitation trade-off*: selecting pairs that are highly uncertain (exploration) versus those that are expected to provide strong gradients for refining the current model (exploitation). Their method operates entirely in the embedding space, aligning well with recent theoretical insights about the structure of reward models and their reliance on pretrained LLM representations.

Take-away **Building on top of linear BT models**, Fisher Information and optimal experimental design provide a theoretically grounded framework for active preference learning, highlighting the need to balance exploration and exploitation.

Diverse Preferences and Personalization in Reward Modeling Beyond active sampling, another critical challenge in preference-based reward modeling is accounting for *preference diversity*. In practice, human preferences vary significantly across users, tasks, and deployment settings. A single global reward model may fail to capture such heterogeneity, leading to poor generalization and potential misalignment with specific user intents (Sorensen et al., 2024).

To address this, recent works have explored personalized reward modeling through a variety of techniques. One direction is to explicitly learn user-specific latent variables that condition reward predictions (Poddar et al., 2024; Li et al., 2024b; Kobalczyk et al., 2024). Others model reward distributions rather than point estimates, enabling uncertainty-aware reasoning over latent contextual factors (Siththaranjan et al., 2023). Moreover, Chakraborty et al. (2024) introduces a MaxMin training objective to align models with a diverse set of human preferences by optimizing worst-case reward performance across subgroups. Recent work of Luo et al. (2025) innovates the usage of Principal Component Analysis (PCA) for lightweight personalized preference learning in the embedding space.

Decomposed Reward Models (DRMs) (Luo et al., 2025) Given a comparison triple (x, y^+, y^-), where y^+ is the preferred response over y^-, the standard Bradley-Terry (BT) objective under vector representation space is:

$$\max_{\mathbf{w}} \mathbb{E}\left[\log \sigma\left(\mathbf{w}^\top (\phi(x, y^c) - \phi(x, y^r))\right)\right],$$

where $\phi(x, y)$ denotes the feature embedding of response y conditioned on input x, $\mathbf{w}$ is a preference vector, and $\sigma(\cdot)$ is the sigmoid function. Let $\Delta\phi_t = \phi(x_t, y_t^+) - \phi(x_t, y_t^-)$ and define the centered difference vector $z_t = \Delta\phi_t - \mathbb{E}[\Delta\phi]$. Consider the PCA over z_t, with the covariance matrix of feature differences:

$$\Sigma = \mathbb{E}[z_t z_t^\top].$$

By decomposing Σ, we obtain a set of orthogonal basis vectors in the embedding space that capture the main axes of variation in human preferences. Instead of modeling reward using a single vector $\mathbf{w}$, we define d orthogonal reward heads:

$$W = [\mathbf{w}_1, \ldots, \mathbf{w}_d] \in \mathbb{R}^{h \times d},$$

where each $\mathbf{w}_k$ corresponds to a principal direction extracted via PCA. The reward vector is then:

$$\mathrm{DRM}(x, y) = W^\top \phi(x, y) \in \mathbb{R}^d,$$

which decomposes the original reward into d interpretable components.

DRM is shown to be highly interpretable with preference attributes. Different reward heads specialize in different attributes, and the first head aligns with the majority preference, while others focus on different aspects.

Take-away **Building on top of linear BT models**, diverse human preferences can be expressed as vectors in the embedding space, and DRMs model them using a set of orthogonal basis vectors. Such an approach offers a systematic way to understand human preferences by breaking complex preferences into interpretable parts.

4.2 Reward Modeling for Mathematical Reasoning

Revisiting the History of LLM-based Math Reasoning Research LLMs have demonstrated strong competence in mathematical reasoning, yete methods for eliciting this capability have evolved rapidly. Early approaches centered around *prompt optimization*, most notably Chain-of-Thought (CoT) prompting, which encourages step-by-step reasoning to improve final answer accuracy (Wei et al., 2022b). Subsequent variants such as zero-shot CoT (Kojima et al., 2022), self-consistency decoding (Wang et al., 2022), and Tree-of-Thoughts (ToT) prompting (Yao et al., 2023) expanded the space of inference-time strategies, showing that multi-step reasoning could be induced without altering model parameters. These techniques primarily operated at inference time and revealed that models possess latent reasoning abilities that can be surfaced with minimal intervention.

Despite their success, prompt-based methods are shown to be model-dependent (Yang et al., 2023), and the black-box heuristics can not systematically detect or correct errors. This motivated a more structured paradigm based on search and planning with dense rewards (Chan et al., 2024; Lightman et al., 2023). Inspired by classical AI planning, these methods use algorithms such as Monte Carlo Tree Search (MCTS) to explore candidate reasoning paths, evaluating partial solutions with learned reward models or value functions (Zhang et al., 2024b). By providing fine-grained feedback, these methods enable models to plan and search for optimized thoughts (Pouplin et al., 2024). While leveraging dense reward and MCTS may improve the math reasoning abilities (Wang et al., 2023c), it also introduces new challenges related to dense reward design, computational efficiency, and vulnerability to reward hacking (Guo et al., 2025; Gao et al., 2023).

More recently, the field has shifted toward reinforcement learning with verifiable rewards (RLVR), which leverages the fact that correctness in mathematical reasoning is often easily verifiable. Models such as DeepSeek-r1 have been trained using sparse but reliable correct-wrong signals, leading to significantly improved reasoning capabilities (Guo et al., 2025; Jaech et al., 2024). These models exhibit long, internally consistent chains of thought and frequently demonstrate behaviors such as self-reflection and backtracking. By directly optimizing for correctness, this paradigm departs from preference-based RLHF approaches and moves toward grounded, data-driven learning, without the need for neural reward models.

Evolving Understanding the Performance Gain from RL: Importance of Structure and Format Despite being framed as reinforcement learning, many recent advances in RLVR-based mathematical reasoning seem to benefit less from exploration in an RL environment, and more from the model's alignment with effective response formats. Studies such as Shao et al. (2025); Wang et al. (2025b) demonstrate that even spurious or minimal reward signals can significantly improve model performance. These improvements are often attributed not to the discovery of fundamentally new reasoning strategies but rather to the emergence of structured, verifiable, and execution-friendly templates, such as programmatic responses, long chain-of-thought derivations, or format-constrained contents.

In this light, RLVR can be viewed as a mechanism for internalizing template-level prompt optimization: unlike inference-time prompting strategies that guide structure externally, RLVR encourages the model to internalize such structures through training. This shift underscores a deeper insight — that for complex reasoning tasks, such as math or code, the structure of the answer is as important as its content. The field may thus be entering a phase where format and reasoning inductive bias take center stage, even within the RL framework.

This convergence between RLVR and earlier prompt-based methods underscores the importance of revisiting prompt optimization, not merely as a heuristic, but as a principled framework for guiding model behavior through structured format design. In the following section, we examine recent advances on IRL-based prompt optimization for reasoning tasks, with a focus on the role of reward models in such a process.

Revisiting Prompt Optimization: Building Proxy Verifiers from Prompting Experience In the field of prompt optimization, recent work has explored a diverse range of strategies to enhance the problem-solving capabilities of large language models. These include CoT (Kojima et al., 2022), ToT (Yao et al., 2023), in-context optimization (Cui et al., 2024), multi-agent debate frameworks (Smit et al., 2023; Du et al., 2023; Liang et al., 2023), task decomposition (Khot et al., 2022; Zhou et al., 2022a),and automated prompt search (Pryzant et al., 2023; Guo et al., 2023), including approaches that leverage LLMs themselves to optimize prompts (Zhou et al., 2022b; Yang et al., 2023). For a comprehensive overview of this line of research, we refer readers to the recent surveys by Li et al. (2025) and Cui et al. (2025).

Among these methods, automated prompt optimization techniques that interact directly with the task environment by querying the LLM and receiving verifiable rewards often achieve strong performance without relying on explicit reward models. However, because they require repeated interactions with black-box models, these methods are computationally expensive and often impractical in real-world settings. Although effective, such approaches do not utilize reward models and do not exploit existing offline data to reduce interaction costs.

To address this limitation, Prompt-OIRL(Sun et al., 2023a) proposes a simple and cost-effective IRL-based method that reuses historical prompting trial-and-error experience to train a reward model for offline prompt evaluation and evaluation. Prompt-OIRL enables adaptive, query-dependent prompt selection without requiring additional calls to the LLM at inference time. This provides a practical and scalable solution to prompt optimization in settings where interaction cost is a bottleneck. The algorithm proceeds as follows:

Prompt Optimization with Offline IRL (Sun et al., 2023a) Prompt-OIRL builds upon prior work in prompt optimization by reusing experimental artifacts. Given an open-source query q from a dataset $\mathcal{D}_{\text{q}}$, a set of prompt candidates $p \in \mathcal{P}$ (either as prefix or suffix), and correctness labels $r^{(p,q)}$ indicating whether applying prompt p to query q yields a correct answer (i.e., $r^{(p,q)} = 1$ if correct, and 0 otherwise), the method constructs a reward-labeled dataset for training.

In the *Reward Modeling* phase, a reward model $\Upsilon_\theta^{(p,q)}$, parameterized by θ, is trained to predict $r^{(p,q)}$ using a cross-entropy loss:

$$\mathcal{L}_{\text{CE}}(\theta;\mathcal{P},\mathcal{D}_{\text{q}}) = -\mathbb{E}_{p\in\mathcal{P},q\in[\mathcal{D}_{\text{q}}]}\left[r^{(p,q)}\log\sigma\left(\Upsilon_\theta(p,q)\right) + (1-r^{(p,q)})\log\left(1-\sigma\left(\Upsilon_\theta(p,q)\right)\right)\right]$$

Then in the *Prompt Optimization* phase, the reward model Υ_θ is used as a proxy to optimize prompt p^* for any given query q_i:

$$p^* = \arg\max_{p} \Upsilon_\theta(p, q_i)$$

Prompt-OIRL has shown significant improvement on mathematical reasoning tasks *through reward modeling* for prompt optimization.

Take-away Research in LLM-based mathematical reasoning has evolved from heuristic prompting to RLVR, yet recent findings suggest that RL's effectiveness often arises from structured response formats such as templating. Viewed as a general form of prompting, templating bridges RLVR and prompt optimization, highlighting the potential of automated methods like Prompt-OIRL for future progress.

4.3 Reward Modeling from Demonstration Datasets

While RLHF from preference learning and verifiable reward has demonstrated great success in aligning LLMs according to user intention or factual correctness, such data with binary identifiable labels is not universally applicable. Only a limited subset of tasks has a clear objective answer that is verifiable; for the majority, user-centered subjective evaluation is always essential.

Among subjective feedback types, preference data has become the most widely used, largely due to its scalability in practical annotation workflows. However, collecting high-quality preference annotations poses several challenges, including annotation noise and ambiguity (Zheng et al., 2023), high labeling costs (Guo et al., 2024; Xiong et al., 2023; Shen et al., 2025), and potential privacy concerns when sharing data with annotators (Li et al., 2023a; Pouplin et al., 2024).

Beyond preferences, alternative feedback modalities such as demonstrations, scalar judgments, and critiques can often provide richer supervision, particularly in personalized or open-ended tasks (Tandon et al., 2021; Shi et al., 2022; Scheurer et al., 2022; Xiao et al., 2024; Li et al., 2024a; Chen et al., 2024b; Sun et al., 2025b). Recent work has explored Alignment from Demonstration (AfD) (Sun & van der Schaar, 2024), which learns reward models from expert demonstrations rather than pairwise comparisons. This direction aligns naturally with classical IRL literature. The most straightforward approach to AfD, like any IRL task, is BC. And recent works on AfD mainly work on going beyond such an approach.

Formally, Sun & van der Schaar (2024) revisited the occupancy matching problem of IRL Ho & Ermon (2016); Ross et al. (2011); Fu et al. (2017); Orsini et al. (2021) to enhance the performance of AfD. Using $\rho^\beta(s,a) = \pi_\beta(a|s)\sum_{t=0}\gamma^t \mathrm{Prob}(s_t = s|\pi_\beta)$ to denote the state-action occupancy measure of the behavior policy (i.e., the demonstrator), and $\rho^\pi(s,a)$ the state-action occupancy measure of the current policy. In the context of LLM generation, with x the input query and $y = (y^{(0)}, y^{(1)}, ..., y^{(T)} = \texttt{EOS})$ the output response containing a maximum of $T+1$ tokens, the occupancy measure is

$$\begin{aligned}\rho^\pi(s_k, a_k) &= \rho^\pi(s_k = (x, y^{(0:k-1)}), a_k = y^{(k)}) \\ &= \pi(a_k = y^{(k)}|s_k = (x, y^{(0:k-1)}))p(s_k) \\ &= ... \\ &= p(s_0)\Pi_{t=0}^{t=k}\pi(a_t = y^{(t)}|s_t = (x, y^{(0:t-1)}))\end{aligned} \tag{5}$$

In alignment, the completed generations are of more research interest. Denoting 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), we have

$$\begin{aligned}d^\pi(y|x) = \Pi_{t=0}^{t=T}\pi(a_t = y^{(t)}|s_t = (x, y^{(0:t-1)})) = \rho^\pi(s_T, a_T)/p(x), \\ d^\beta(y|x) = \Pi_{t=0}^{t=T}\pi_\beta(a_t = y^{(t)}|s_t = (x, y^{(0:t-1)})) = \rho^\beta(s_T, a_T)/p(x),\end{aligned} \tag{6}$$

for the current policy and behavior policy, individually. From a divergence minimization perspective, we have

Divergence Minimization Perspectives of AfD (Sun & van der Schaar (2024)

1. **Forward KL: SFT.** Consider the objective using the **forward KL divergence** between the demonstration and policy conditional trajectory distributions:

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

   This corresponds to the SFT objective $\mathcal{L}_{\mathrm{SFT}} = -\max_\pi \mathbb{E}_{(s,a)\sim\mathcal{D}_{\mathrm{demo}}}\left[\log(\pi(a|s))\right]$.

2. **Reverse KL: Adversarial Imitation.** Instead, minimizing the **Reverse KL divergence** leads to the following learning objective:

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

   Using generative adversarial methods to estimate the second term $d^\beta(y|x)$ with a parameterized discriminative model D_ϕ, and optimizing it with

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

   at convergence, the policy learning objective becomes

$$\max_\pi \mathbb{E}_{(y|x)\sim d^\pi}\left[r(y|x)\right],$$

   where

$$r(y|x) = \log D_\phi(y|x) - \log(1 - D_\phi(y|x))$$

   is the parameterized reward model.

From those derivations, we see that both SFT and Reward Modeling are instantiations of divergence minimization in AfD. As the forward KL and reverse KL divergences have mass-covering and mode-seeking properties, those different objectives also lead to different model behaviors after alignment (Chu et al., 2025).

Take-away Beyond preference-based RLHF, alignment from demonstrations (AfD) offers a principled alternative for learning from richer supervision. By formalizing AfD through occupancy matching and divergence minimization, recent work shows that both SFT and reward modeling can be understood as special cases, paving the way for more general and theoretically grounded alignment methods.

4.4 Improving LLM Generation with Reward Models

We discuss in this section the techniques that optimize LLM outputs using learned reward models. These methods vary in terms of whether they require model fine-tuning, whether they rely on learned value estimators, and the stage at which reward feedback is incorporated (training-time or inference-time). Table 5 summarizes key approaches across this landscape.

Method	Fine-Tune	Value-Estimator	Example Work	Properties
Best-of-N	No	No	Stiennon et al. (2020); Gao et al. (2023); Gui et al. (2024)	Simple to implement; no training needed; improves output quality. Computationally expensive in inference.
Iterative Tuning	Yes	No	Dong et al. (2023); Yuan et al. (2023); Liu et al. (2023)	Stable and effective; no RL required. Iterative training limits parallelization.
PPO (Classical RLHF)	Yes	Yes (GAE)	Ouyang et al. (2022); Stiennon et al. (2020)	Well-established and widely used. Complex to train; sensitive to hyperparameters.
Monte-Carlo	Yes	No (MC)	Li et al. (2023b); Shao et al. (2024); Yu et al. (2025)	Conceptually simple; no value network needed; strong empirical results.
Reward-Guided Decoding	No	No (RM)	Deng & Raffel (2023); Khanov et al. (2024); Liao et al. (2025); Chen et al. (2024a); Rashid et al. (2024)	Allows on-the-fly control without fine-tuning. Searching can be expensive. Performance is highly dependent on fine-grained reward model fidelity.

Table 5: Overview of generation optimization methods for LLM alignment with a reward model.

Best-of-N Sampling and RAFT: Filtering and Iterative Reranking The simplest form of reward-guided optimization is Best-of-N (BoN) sampling, where multiple candidate completions are generated, scored by a reward model, and the highest-scoring output is selected (Stiennon et al., 2020). This method is straightforward and requires no additional fine-tuning, but becomes computationally expensive for long-form generation or when N is large. From the performance perspective, BoN can achieve competitive performance when compared to the RL-based optimization techniques (Gao et al., 2023; Gui et al., 2024). And this makes BoN performance a reliable evaluation metric for reward model research (Sun et al., 2024b).

A more efficient and stable alternative is to *parameterize* the BoN policy through iterative supervised fine-tuning on reward-selected outputs (Dong et al., 2023; Yuan et al., 2023; Liu et al., 2023). Such that in the inference time, the BoN performance can be achieved without large-scale sampling. These methods refine the model by repeatedly fine-tuning on top-ranked completions from a small candidate set, effectively incorporating reward signals without the instability and complexity of the RL-based approaches. Recent discoveries on this line of research further demonstrate its strong ability compared to state-of-the-art RL algorithms in LLM post-training (Xiong et al., 2025).

PPO, REINFORCE, GRPO, DAPO: From Temporal Difference to Monte-Carlo Estimation Among training-time methods, Proximal Policy Optimization (PPO) (Schulman et al., 2017) is the most widely adopted algorithm for LLM alignment (Ouyang et al., 2022; Stiennon et al., 2020; Bai et al., 2022a). Its canonical implementation incorporates a value network and Generalized Advantage Estimation (GAE) (Schulman et al., 2015) to stabilize value propagation. However, this standard setup overlooks key differences between LLM post-training and conventional RL tasks. Unlike typical RL benchmarks (Bellemare et al., 2013; Aitchison et al., 2023; Tassa et al., 2018), LLM generation receives sparse, trajectory-level feedback. For instance, correctness in mathematical reasoning is assessed only after a full solution is generated; human preference labels are typically given at the response level in chatbot alignment tasks.

This challenge is commonly referred to as the credit assignment problem in RL literature (Pignatelli et al., 2023), which has been tackled using techniques such as reward redistribution and decomposition (Ren et al., 2021; Arjona-Medina et al., 2019), memorization-based methods (Ke et al., 2018), and attention-based mechanisms (Ferret et al., 2019). In the LLM setting, Chan et al. (2024) proposed a token-level redistribution scheme that leverages attention scores to assign trajectory-level rewards to individual tokens, thereby improving the stability and efficiency of PPO-based post-training.

Given the sparsity of rewards, another line of work sidesteps value estimation entirely by adopting Monte-Carlo based return estimation, such as REINFORCE (Li et al., 2023b) and GRPO (Shao et al., 2024), which directly optimize expected returns using trajectory-level feedback. These methods have shown strong empirical performance in tasks like mathematical reasoning and code generation. DAPO (Yu et al., 2025) further builds on GRPO with additional empirical insights, improving both stability and training efficiency.

Reward-Guided Decoding: Inference-Time Optimization without Fine-Tuning Unlike training-time methods that update model parameters, reward-guided decoding directly modifies the sampling procedure at inference time using a reward model to steer generation. These approaches operate by reweighting token probabilities based on token-level or trajectory-level reward feedback, offering a flexible alternative to policy model training.

Recent work has explored a range of reward-guided decoding strategies. RAD (Deng & Raffel, 2023) introduces a unidirectional reward model to rescore tokens during generation, improving controllability without retraining. ARGS (Khanov et al., 2024) generalizes this idea to broader alignment settings by adjusting token sampling using reward signals. PAD (Chen et al., 2024a) extends reward-guided decoding to support personalized preferences at decoding time, while RSD (Liao et al., 2025) leverages a draft model and reward evaluation to enable efficient speculative decoding.

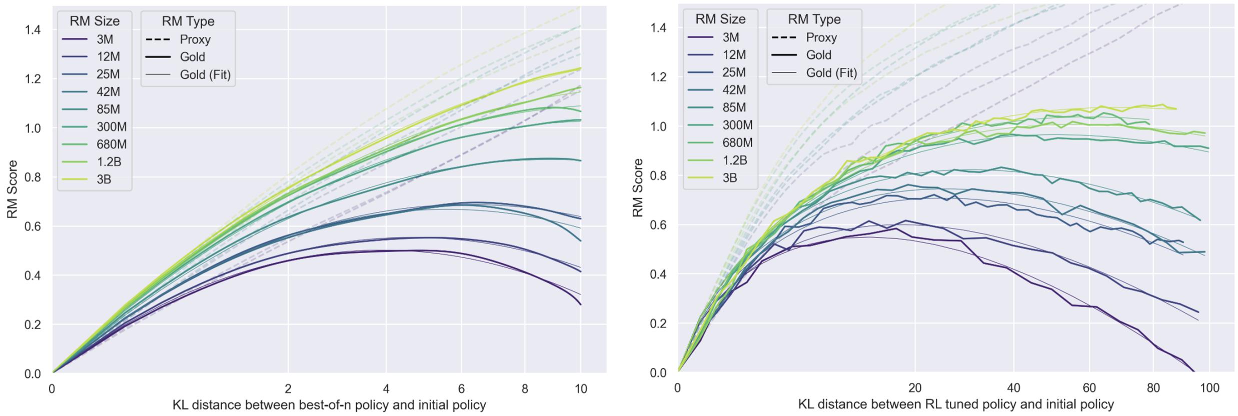
While promising, inference-time alignment methods often rely on trajectory-level rewards applied at the token level. Rashid et al. (2024) highlighted such a mismatch and addressed it by training Bradley-Terry reward models on partial sequences to derive a consistent token-level policy. In more general practices, the effectiveness of such process-based reward models may vary by task and should be evaluated accordingly.

Take-away LLM generation can be optimized using reward models either through training-time policy updates or inference-time decoding strategies. Besides the classical method of PPO, simpler alternatives such as iterative fine-tuning and Monte-Carlo value estimation based methods provide strong empirical performance with reduced complexity. The choice of method should consider reward sparsity, task structure, and computational constraints.

4.5 Risks, Challenges, and Opportunities

Reward Overoptimization Before concluding this section, we would like to discuss some challenges and opportunities in reward modeling. Since the reward models are learned from data, it may be overfitted — just like any data-driven machine learning models. The most well-known challenge is the reward hacking problem, or reward overoptimization (Gao et al., 2023). The key insight here is Goodhart's Law, which states, "When a measure becomes a target, it ceases to be a good measure." Optimizing too much against a learned reward model will eventually hinder the true objective. As illustrated in Figure 2.

To mitigate reward model overoptimization, one practical direction is to incorporate uncertainty estimation into reward modeling, often via ensemble methods (Coste et al., 2023; Ahmed et al., 2024; Zhang et al., 2024d). Another line of work focuses on regularizing the learning process by incorporating auxiliary objectives such as generative predictions to regularize value learning (Yang et al., 2024). This insight underpins the



(a) BoN

(b) RL

Figure 2: Reward model overoptimization (Figure 1 of Gao et al. (2023)). The x-axis represents the degree of optimization, measured by the KL divergence between the optimized policy and the original checkpoint. The y-axis indicates the reward score assigned by different reward models. The two panels correspond to different optimization methods: Best-of-N sampling and PPO-based training. Each curve color denotes a different reward model size. Across all settings, the gap between the solid line (score assigned by the optimized reward model) and the dashed line (score assigned by a held-out reference reward model) quantifies the degree of overoptimization—i.e., the extent to which the optimized policy exploits idiosyncrasies of the reward model rather than aligning with the intended objective.

framework of Generative Reward Models (GRMs) (Mahan et al., 2024; Wang et al., 2025a), which leverages the generative capabilities of LLMs to improve reward estimation in discriminative settings (Zhang et al., 2024c). More recently, Liu et al. (2025) proposes to scale inference-time computation and exploit the advanced reasoning abilities of LLMs to further enhance reward modeling performance and reliability.

Besides technical improvements, model behavior analysis can also add important insight into understanding overoptimization behaviors. In model evaluation, it has been discovered that users would prefer lengthy responses over concise ones, and such length bias can be captured by reward models (Hu et al., 2024; Wang et al., 2023b; Wu & Aji, 2023); hence, length-controlled evaluation has been widely adopted (Dubois et al., 2024). Liu et al. (2024) considered a causal approach to disentangle contextual artifacts and irrelevant signals, such that the robustness of reward models can be improved.

Data Matters: from Offline to Online Datasets The second challenge lies in the off-policy nature of available data. In many alignment settings, especially when leveraging open-source datasets, the responses are typically generated by outdated or mismatched models. Training reward models or optimizing policies on such off-policy data introduces distribution mismatch and can degrade performance (Xiong et al., 2023). Prior work has emphasized that data quality outweighs quantity under such conditions — smaller, high-quality datasets often yield better results than large but stale ones (Zhou et al., 2023a; Sun et al., 2024a).

Given limited annotation budgets, online learning or active preference collection offers a more efficient alternative to static offline datasets. As discussed in Section 4.1, principled algorithms can help optimize annotation efforts and improve reward model quality. Future work may explore methods for converting off-policy data into usable on-policy annotations for reward modeling.

Take-away Across the challenges discussed above, *generalization* to unseen prompts, responses, and even underlying LLM policies remains the central obstacle in reward modeling. Algorithmic advances have aimed to mitigate overoptimization, detect and analyze reward hacking, and leverage LLMs' reasoning capabilities to improve reward modeling. On the data-centric side, the off-policy nature of preference data presents a major bottleneck. Future research may benefit from exploring diverse feedback modalities, such as critiques, and developing methods that better bridge the gap between offline supervision and on-policy learning.

References

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

Ahmed M Ahmed, Rafael Rafailov, Stepan Sharkov, Xuechen Li, and Sanmi Koyejo. Scalable ensembling for mitigating reward overoptimisation. *arXiv preprint arXiv:2406.01013*, 2024.

Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the arcade learning environment down to five games. In *International Conference on Machine Learning*, pp. 421–438. PMLR, 2023.

AlphaProof and AlphaGeometry teams. Ai achieves silver-medal standard solving international mathematical olympiad problems. https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/, July 2024. URL https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/. DeepMind Blog.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

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

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*, 2022b.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279, 2013.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

U Bockenholt. A logistic representation of multivariate paired-comparison models. *Journal of mathematical psychology*, 32(1):44–63, 1988.

Paolo Bory. Deep new: The shifting narratives of artificial intelligence from deep blue to alphago. *Convergence*, 25(4):627–642, 2019.

Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauzá, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving generalist agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1:8, 2024.

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*, pp. 783–792. PMLR, 2019.

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

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Singh Bedi, and Mengdi Wang. Maxmin-rlhf: Alignment with diverse human preferences. *arXiv preprint arXiv:2402.08925*, 2024.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical science*, pp. 273–304, 1995.

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.

Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. Pad: Personalized alignment of llms at decoding-time. *arXiv preprint arXiv:2410.04070*, 2024a.

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*, 2024b.

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.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36:47704–47720, 2023.

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

Wendi Cui, Jiaxin Zhang, Zhuohang Li, Hao Sun, Damien Lopez, Kamalika Das, Bradley Malin, and Sricharan Kumar. Phaseevo: Towards unified in-context prompt optimization for large language models. *arXiv preprint arXiv:2402.11347*, 2024.

Wendi Cui, Jiaxin Zhang, Zhuohang Li, Hao Sun, Damien Lopez, Kamalika Das, Bradley A Malin, and Sricharan Kumar. Automatic prompt optimization via heuristic search: A survey. *arXiv preprint arXiv:2502.18746*, 2025.

Haikang Deng and Colin Raffel. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. *arXiv preprint arXiv:2310.09520*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

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.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023.

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

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

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

Yunzhen Feng, Ariel Kwiatkowski, Kunhao Zheng, Julia Kempe, and Yaqi Duan. Pilaf: Optimal human preference sampling for reward modeling. *arXiv preprint arXiv:2502.04270*, 2025.

Johan Ferret, Raphaël Marinier, Matthieu Geist, and Olivier Pietquin. Self-attentional credit assignment for transfer in reinforcement learning. *arXiv preprint arXiv:1907.08027*, 2019.

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

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

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Scott Fujimoto, Wei-Di Chang, Edward Smith, Shixiang Shane Gu, Doina Precup, and David Meger. For sale: State-action representation learning for deep reinforcement learning. *Advances in neural information processing systems*, 36:61573–61624, 2023.

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

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

Lin Gui, Cristina Gârbacea, and Victor Veitch. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *arXiv preprint arXiv:2406.00832*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*, 2023.

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.

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.

Gillian M Hayes and John Demiris. *A robot controller using learning by imitation*. University of Edinburgh, Department of Artificial Intelligence Edinburgh, UK, 1994.

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.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

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.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. Explaining length bias in llm-based preference evaluations. *arXiv preprint arXiv:2407.01085*, 2024.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. *arXiv preprint arXiv:2406.09279*, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

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.

Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang. Towards efficient exact optimization of language model alignment. *arXiv preprint arXiv:2402.00856*, 2024.

Shengyi Jiang, Jingcheng Pang, and Yang Yu. Offline imitation learning with a misspecified simulator. *Advances in neural information processing systems*, 33:8510–8520, 2020.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can llms actually correct their own mistakes? a critical survey of self-correction of llms. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. In *Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14*, pp. 313–329. Springer, 2021.

Nan Rosemary Ke, Anirudh Goyal ALIAS PARTH GOYAL, Olexa Bilaniuk, Jonathan Binas, Michael C Mozer, Chris Pal, and Yoshua Bengio. Sparse attentive backtracking: Temporal credit assignment through reminding. *Advances in neural information processing systems*, 31, 2018.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. Args: Alignment as reward-guided search. *arXiv preprint arXiv:2402.01694*, 2024.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*, 2022.

Katarzyna Kobalczyk, Claudio Fanconi, Hao Sun, and Mihaela van der Schaar. Few-shot steerable alignment: Adapting rewards and llm policies with neural processes. *arXiv preprint arXiv:2412.13998*, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

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.

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*, 2023a.

Jiaxiang Li, Siliang Zeng, Hoi-To Wai, Chenliang Li, Alfredo Garcia, and Mingyi Hong. Getting more juice out of the sft data: Reward learning from human demonstration improves sft for llm alignment. *Advances in Neural Information Processing Systems*, 37:124292–124318, 2024a.

Wenwu Li, Xiangfeng Wang, Wenhao Li, and Bo Jin. A survey of automatic prompt engineering: An optimization perspective. *arXiv preprint arXiv:2502.11560*, 2025.

Xinyu Li, Ruiyang Zhou, Zachary C Lipton, and Liu Leqi. Personalized language modeling from personalized human feedback. *arXiv preprint arXiv:2402.05133*, 2024b.

Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023b.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. Reward-guided speculative decoding for efficient llm reasoning. *arXiv preprint arXiv:2501.19324*, 2025.

Q Vera Liao and Jennifer Wortman Vaughan. Ai transparency in the age of llms: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*, 10, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL `https://transformer-circuits.pub/2025/attribution-graphs/biology.html`.

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

Tianqi Liu, Wei Xiong, Jie Ren, Lichang Chen, Junru Wu, Rishabh Joshi, Yang Gao, Jiaming Shen, Zhen Qin, Tianhe Yu, et al. Rrm: Robust reward model training mitigates reward hacking. *arXiv preprint arXiv:2409.13156*, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling. *arXiv preprint arXiv:2504.02495*, 2025.

Feng Luo, Rui Yang, Hao Sun, Chunyuan Deng, Jiarui Yao, Jingyan Shen, Huan Zhang, and Hanjie Chen. Rethinking diverse human preference learning through principal component analysis. *arXiv preprint arXiv:2502.13131*, 2025.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models. *arXiv preprint arXiv:2410.12832*, 2024.

Daniel J Mankowitz, Andrea Michi, Anton Zhernov, Marco Gelmi, Marco Selvi, Cosmin Paduraru, Edouard Leurent, Shariq Iqbal, Jean-Baptiste Lespiau, Alex Ahern, et al. Faster sorting algorithms discovered using deep reinforcement learning. *Nature*, 618(7964):257–263, 2023.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

John Menick. Move 37: Artificial intelligence, randomness, and creativity. *Mousse Magazine*, 55:53, 2016.

AI Meta. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2024.

Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. A survey of explainable reinforcement learning. *arXiv preprint arXiv:2202.08434*, 2022.

Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nova, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Subhojyoti Mukherjee, Anusha Lalitha, Kousha Kalantari, Aniket Deshmukh, Ge Liu, Yifei Ma, and Branislav Kveton. Optimal design for human preference elicitation, 2024. https://www.amazon.science/publications/optimal-design-for-human-preference-elicitation.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. *arXiv preprint arXiv:2402.08114*, 2024.

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)*, pp. 6292–6299. IEEE, 2018.

Tianwei Ni, Harshit Sikchi, Yufei Wang, Tejus Gupta, Lisa Lee, and Ben Eysenbach. f-irl: Inverse reinforcement learning via state marginal matching. In *Conference on Robot Learning*, pp. 529–551. PMLR, 2021.

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.

OpenAI. Introducing deep research. https://openai.com/index/introducing-deep-research/, 2025. Accessed: 2025-04-16. Deep Research is a new agentic AI capability integrated within ChatGPT that autonomously conducts multi-step web research and synthesizes comprehensive reports.

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.

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.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4): 1–14, 2018.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pp. 745–750, 2007.

Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, Olivier Pietquin, and Laura Toni. A survey of temporal credit assignment in deep reinforcement learning. *arXiv preprint arXiv:2312.01072*, 2023.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning. *Advances in Neural Information Processing Systems*, 37:52516–52544, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

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

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.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023.

Yunpeng Qing, Shunyu Liu, Jie Song, Huiqiong Wang, and Mingli Song. A survey on explainable reinforcement learning: Concepts, algorithms, challenges. *arXiv preprint arXiv:2211.06665*, 2022.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

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.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Ahmad Rashid, Ruotian Wu, Julia Grosse, Agustinus Kristiadi, and Pascal Poupart. A critical look at tokenwise reward-guided text generation. *arXiv preprint arXiv:2406.07780*, 2024.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Zhizhou Ren, Ruihan Guo, Yuan Zhou, and Jian Peng. Learning long-term reward redistribution via randomized return decomposition. *arXiv preprint arXiv:2111.13485*, 2021.

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*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Training language models with natural language feedback. *arXiv preprint arXiv:2204.14146*, 8, 2022.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yunyi Shen, Hao Sun, and Jean-François Ton. Reviving the classics: Active reward modeling in large language model alignment. *arXiv preprint arXiv:2502.04354*, 2025.

Ruizhe Shi, Minhak Song, Runlong Zhou, Zihan Zhang, Maryam Fazel, and Simon S Du. Understanding the performance gap in preference learning: A dichotomy of rlhf and dpo. *arXiv preprint arXiv:2505.19770*, 2025.

Weiyan Shi, Emily Dinan, Kurt Shuster, Jason Weston, and Jing Xu. When life gives you lemons, make cherryade: Converting feedback from bad responses into good labels. *arXiv preprint arXiv:2210.15893*, 2022.

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.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. Distributional preference learning: Understanding and accounting for hidden context in rlhf. *arXiv preprint arXiv:2312.08358*, 2023.

Andries Smit, Paul Duckworth, Nathan Grinsztajn, Thomas D Barrett, and Arnu Pretorius. Should we be going mad? a look at multi-agent debate strategies for llms. *arXiv preprint arXiv:2311.17371*, 2023.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. A roadmap to pluralistic alignment. *arXiv preprint arXiv:2402.05070*, 2024.

A Springall. Response surface fitting using a generalization of the bradley-terry paired comparison model. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 22(1):59–68, 1973.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Hao Sun and Mihaela van der Schaar. Inverse-rlignment: Inverse reinforcement learning from demonstrations for llm alignment. *arXiv preprint arXiv:2405.15624*, 2024.

Hao Sun, Alihan Hüyük, and Mihaela van der Schaar. Query-dependent prompt evaluation and optimization with offline inverse rl. In *The Twelfth International Conference on Learning Representations*, 2023a.

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

Hao Sun, Alex James Chan, Nabeel Seedat, Alihan Hüyük, and Mihaela van der Schaar. When is off-policy evaluation (reward modeling) useful in contextual bandits? a data-centric perspective. *Journal of Data-centric Machine Learning Research*, 2024a.

Hao Sun, Yunyi Shen, and Jean-Francois Ton. Rethinking bradley-terry models in preference-based reward modeling: Foundations, theory, and alternatives. *arXiv preprint arXiv:2411.04991*, 2024b.

Hao Sun, Yunyi Shen, Jean-Francois Ton, and Mihaela van der Schaar. Reusing embeddings: Reproducible reward model research in large language model alignment without gpus. *arXiv preprint arXiv:2502.04357*, 2025a.

Hao Sun, Yunyi Shen, and Mihaela van der Schaar. Openreview should be protected and leveraged as a community asset for research in the era of large language models. *arXiv preprint arXiv:2505.21537*, 2025b.

Haoyuan Sun, Yuxin Zheng, Yifei Zhao, Yongzhe Chang, and Xueqian Wang. Generalizing offline alignment theoretical paradigm with diverse divergence constraints. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*, 2024c.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Niket Tandon, Aman Madaan, Peter Clark, and Yiming Yang. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. *arXiv preprint arXiv:2112.09737*, 2021.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

DeepSeek Team. Grpo: Generalized reinforcement preference optimization. *arXiv preprint arXiv:2405.00000*, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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.

Chenglong Wang, Yang Gan, Yifu Huo, Yongyu Mu, Qiaozhi He, Murun Yang, Bei Li, Tong Xiao, Chunliang Zhang, Tongran Liu, et al. Gram: A generative foundation reward model for reward generalization. *arXiv preprint arXiv:2506.14175*, 2025a.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023a.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023b.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023c.

Qing Wang, Jiechao Xiong, Lei Han, Han Liu, Tong Zhang, et al. Exponentially weighted imitation learning for batched historical data. *Advances in Neural Information Processing Systems*, 31, 2018.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025b.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022b.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Minghao Wu and Alham Fikri Aji. Style over substance: Evaluation biases for large language models. *arXiv preprint arXiv:2307.03025*, 2023.

Teng Xiao, Mingxiao Li, Yige Yuan, Huaisheng Zhu, Chao Cui, and Vasant G Honavar. How to leverage demonstration data in alignment for large language model? a self-imitation learning perspective. *arXiv preprint arXiv:2410.10093*, 2024.

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.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*, 2025.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*, 2023.

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.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for llms. *arXiv preprint arXiv:2406.10216*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

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

Lantao Yu, Tianhe Yu, Jiaming Song, Willie Neiswanger, and Stefano Ermon. Offline imitation learning with suboptimal demonstrations via relaxed distribution matching. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11016–11024, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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.

Tom Zahavy, Vivek Veeriah, Shaobo Hou, Kevin Waugh, Matthew Lai, Edouard Leurent, Nenad Tomasev, Lisa Schut, Demis Hassabis, and Satinder Singh. Diversifying ai: Towards creative chess with alphazero. *arXiv preprint arXiv:2308.09175*, 2023.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets LLM finetuning: The effect of data, model and finetuning method. In *The Twelfth International Conference on Learning Representations*, 2024a. URL `https://openreview.net/forum?id=5HCnKDeTws`.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024b.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*, 2024c.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=u1cQYxRI1H`.

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*, 2024d.

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.

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.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*, 2024.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023a.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022a.

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

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The eleventh international conference on learning representations*, 2022b.

Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyu Wang, Yusuf Aytar, Misha Denil, Nando de Freitas, and Scott Reed. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885*, 2020.