RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs

SHREYAS CHAUDHARI, University of Massachusetts Amherst, Amherst, United States
PRANJAL AGGARWAL, Carnegie Mellon University, Pittsburgh, United States
VISHVAK MURAHARI, Princeton University, Princeton, United States
TANMAY RAJPUROHIT, Independent Researcher, Seattle, United States
ASHWIN KALYAN, Independent Researcher, Seattle, United States
KARTHIK NARASIMHAN, Princeton University, Princeton, United States
AMEET DESHPANDE, Princeton University, Princeton, United States
BRUNO CASTRO DA SILVA, University of Massachusetts Amherst, Amherst, United States

A significant challenge in training large language models (LLMs) as effective assistants is aligning them with human preferences. Reinforcement learning from human feedback (RLHF) has emerged as a promising solution. However, our understanding of RLHF is often limited to initial design choices. This article analyzes RLHF through reinforcement learning principles, focusing on the reward model. It examines modeling choices and function approximation caveats, highlighting assumptions about reward expressivity and revealing limitations like incorrect generalization, model misspecification, and sparse feedback. A categorical review of current literature provides insights for researchers to understand the challenges of RLHF and build upon existing methods.

CCS Concepts: • **Computing methodologies** → **Natural language generation**; **Reinforcement learning**; *Learning from critiques;*

Additional Key Words and Phrases: Preference learning, function approximation, survey and taxonomy

ACM Reference Format:
Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. 2025. RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs. *ACM Comput. Surv.* 58, 2, Article 53 (September 2025), 38 pages. https://doi.org/10.1145/3743127

Shreyas Chaudhari and Pranjal Aggarwal contributed equally to this research.
Authors' Contact Information: Shreyas Chaudhari, University of Massachusetts Amherst, Amherst, Massachusetts, United States; e-mail: shreyaschaudhari@gmail.com; Pranjal Aggarwal, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States; e-mail: pranjala@cs.cmu.edu; Vishvak Murahari, Princeton University, Princeton, New Jersey, United States; e-mail: murahari@princeton.edu; Tanmay Rajpurohit, Independent Researcher, Seattle, Washington, United States; e-mail: tanmay.rajpurohit@gmail.com; Ashwin Kalyan, Independent Researcher, Seattle, Washington, United States; e-mail: asaavashwin@gmail.com; Karthik Narasimhan, Princeton University, Princeton, New Jersey, United States; e-mail: karthikn@cs.princeton.edu; Ameet Deshpande, Princeton University, Princeton, New Jersey, United States; e-mail: asd@princeton.edu; Bruno Castro da Silva, University of Massachusetts Amherst, Amherst, Massachusetts, United States; e-mail: bsilva@cs.umass.edu.

ACM 0360-0300/2025/09-ART53
https://doi.org/10.1145/3743127

1 Introduction

Large Language Models (**LLMs**) demonstrate remarkable capabilities that extend beyond basic language tasks, leading to their widespread adoption across various industries. The remarkable utility of these models holds the potential to transform established workflows in critical sectors such as technology, healthcare, finance, and education [173, 195, 203]. As they become integral to these domains, it's crucial to ensure that the behavior of LLMs is predictable, safe, and trustworthy—meeting the expectations set for a human performing the same tasks. This challenge of making LLMs exhibit human-like qualities, known as *alignment* with human objectives, is central to making these models suitable for diverse tasks. An effective method for addressing this challenge is **reinforcement learning from human feedback** (**RLHF**).

RLHF first gained popularity due to its ability to solve **reinforcement learning** (**RL**) problems like simulated robotic locomotion and playing Atari games [30] without access to a reward function, by simply leveraging human feedback about preferences on demonstrated behaviors. It has since been adopted for fine-tuning LLMs using human feedback. This leads to a natural inquiry: How can a method designed to master games be effectively used to align LLMs with human objectives? The method has proven to be immensely successful [132], but not without well-documented limitations [24]. A comprehensive understanding of *why* it achieves its success remains largely elusive. Consequently, research efforts on the topic are stuck in a local minima, with variants focused on augmenting the components of the method—including the training algorithm [153], **reward model** (**RM**) [197], and even RL-free approaches [149]. However, some fundamental limitations of the approach remain obscured due to the overarching goal of recent work to refine the initial design choices.

In this work, we develop a comprehensive understanding of RLHF by analyzing the core components of the method. We begin the study by motivating the necessity for RLHF by highlighting the problem of objective mismatch in pre-trained **language models** (**LMs**) (Section 2). To formulate foundational questions about the framework, we adopt a Bayesian perspective of RLHF. It serves to highlight the significance of the reward function in particular (Section 4). The reward function forms the central cog of the RLHF procedure, and the design choices used to model it form a major focus of our study.

The current formulation of RLHF relies on a set of assumptions to model the reward function (Sections 4.1, 4.2). Following the delineation of these assumptions, an analysis of the RM independent of specific modeling choices follows. The analysis, in a principled manner, provides an understanding of issues such as:

(1) The impractical requirement for extensive amounts of feedback data for training accurate RMs.
(2) The combination of very limited feedback data and the use of function approximation results in misgeneralization, wherein inaccurate reward values are assigned to inputs not seen during training.

These imperfections of the RM, along with challenges such as reward sparsity and RM misspecification, are highlighted in the article (Section 5.1). The course of the analysis leads to the formalization of concepts such as an *oracular reward* that serve as the theoretical golden standard for future efforts (Section 4.1). An overview of the RLHF procedure along with the various challenges studied in this work is provided in Figure 1.

The discussion is followed by an extensive survey of an expanding body of literature related to the topic. The survey is organized into sections that outline the framework of RLHF. Starting with a high-level overview of LLMs, the survey systematically covers various aspects: different types of human (and non-human) feedback (Section 6.3), training methods in RLHF (Section 6.6),

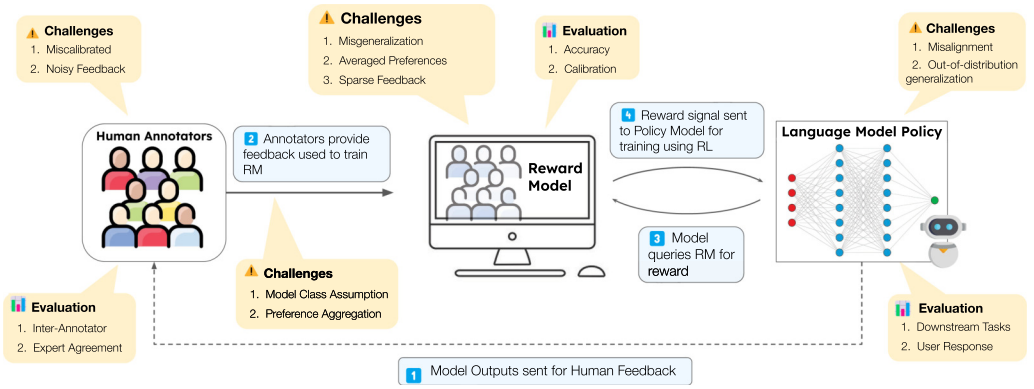

Fig. 1. Overview of the RLHF procedure, illustrating the challenges encountered at each step. The article conducts a detailed examination of these challenges, providing valuable insights into each stage of the procedure.

and alternative approaches that do not rely on RL or RMs (Section 6.10). This structure aims to provide a comprehensive overview of the extensive landscape of works that have contributed to the remarkable success of RLHF.

2 Motivation: Eliminating Objective Mismatch in Pre-Trained Language Models

Large **pre-trained language models** (**PLMs**) are massive neural networks that are trained on a huge corpus of texts using a self-supervised learning objective. Originally utilized for representation learning [39, 108] with encoder-only models, recent research, particularly influenced by Brown et al. [23], has shifted its focus toward training PLMs to directly generate answers for textual problems. State-of-the-art PLMs typically employ an auto-regressive transformer architecture [187] and are trained with a causal language modeling objective. These models implicitly capture a conditional probability distribution π_θ, reflecting the likelihood of sampling the next token after observing a sequence of previous tokens. The probability of a text sequence $x := (x_1, \ldots, x_T)$, under this model is denoted as $\Pr(x; \pi_\theta) = \prod_{t=1}^{T-1} \pi_\theta(x_{t+1} \mid x_t, \ldots, x_1)$. The model is trained to estimate the pre-training data generating probability distribution over text sequences by minimizing the (forward) KL divergence between the model's data-generating distribution and the pre-training data distribution, denoted by $P_{\text{pre-train}}(\cdot)$.

$$\min_\theta D_{\text{KL}}(P_{\text{pre-train}}(x) \,||\, \Pr(x; \pi_\theta)) = \min_\theta \mathbb{E}_{x \sim P_{\text{pre-train}}}[\log P_{\text{pre-train}}(x)] - \mathbb{E}_{x \sim P_{\text{pre-train}}}[\log \Pr(x; \pi_\theta)]. \quad (1)$$

The first term, representing the entropy of $P_{\text{pre-train}}$, is independent of θ and can be disregarded during optimization. Consequently, the objective simplifies to the following cross-entropy minimization form: $\min_\theta -\mathbb{E}_{x \sim P_{\text{pre-train}}}[\log \Pr(x; \pi_\theta)]$. The expectation is approximated using samples from an unsupervised pretraining text corpus $\mathcal{D}$, which comprises text sequences sampled from $P_{\text{pre-train}}$. This leads us to the following objective:

$$\min_\theta -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \sum_{t=1}^{T-1} \log \pi_\theta(x_{t+1} \mid x_t, \ldots, x_1). \quad (2)$$

The remarkable property about PLMs lies in the contrast between the simplicity of the training recipe and the remarkable results that they deliver [23]. Simply capturing language statistics along with scaling up the number of trainable parameters, endows PLMs with robust semantic representations, vast commonsense knowledge, and strong pattern-following capabilities. However,

for adopting PLMs to assist humans with tasks that require an understanding of human intentions and the ability to follow instructions, the simple training recipe of PLMs is insufficient. These models demonstrate a shallow understanding of human intentions, often generating undesirable outputs, including incorrect facts or conveying biased and toxic opinions.

Fundamentally, PLMs suffer from an *objective mismatch* problem: the training-time objective of capturing language statistics does *not* necessarily align with the deployment-time objective of fulfilling a human user's specific goals. Eliminating this mismatch at first glance seems feasible: just train PLMs to optimize for the user objective. Unfortunately, for many tasks, it is impossible to express the user objective as an optimization target. For example, when a user's objective pertains to eliciting humorous responses, establishing specific criteria for objectively evaluating the humor in a generated response becomes an inherently challenging task.

There are currently two primary ways to deal with the problem: the behaviorist approach and the cognition-driven approach. The behaviorist approach, implemented by **supervised fine-tuning** (**SFT**), aims to replicate observable behaviors that humans perceive as desirable without explicit consideration of the underlying user objective. For instance, if a user desires good summaries of articles, this approach trains a model to imitate examples of good summaries without explicitly defining the criteria for a good summary. In contrast, the cognition-driven approach, implemented by RLHF, aims to uncover the underlying user objective that governs the observed behaviors. It then updates the model by optimizing the uncovered objective. This approach relies on certain assumptions—which in the case of RLHF are: (i) the user objective can bear the form of a reward function, which can assign a numerical score to behaviors of the model, and (ii) this function can be approximated by a machine learning model (e.g., a neural network). RLHF estimates this reward function and updates the PLM via RL to optimize for rewards. Regardless of the approach, the process of addressing the objective mismatch problem is commonly referred to as the *fine-tuning* or *alignment* process. Presently, state-of-the-art LMs typically initiate this process with the behaviorist approach, followed by the cognition-driven approach.

Bayesian Interpretation of RLHF: RLHF relies on observing *human feedback* to deduce the (latent) user reward function. Human feedback is provided on the outputs from an LM. RLHF assumes that there exists an underlying human reward function that governs the feedback they provide in a particular manner, i.e., there exists some mapping from reward to actions of a human. Suppose the reward function is being inferred by a model R_ϕ parameterized by ϕ. Adopting a Bayesian inference perspective [83], the parameters ϕ can be viewed as a hypothesis with the dataset of human feedback $\mathcal{D}_{\text{HF}}$ as the evidence for this hypothesis. Given a prior distribution over the hypothesis $\Pr(\phi)$, we can apply Bayes' rule to derive the posterior distribution over the hypotheses after observing the evidence as

$$\Pr(\phi \mid \mathcal{D}_{\text{HF}}) \propto \Pr(\mathcal{D}_{\text{HF}} \mid R_\phi) \Pr(\phi) \tag{3}$$

Reward modeling in RLHF can be seen as computing the **maximum a posteriori** (**MAP**) estimate of the parameters of an RM,

$$\phi_{\text{MAP}} = \arg\max_{\phi} \Pr(\phi \mid \mathcal{D}_{\text{HF}}) = \arg\max_{\phi} \underbrace{\Pr(\mathcal{D}_{\text{HF}} \mid R_\phi)}_{\text{(a)}} \underbrace{\Pr(\phi)}_{\text{(b)}} \tag{4}$$

The first term (a) is the log-likelihood of the feedback dataset, specifying how a human's internal objective (reward function) governs their feedback. The second term (b) represents constraints on the hypothesis space, which is enforced through explicit and implicit regularization techniques in neural-network training.

The presented framework raises two major questions:

(1) What is the form of the likelihood function $\Pr(\mathcal{D}_{\text{HF}} \mid R_\phi)$? In other words, how do we mathematically model the influence of a human's latent objective on their observable feedback?
(2) What is the RL algorithm used for optimizing the RM? In other words, how do we ensure the model acts consistently with its objective?

A set of answers to these questions forms the basis for an RLHF algorithm. The RLHF methodology, popularized by Christiano et al. [30], employs pairwise ranking feedback and uses the Bradley–Terry model [21] as the likelihood function. **Proximal Policy Optimization** (**PPO**) [165] is selected as the RL algorithm.

Before we move into the analysis of this method, we urge the readers to take a moment to reflect on the choices and assumptions we have made so far to derive the general recipe of RLHF. Are there alternative choices? Can the assumptions be relaxed or improved? Thinking critically about these foundational decisions is the key to understanding the strengths and weaknesses of RLHF algorithms and innovating them. For example, the recently proposed **direct preference optimization** (**DPO**) approach [149] replaces RL with a reformulation of the objective. Next, we formalize the problem setup of text generation as an agent interacting with a sequential decision process, laying the foundation for the analysis of RLHF. We refer the reader to Section 6.6 for a detailed outline of the RLHF procedure, and Figure 5 for a summarized overview.

3 Formulation: Text Generation as Sequential Decision-Making

In this section, we formulate the text generation procedure from an LM as a sequential decision-making process. This formulation is essential for constructing RL algorithms.

Markov decision process. A common framework for modeling sequential decision-making processes is ***Markov Decision Process*** (**MDP**) [111]. An MDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, p, R, \rho)$ where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\rho : \mathcal{S} \to \Delta(\mathcal{S})$ is the initial state distribution. Each sequential timestep of the process is denoted by t, and s_t, a_t, r_t denote the values of the state, action, and reward at timestep t. A discounting factor $\gamma \in (0, 1]$ is defined for discounting rewards over time, particularly useful for modeling an MDP with an infinite number of timesteps (i.e., an infinite-horizon MDP). However, the outputs of LMs are truncated after a finite number of steps. We use T to denote the maximum timestep.

An agent acts in an MDP using a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. The agent starts in state $s_1 \sim \rho(\cdot)$. At timestep t, it chooses an action $a_t \sim \pi(\cdot \mid s_t)$, executes the action, transitions to a new state $s_{t+1} \sim p(\cdot \mid s_t, a_t)$, and receives a reward $r_t = R(s_t, a_t)$. The term "Markov" in MDP refers to the Markov property, in that the distribution over the next state s_{t+1} depends on only the current state s_t and action a_t.

Language models as agents in MDP. For simplicity, we consider text generation tasks that include only one turn of interaction between the user and the model. We make a distinction between the text that a user inputs into the model, denoted by c and referred to as the *context* or the *prompt*, and the text that the model generates by itself to the context, denoted by o and referred to as the *output* or simply the *generated text*.

Let V be the set of all tokens that the model can generate (the vocabulary), C the set of all possible contexts, and O the set of all possible outputs. Given a context $c \in C$ as input, the model generates an output $o \in O$ token by token. Specifically, let o_t be the tth token in generated output o, then the model parameterized by θ first outputs token $o_1 \sim \pi_\theta(\cdot \mid c)$, and then conditioned on the concatenation of o_1 and c it generates $o_2 \sim \pi_\theta(\cdot \mid [c, o_1])$, and so on.

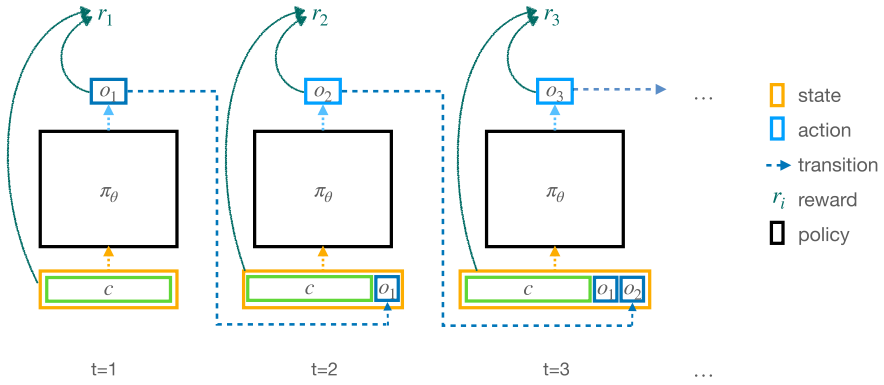

Fig. 2. Text generation from LLMs modeled as a Markov decision process. The generation process is auto-regressive, utilizing the token output (action) from the previous timestep and the context (state) as input to produce the next token through the LM (policy). Given a context c, the LM produces the token o_1 at the first timestep. A concatenation of the two $[c, o_1]$ forms the input to the policy at the next timestep (Table 1). A reward function scores the generated output for a given context.

Table 1. Mapping from Text Generation to MDP

MDP element	Description	Text generation equivalence
s_1	Initial state	c
s_t	State at time step t	$[c, o_{1:t-1}] = [s_1, a_{1:t-1}] = [s_{t-1}, a_{t-1}]$
a_t	Action taken at time step t	o_t
$\pi(a_t \mid s_t)$	Policy	$\pi_\theta(o_t \mid [c, o_{1:t-1}])$
r_t	Reward at time step t	$R_\phi([c, o_{1:t-1}])$
$\rho(s_1)$	Initial state distribution	$\Pr(c)$
$p(s_{t+1} \mid s_t, a_t)$	Transition function	$\delta([s_t, a_t])$

We can see that this generation process resembles an agent traversing in an MDP (Figure 2). The model acts according to a policy π_θ. The start-state distribution ρ is the distribution over user-provided contexts. The action space is the vocabulary V. The action a_t is the generated token o_t. The state s_t is the concatenation of the context c and all the tokens the model has generated up to timestep $t-1$. The transition function $p(\cdot \mid s_t, a_t) = \delta([s_t, a_t])$ is a delta distribution, i.e., the next state is deterministic given the current state and action. Reward r_t given at timestep t is computed by the RM as $R_\phi(s_t, a_t)$ which is either a human or a function learned from human feedback.

The text generation MDP has several special properties:

(1) The action space is extremely large. For example, the LLaMa model [184, 185] employs a vocabulary of size 32K. Having a gigantic action space blows up the search space for RL algorithms.
(2) The structure of the state space is complex, as a state is essentially a text sequence. Pre-training on large amounts of texts is necessary to learn an initially good representation of this space.
(3) The initial state distribution has an enormous support. All conceivable contexts lie in the support, thus strongly testing the ability of the policy to generalize to out-of-distribution states.
(4) The reward function used for training *can* differ from the evaluation reward function. This is because the humans providing rewards during evaluation may be different from the humans involved in training the RM. Analogous to transfer learning in RL, the agent must then adapt to the new reward function.

(5) The transition function is deterministic. Algorithmic tools designed for deterministic MDPs can be applied.

Thus, solving a text generation MDP requires specialized treatment that takes advantage of its properties and overcomes its inherent challenges. RL [18, 182] provides solutions for optimally solving an MDP, i.e., learning a policy that maximizes the accumulated reward. Consequently, RLHF updates the LM to generate more *rewarding* outputs. Naturally, the reward function plays a critical role in the process of fine-tuning model outputs, determining practical and fundamental limits [24] of the efficacy of RLHF.

4 The Role of Reward

The goal of reward learning in RLHF is to convert human feedback into an optimizable reward function. The reward serves a **dual purpose**: it encodes the *task information* (for example, identical input-output pairs would receive distinct rewards depending on whether the task involved summarization or text expansion) [1] as well as *preferences* over those outputs (a condescending summary is rewarded less than a neutral summary). The reward thus encodes relevant information for measuring (Section 4.3) as well as inducing alignment with human objectives. By setting the reward function of the sequential decision process to the one estimated from human feedback R_ϕ, RL algorithms can be used to learn an LM policy that maximizes the cumulative reward, resulting in an *aligned* LM.

4.1 Oracular Reward and the Role of Human Feedback

An implicit assumption made in RLHF is that a human's feedback behavior is governed by and can be represented as an *oracular* reward function $R^\star : C \times O \to \mathbb{R}$. We assume that this function is deterministic in line with the current methodology. The function takes as input a context c and an output o, and outputs a scalar number reflecting the preference on o as a continuation of c. Because the $[c, o]$ is essentially a state in the MDP formulation, the reward function is essentially defined over states of the MDP. *The LM that maximizes the oracular reward accurately reflects the goals and preferences inherent in the human feedback*, and maximization of this reward consequently aligns the model with the human preferences. The oracular reward may not be accessible or learnable, but under the reward hypothesis [172, 180], the mere existence of such a reward may be assumed—though this may be challenged [81]. The oracular reward forms the golden standard for training as well as evaluating any LM.

In general, humans can give a variety of feedback. RLHF operates with feedback that discloses information about the oracular reward function. Most methods focus on two types of feedback: *point-wise numerical feedback* (or rating), and *pairwise ranking feedback* (or preferences). Providing ratings is the most straightforward way to communicate the reward function. Given a pair (c, o), the rating is a scalar $r = R^\star(c, o)$. While ratings can be fed directly into an RL algorithm, learning an RM takes advantage of the generalizability of the RM on unseen outputs and contexts.

Preference feedback compares two outputs generated for the same context. Given two outputs o and o' generated for context c, a human denoted a preference $o \succ o'$ if the first input is preferred and $o' \succ o$ otherwise. Preferences in their raw form are not compatible learning signals for RL algorithms. Hence, an RM must be learned for this type of feedback. To do so, an assumption must be made about the relationship between preferences and $R^\star$. We will discuss this in more detail in the next section. A discussion about the various methodologies used for encoding preferences can be found in Section 6.5. An alternative approach for ranking outputs on the basis of preferences is provided by the *learning-to-rank* paradigm [107].

[1] Unless the task is specified in the input prompt itself, in which case the inputs differ.

Using preference feedback offers several advantages compared to using ratings. Firstly, we get more training data for the RM. In practice, people collect a ranking of N outputs and create preference pairs [136]. Collecting N ratings for N outputs provides we get N training points. Ranking N outputs provided $N(N-1)/2$ pairwise comparisons. Second, preferences require assigning a only relative order rather than an absolute precise score to an output; the latter task could take significantly more cognitive effort and is more prone to inconsistency. Finally, a preference is presumably easier to provide because it offers a "baseline" for comparison (the worse output). In contrast, when giving a rating, a human can rely on only the evaluation guidelines.

A note on stochastic rewards: The reward function is considered to be a deterministic mapping from text to a scalar value. This amounts to averaging the preferences of all humans that provided human feedback. Moreover, it assumes that a human must always rate an input-output pair with the same score, discounting the inherent variability of human preferences. There are numerous scenarios—like personalization, in-context adaptation to ongoing dialogue, and diverse output generation—where a deterministic mapping is limiting. The rewards are more appropriately modeled as being stochastic, wherein each input-output pair is scored by a distribution over scalar rewards, say $r \sim R_{\text{human}}(\cdot \mid c, o)$. This modeling accounts for the two sources of uncertainty: (i) uncertainty over the specific human from a group of humans who provide feedback, and (ii) variability in a human's preferences due to changes in unobserved factors [128]. Work in RL aims to address this by learning Bayesian preferences, primarily for uncertainty quantification and safety analysis [22, 152], and can be adapted to model a distribution of preferences over text. Recent efforts along these lines [15] have proven to be effective. We focus on deterministic rewards for the analysis that follows.

4.2 Reward Modeling

Learning an RM serves two purposes: (i) to convert RLHF into a canonical RL problem, and (ii) to reduce the cost of online feedback-collection. RL algorithms define their objective in terms of a reward function. To apply these algorithms, we need to infer a reward function from a feedback dataset, collecting which is notoriously expensive. Currently, LLMs require thousands to millions of feedback data points. To gather that amount, many human evaluators need to be recruited to work in parallel. To ensure the assumptions regarding the oracular reward function hold, the evaluators must be trained to agree with one another on the evaluation criteria. This process is continual: multiple rounds of feedback collections need to be conducted to iteratively improve the model. The premise of approaches that learn an RM is that the generalization error of the RM is expected to decrease faster than that of the policy as a function of the number of labeled data points, arising from the notion that supervised learning is often considered a simpler problem than generative modeling.

Following the previous section, we denote the RM by $R_\phi(c, o)$ and the feedback dataset by $\mathcal{D}_{\text{HF}}$. Our goal is to decide a likelihood function $\Pr(\mathcal{D}_{\text{HF}} \mid \phi)$ and find ϕ that maximizes this function. With rating feedback, the reward-modeling problem can be formulated as a prediction problem with continuous output. A common objective for this type of problem is the minimization of the **mean squared error** (**MSE**):

$$\min_{\phi} \sum_{(c,o,r)\in\mathcal{D}_{\text{HF}}} (R_\phi(c, o) - r)^2 \tag{5}$$

To incorporate preference feedback, we need to choose the form of the likelihood function denoting each preference, i.e., $\Pr((o \succ o', c) \mid \phi)$. The RLHF method of Ouyang et al. [136] employs the Bradley–Terry model to represent the likelihood of a data point:

$$\Pr((o \succ o', c) \mid \phi) = \sigma[R_\phi(c, o) - R_\phi(c, o')] \tag{6}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. The reward learning objective is the maximization of the log-likelihood of the dataset $\mathcal{D}_{\text{HF}}$, i.e., $\max_\phi \sum_{(c,o,o')\in\mathcal{D}_{\text{HF}}} \log \Pr((o \succ o', c) \mid \phi) = \max_\phi \sum_{(c,o,o')\in\mathcal{D}_{\text{HF}}} \log \sigma[R_\phi(c,o) - R_\phi(c,o')]$. In Section 5, we further generalize the form of feedback and the likelihood function to conduct an analysis independent of the specifics of particular design choices.

4.3 Measuring Alignment

Evaluation of natural language tasks is a difficult problem, and the study of evaluation metrics is an active area of research. Of particular importance, and difficulty, is to measure the *alignment* of an LM to a human's objectives, which in practice is evaluated along the axes of helpfulness, harmlessness, and honesty. The oracular reward that governs a human's preferences serves as a yardstick for measuring the degree of alignment. The task of alignment is then reformulated as encoding the preferences demonstrated by a human into a reward function, and updating the parameters of the LM to produce output that maximizes this reward.

A reward provides an analytical metric to measure the overall performance of an LM π, where the performance captures the degree of alignment with human preferences along with the degree of satisfaction of the task itself [126]. The performance of a model π, for distribution over contexts $d_C(\cdot)$, can be measured by averaging the rewards for the outputs generated by π given the contexts. Let the performance be denoted by $J(\pi)$:

$$J(\pi) := \sum_c \sum_o d_C(c)\pi(o \mid c)R^\star(c,o) = \mathbb{E}_{c\sim d_C(\cdot)}\left[\mathbb{E}_{O\sim\pi(\cdot|c)}\left[R^\star(c,O)|C=c\right]\right] \tag{7}$$

The context distribution $d_C(\cdot)$ can be the distribution of contexts in the training data, test data, or a held-out validation dataset, depending on the data on which the performance of the model is being evaluated. The sequential nature of the output generation equivalently allows us to express $J(\pi)$ as

$$J(\pi) := \mathbb{E}_\pi\left[\sum_t R^\star(s_t,a_t)\right] = \mathbb{E}_\pi\left[\sum_t R^\star(c,o_{1:t-1},o_t)\right] = \sum_t \mathbb{E}_\pi\left[R^\star(c,o_{1:t-1},o_t)\right] \tag{8}$$

In practice, most current RMs only provide a reward after the complete output has been generated and Equation (8) reduces to Equation (7). The definition of $J(\pi)$ uses the *oracular* reward that is not accessible in practice. An estimate of the performance can obtained from the estimated reward R_ϕ, by plugging it into Equation (7):

$$\widehat{J}(\pi) := \mathbb{E}_{c\sim d_C(\cdot)}\left[\mathbb{E}_{O\sim\pi(\cdot|c)}\left[R_\phi(c,O)|C=c\right]\right] \tag{9}$$

The PLM is denoted by $\pi_{\text{pre}} : C \to \Delta(O)$ and the model updated using RLHF by $\pi_{\text{rlhf}} : C \to \Delta(O)$. The goal of RLHF is to update the parameters of π_{rlhf} such that $J(\pi_{\text{rlhf}}) \geq J(\pi_{\text{pre}})$, i.e., as evaluated using the oracular reward. In practice, it is only possible to verify that $\widehat{J}(\pi_{\text{rlhf}}) \geq \widehat{J}(\pi_{\text{pre}})$, which may be non-informative when the estimated RM R_ϕ has inaccuracies for the context-output pairs being evaluated.

5 Inferring the Reward from Human Feedback

In the following sections, we study the properties of the reward estimated from human feedback. As reviewed in Section 6.5, various procedures exist for encoding human feedback into an RM. Both the form of human feedback and the encoding mechanism continue to be studied further, with the procedures continually evolving and improving. Currently, the most common form of human feedback is pair-wise preference feedback that is encoded into a reward according to the Bradley–Terry model (Section 4.2). To perform an analysis agnostic to specifics of a particular reward learning method,

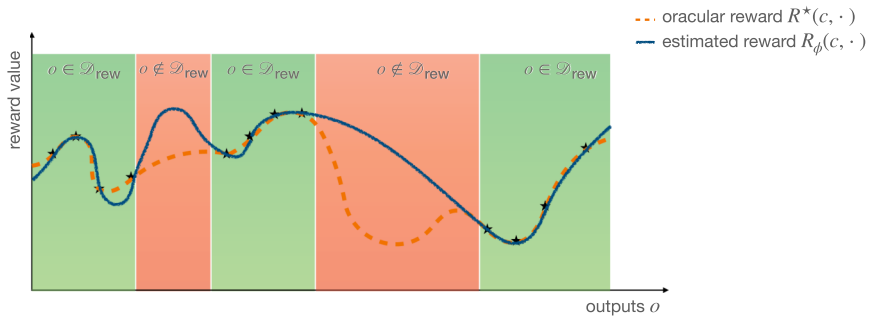

Fig. 3. The RM tends to misgeneralize for inputs not found in its training data, i.e., for $(c, o) \notin \mathcal{D}_{\text{rew}}$. This occurs in two ways: (1) when the context is not sampled by the prompting distribution for generating output and receiving feedback on (represented by κ), and (2) when the support of the output generating distribution—the LM—for a context does not span all possible outputs (represented by ρ). The latter is depicted in this figure.

(1) Let `feedback` denote a general form of sufficiently informative feedback.
(2) Let Ω denote the *model of human behavior*, or the encoding mechanism, that maps the feedback and the text to a reward value.

The generality of this formulation allows the following analysis to cover all existing RLHF-style approaches (for example, RLAIF [11]) as well as future methods for fine-tuning LLMs, that employ an RM.

Let $\mathcal{D} := \{(c, o) : c \in C, o \in O\}$ denote a hypothetical dataset of all possible contexts and outputs that an LM can encounter, i.e., a humongous dataset of size $|C|\times|O| = |V|^T$. This dataset cannot be realized in practice and is invoked to shed light on the practical limitations of the existing methodology. Denote the dataset of collected human feedback by $\mathcal{D}_{\text{HF}} := \{(c, o, \texttt{feedback}) : c \in C_{\text{HF}}, o \in O_{\text{HF}}\}$ where $C_{\text{HF}} \subset C, O_{\text{HF}} \subset O$ are the subsets of context-output pairs (human-)annotated with `feedback`.[2] The reward encoding mechanism that maps context-output pairs along human feedback to rewards (for instance, the Bradley–Terry model) is denoted by $\Omega : (c, o, \texttt{feedback}) \to \mathbb{R}$.[3]

To uncover $R^\star$, it is *assumed* that Ω accurately maps back human feedback to the oracular reward, i.e., for *sufficiently informative* `feedback`, we have

$$\Omega(c, o, \texttt{feedback}) = R^\star(c, o), \ \forall\, c, o \in C_{\text{HF}}, O_{\text{HF}}.$$

Under that assumption, Ω can operate on $\mathcal{D}_{\text{HF}}$ to create a dataset of context-output-reward tuples, $\mathcal{D}_{\text{rew}} = \{(c, o, r) : c \in C_{\text{HF}}, o \in O_{\text{HF}}\}$ where $r = R^\star(c, o)$. With $\mathcal{D}_{\text{rew}}$, learning the RM R_ϕ reduces to a regression problem employing a function approximator. The regression problem is however *underdetermined* [19], and consequently multiple R_ϕ functions can perfectly fit the training data $\mathcal{D}_{\text{rew}}$. However, almost all of these functions fail to accurately represent the oracular reward (Figure 3). Due to the cost of human annotation, practically human feedback can be collected on a very small subset of context and output pairs, i.e., $C_{\text{HF}}, O_{\text{HF}} \subset C, O$. The size of the reward and feedback datasets relative to the hypothetical dataset of all possible inputs and outputs $\mathcal{D}$ can be measured by:

(1) Context coverage: $\kappa := \frac{|C_{\text{HF}}|}{|C|}$
(2) Output coverage: $\rho := \frac{|O_{\text{HF}}|}{|O'|}$, where $O' = \{o : (c, o) \in \mathcal{D}, \ \forall c \in C_{\text{HF}}\}$

[2]The subsets are significantly smaller than C and O. Additionally, the feedback can be of any form: ratings, pair-wise feedback, or language feedback (Section 6.3).
[3]`feedback` is overloaded to capture additional mechanism-specific metadata. For instance, for pair-wise preference, `feedback` can store the preference relation and the (c, o) pair compared against.

Well-understood results in supervised learning suggest that the ratios ρ and κ along with the generalization capabilities of the function approximator [13, 124, 161] determine the generalization performance of the RM for $(c, o) \in C, O$. In practice, the values of ρ and κ are extremely small and consequently the RM often incorrectly generalizes on unseen (out-of-distribution) context-output pairs, assigning incorrect rewards to such inputs. In the following sections, we study practical limitations of estimating RMs.

5.1 Limitations of the Reward Model

The RM parameterized $R_\phi : C \times O \to \mathbb{R}$ is trained on D_{rew} using a sufficiently representative function approximator, to perfectly fit the training data, that is $R_\phi(c, o) = R^\star(c, o),\ \forall\ o, c \in D_{\text{rew}}$. The limitations of the resultant RM may be studied under the following categories:

Misgeneralization: Human feedback is obtained on a very small subset of all possible context-output pairs. This partial coverage over contexts and outputs in D_{rew} combined with the use of function approximators for learning the RM results in the RM $R_\phi(c, o)$ incorrectly generalizing to data points that are *out-of-distribution* relative to D_{rew}. We have assumed a sufficiently representative function approximator that perfectly fits the training data, $\mathbb{E}_{c,o\sim\mathcal{D}_{\text{rew}}}\left[(R^\star(c, o) - R_\phi(c, o))^2\right] = 0$. However, it cannot be ensured that $\mathbb{E}_{c,o\notin\mathcal{D}_{\text{rew}}}\left[(R^\star(c, o) - R_\phi(c, o))^2\right]$ will be zero. It would require a function approximator to perfectly generalize outside the training data distribution, which is not generally attainable, especially when the ratios ρ, κ are minuscule.

The benefits of RL algorithms over other methods for finetuning are contingent on access to an accurate reward function, necessitating accurate out-of-distribution generalization of the RM.

The inaccurate extrapolation out-of-distribution results in an "imperfect" RM that provides feedback on context-output pairs in a manner that when optimized, arbitrarily misaligns with human feedback (and resultant preferences) for those context-output pairs. The output distribution of π_{rlhf} trained on this inaccurate feedback and can only be as good (or bad) as the reward signal provided by the RM. This *inaccurate generalization in the RM* is one of the primary causes of phenomena like "reward hacking" [193] and "hallucinations" [73].

Delayed feedback and Reward Sparsity: RL algorithms benefit from dense rewards as they serve to quickly guide the agent to rewarding states, providing informative feedback to intermediate actions along the trajectory. In RLHF, the feedback from human annotators is obtained for *complete* output generations. Consequently, the RM is trained to provide reward feedback only at the end of the generated output for a given context. This delayed feedback increases the difficulty of optimization with RL algorithms, increasing their sample complexity. Sparse feedback is a constraint inherent to dealing with text and language [175], as it is often unlikely for a human to provide feedback on incomplete sentences. Methods in RL developed to deal with sparse feedback, for instance by stitching together information from partial trajectories [4], cannot be applied directly to textual output due to the semantic constraints of dealing with partial sentences. Denser rewards and corresponding feedback result in faster training, improved sample efficiency [198], and potentially better generalization. Insights from linguistics may be employed to obtain feedback on partial output generations and in turn denser rewards.

Marginalization over preferences: The RM averages over the preferences of all human annotators (and other sources of feedback) to output a deterministic scalar reward for a given context-output pair. The expectation is that averaging over the preferences of multiple sources would be representative of the preferences of an average human persona [38]. The results in rewards that are inconsistent with any single human's preferences. Such preferences are more appropriately denoted by an *distribution* of rewards for a context-output pair. A deterministic model, in addition to discounting the uncertainty and variability of human preferences, cannot model such a distribution, highlighting a case of *model misspecification*.

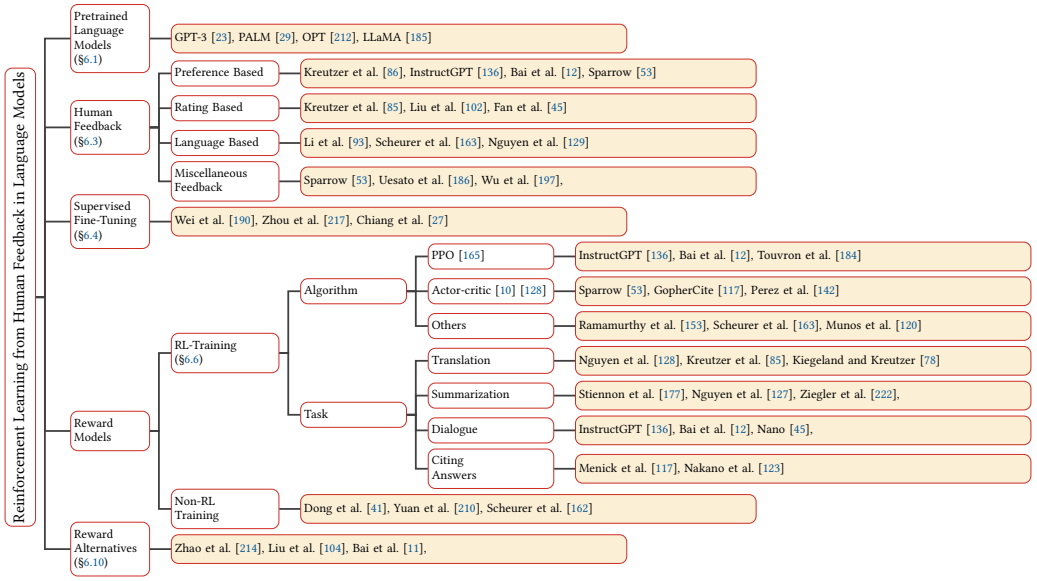

Fig. 4. Categorization of different components in the RLHF and example representative works from literature.

The RM forms the core component of RLHF and dictates the performance of an LM. The aforementioned shortcomings of the RM highlight the need for safety measures that must be employed while using an LM fine-tuned using RLHF.

In the following sections, we present a review of works that lead up to and are being rapidly added to this active area of research. This review provides context for the first half of this work and also serves as a comprehensive introduction for readers interested in getting started and understanding the topic of RLHF for LMs. The review is categorically summarized in Figure 4.

6 Review of Reinforcement Learning from Human Feedback for Language Models

6.1 Language Model Pre-Training: Foundation for Large Language Models

LMs have gained significant attention in recent years due to their impressive abilities to model language and retain textual knowledge. The Transformer architecture, characterized by its use of self-attention mechanisms, has become the standard for LMs [187]. It is employed in a range of models, including BERT, T5, LLaMA, GPT-3, PALM, GLaM [23, 29, 39, 42, 150, 185].

Pre-training has played an important role in the development of LLMs, significantly contributing to their remarkable performance across a myriad of downstream tasks [23, 29, 212]. This process involves training models with an unsupervised training objective on extensive datasets, often comprised of a diverse mix of web content, literary works, scientific documents, and code repositories [148, 201]. The scale of these datasets is critical, with studies highlighting the superior performance of smaller models trained on larger datasets [67, 74, 185]. In addition to scale, the quality of training data, ensured through deduplication and filtering of low-quality content, is a key determinant of model performance [42, 65, 91, 148]. **Masked Language Modeling** (**MLM**) [39] and Causal Language Modeling [147] are the most common objectives used for pretraining, with latter showing notable success in recent LLM series such as GPT, PaLM, OPT [5, 133, 212].

Studies demonstrate that pre-training by itself is responsible for the bulk of the observed capabilities even in downstream tasks [23, 150]. The simple pre-training objective of next, or masked, token prediction imbibes the LMs with a range of capabilities. They are few-task learners, without

the need for fine-tuning. This applies to a variety of tasks from text generation, reasoning, question answering, summarization, and translation to name a few. However, though scaling PLMs exhibit remarkable performance across a variety of tasks, they suffer from several limitations, such as the inability to follow human instructions [136]. This is because PLMs suffer from objective mismatch problems (See Section 2), as they are trained on generic internet data. As a result, PLMs need to learn to mimic the conflicting behavior of billions of humans. Further, the Maximum Likelihood Estimate on the next token prediction for such data doesn't explicitly penalize the model for hallucinating concepts, i.e., generating concepts not encapsulated within its internal representation, and even important and unimportant errors are given equal weightage. Moreover, pretrained models often show unintended behavior such as generating harmful, biased, untruthful, and low-quality content [142].

Supervised-Finetuning. To address the shortcomings faced by PLMs, a straightforward approach is to fine-tune them on a set of high-quality downstream datasets that are indicative of the intended task and behavior. For example, for instruction-following, human annotations can be collected on a set of input prompts, or input instances of existing public datasets can be re-formatted for instruction-following format. The model is then simply fine-tuned on these human demonstrations, often with the same pretraining objective. This increases the likelihood of generating desirable text and makes the model less biased and harmful. Nonetheless, in order to generate high-quality text, it is crucial to note that the task of distinguishing between high and low-quality text is inherently subjective and challenging, with end users being humans. Thus, quality assessment rests on human judgment and varies significantly based on the individual evaluator's perspective [45, 208, 222]. Incorporating human feedback into such a process can be challenging, and collecting high-quality human demonstrations can be expensive and not scalable.

6.2 Reinforcement Learning from Human Feedback (RLHF): Overview and Motivation

The Importance of Human Feedback in Language Models. The alignment of a model with the user's intentions and preferences is critical, and incorporating human feedback in model training is a key step toward achieving this (Section 2). However, the process of obtaining high-quality human feedback, particularly in the form of human demonstrations, can be a resource-intensive process, both in terms of time and cost. A more efficient approach is to collect feedback on the outputs generated by the model and train the LM to incorporate this feedback. However, collecting such a large amount of feedback is also costly and impractical for real-time/online collection during training.

The Role of RLHF in Language Models. RLHF offers a solution to these challenges. In RLHF, human feedback is collected offline and used to train an RM. This RM then acts as a surrogate for human feedback during training, providing reward signals to the LM. RL algorithms form the natural candidates for training a model from scalar evaluative feedback, as provided by the RM. This forms the essence of RLHF [30] as used to train LMs. This approach is more sample-efficient and has shown more promising results compared to SFT alone [136].

Applications of RLHF in Language Models. In early works, RL has been used in training LMs across various domains such as dialogue generation [71, 93, 208], machine translation [46, 85, 128, 175], text generation [95, 169, 218, 221], semantic parsing [90], summarization [177, 194, 222]. More commonly, these methods were trained using non-differentiable automated evaluation metrics such as BLEU, ROUGE [76, 155, 168], or simulated feedback [128]. However, while the combination of RL and human feedback has been extensively studied [30, 82], it is only recently that RLHF with LLMs has achieved significant success in sequence-to-sequence tasks such as Summarization [177, 194, 222], providing reliable answers with citations to queries [53, 123], creating Helpful, Harmless and Honest dialogue agents aligned with broad human values [12, 136].

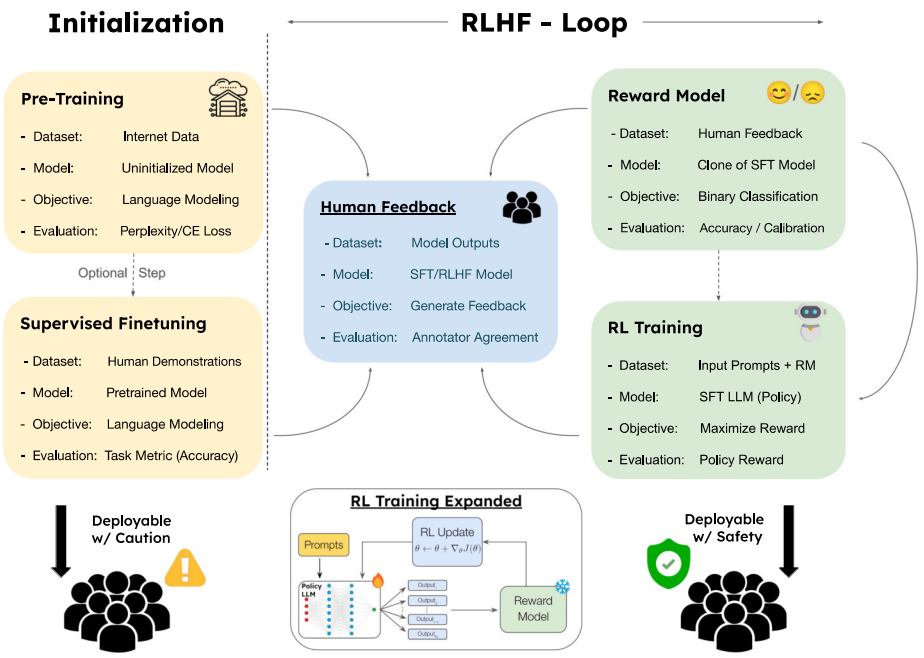

Fig. 5. Workflow of RLHF. A pretraining phase, and optionally SFT on human demonstrations, is followed by all RLHF workflows for training LMs. This is followed by an iterative loop starting with collecting human feedback on model-generated outputs, training an RM, and updating the LM using a suitable RL algorithm.

Formulating Language Modeling as an RL Problem. RL is a learning paradigm for a setting where an agent must make a sequence of decisions while interacting with an environment and obtaining evaluative feedback in the form of rewards. The agent's objective is to maximize the total reward it receives over time. In the context of LMs, the agent is the LM itself, and its actions consist of generating tokens from its vocabulary. The agent's policy, which maps states to actions, is represented by the LM's parameters. The agent receives rewards from the environment, which in this case is a reward function that forms a surrogate from human feedback (Section 4). The agent's objective is to optimize its actions (by updating its policy) to maximize the cumulative reward. A thorough mathematical formulation can be found in Section 3, and has been summarized in Figure 2. While these details are sufficient for further discussion in the article, we refer interested readers to Arulkumaran et al. [7], Sutton and Barto [182] for more details about RL.

The Workflow of RLHF. RLHF, as first popularized by [30] for mastering Atari Games, consists of three crucial stages. An overview of standard RLHF workflow is highlighted in Figure 5. The first stage involves the collection of human feedback on a set of `<input, output>` pairs. These pairs can be sourced from existing datasets or generated by the pre-trained model for a given set of input prompts. The second stage involves learning an RM from the collected human feedback. The RM is trained to output a scalar reward for a given `<input, output>` pair, indicating the favorability of the pair. In essence, the RM is trained to mimic human feedback, such that for a given input, desirable outputs are scored higher than undesirable outputs. The final stage involves the RLHF training of the LM, where the RM provides reward signals on model outputs, usually in the form of scalar reward. The parameters of the LM are then updated based on these reward signals using an appropriate policy-gradient RL algorithm, updating the model to produce more rewarding outputs.

These stages can be performed iteratively, with the intermediately trained model generating more prompts to collect additional human feedback. This feedback is then used to train the RM, and the process is repeated multiple times [12, 117, 177]. In the following sections, we discuss each of these stages in detail. We start with Human Feedback Collection (Section 6.3), followed by training the Initial Policy (Section 6.4), Reward Model Training (Section 6.5), and finally RLHF Training (Section 6.6). Finally, we discuss the properties of RLHF-trained models and their limitations in Section 6.8.

6.3 Human Feedback

In this section, we discuss the nature, objectives, and different types of human feedback, followed by the challenges and strategies associated with collecting high-quality feedback.

6.3.1 Nature and Objectives of Human Feedback. Tasks such as summarization and providing helpful answers are inherently ambiguous and require human judgment to evaluate the quality of the generated text. Automated metrics like BLEU and ROUGE [99] often do not correlate with human judgment [103, 164, 166, 177], making them unreliable for evaluation and training. Thus, acquiring high-quality human feedback to align the model with human behavior becomes crucial. Feedback is typically provided on the outputs generated by the model (or input-output pairs from the dataset), and subsequently, the model is trained to learn from this feedback. However, capturing diverse human preferences is a challenging task. One approach to encapsulate subjective human preferences is to approximate them using "models of human behavior". This concept of human behavior models has roots in diverse fields such as econometrics [114], psychology [131], and inverse RL. A notable example is the Bradley–Terry model [21], a probabilistic model that encodes the preference of one output over another in pairwise competitions. In the context of RLHF, RMs that form surrogates for human preferences serve as such models of human behavior.

The type of feedback collected depends on the intended objective to be displayed by the fine-tuned LM. Askell et al. [8] propose three objectives for an aligned LM: **Helpfulness, Honesty, and Harmlessness** (**HHH**). These objectives can be broadly defined as follows:

- **Helpful:** An LM is considered helpful if it can efficiently complete tasks or answer questions (while being harmless), ask relevant follow-up questions when necessary, and appropriately redirect ill-informed requests. Helpfulness includes context-dependent aspects such as informativeness, coherence, relevance, creativity, and specificity.
- **Honest:** Honesty in an LM implies providing accurate information, expressing appropriate levels of uncertainty, and honestly conveying its capabilities, knowledge, and internal state. LMs are particularly susceptible to hallucination [77, 113], making it essential to penalize such behavior. Unlike helpfulness, honesty is more objectively evaluated.
- **Harmless:** A harmless LM should avoid offensive or biased behavior, refuse to aid in dangerous acts, recognize disguised nefarious attempts, and act with modesty and care when providing advice with potentially sensitive or consequential impacts.

These broad objectives, as mentioned above, encompass specific objectives, which can be considered subcategories. For example, in the case of summarization, the summary should be helpful to the reader and should not contain any false or harmful information. Similarly, the goal of reducing bias in a dialogue agent's responses can be considered a subset of the Harmless objective. At the same time, coherence and creativity in the generated text are aspects of being helpful. These objectives are not mutually exclusive and are context and task-dependent. Even human labelers and researchers have shown disagreements in annotation [86].

6.3.2 Types of Human Feedback. Human Feedback is usually collected on model-generated outputs. Good feedback should incorporate information on where the model output is lacking

and how to improve it. A simple process is to let human labelers provide feedback on a set of model outputs generated from a dataset of prompts or inputs. Alternatively, existing datasets can be repurposed to incorporate implicit feedback, such as rating different user choices [85]. Regardless of the process, human feedback can be collected in various forms, such as binary responses, preference ranking, language feedback, and so on. While the choice of feedback type depends on the downstream task, it is essential to note that the feedback should be collected in a way that is easy for humans (labelers) to provide; there is high agreement among the labelers, and it is also informative. In this section, we classify the feedback into four different categories: rating feedback, ranking feedback, language feedback, and miscellaneous feedback.

Rating Feedback. The simplest form of rating feedback is binary feedback, where the labeler is asked to provide a binary response (yes/no) to a given input [93, 163]. Binary feedback is easy to collect and interpret. Some works have used binary responses to get feedback on multiple questions (such as if the generated text is coherent) [208]. A richer form of feedback is to ask labelers to provide a rating on a scale. The scale can be continuous [54], or be similar to Likert Scale [98] (where user rate using an integer from 1 to k) [71, 85]. A different variant of rating feedback is to provide categorical feedback such as "incorrect", "partially-correct", and "correct" [51]. While rating feedback is easy to specify, often inter-annotator agreement is low because of the subjective nature of the task [86]. Further, the order of examples presented to the annotator may bias the results [207]. Moreover, it is challenging to differentiate between data points with outputs of similar quality since feedback is provided individually to each output without comparison.

Ranking or Preference Feedback. Ranking feedback or Preference-based feedback has been extensively used in the recent development of AI assistants and found to be both convenient to collect and performative. Specifically, the labeler is offered with binary [177] or multiple choice options [222], and asked to select the most appropriate response based on a certain set of instructions (directions). Recently, [219] has shown convergence guarantees for RMs trained using this feedback form. Moreover, given an input prompt, it is common to ask labelers to rank k (> 2) generated responses, which are then repurposed as pairwise comparisons for the RM [136]. However, collecting pairwise feedback might still be difficult for near similar responses and may result in much time spent by the labelers even on single input [163]. Additionally, preference-based feedback provides a very sparse signal, conveying limited information about the reasoning behind the provided feedback. Moreover, it is provided only on the complete text generated by the model (trajectory) and not on specific parts of the text (particular state) [92, 138]. Moreover, preference-based feedback provides no further improvement in terms of inter-annotator agreement when compared to rating feedback [86].

Language Feedback. A more informative way to provide feedback is in free-form language. This provides a dense reward signal, specifying more precisely where the model goes wrong or needs improvement. For example, consider the case where the output generated by the model is "A humorous story about a specific profession involving person A and person B." The previous feedback forms would provide only sparse signals, such as indicating that the output is inappropriate. However, this feedback alone will not help the model identify the cause of inappropriateness, and the single example alone can imply that the text is inappropriate because: "it is wrong to create humor in general," "it is wrong to create humor about specific professions" or "it is wrong to involve individuals in humorous stories" and so on. On the other hand, free-form feedback can provide more precise feedback, such as "It is inappropriate to create humor that targets specific professions." This enables the model to understand the issue from a single example better and generalize to similar cases without learning from more examples.

Language Feedback has been extensively used in various domains such as Dialogue models [61, 93], Summarization [163], Question-Answering [96], Code generation [25]. Recently, [163] has shown that language feedback is more effective than preference-based feedback in the context of summarization systems. Also, as [61] discusses, getting preference-based feedback is plausible for paid labelers but not for real users using real deployed systems. Real users interact with the system through free-form language; hence, getting human feedback in the free-form language is more natural. Although task-dependent, [163] further find that labelers take only 3x times to provide language feedback compared to preference-based feedback, despite providing much granular information. However, incorporating language feedback in the RLHF pipeline is not straightforward, and there has been limited work in this direction.

Miscellaneous Feedback. Apart from providing single feedback, methods have experimented with using a combination of feedback types or altogether different types. For example, [53] uses a combination of rule violation feedback (binary), preference-based feedback, and rating of evidence. References [84, 186] provide segment-level feedback instead of the whole text, and [197] provide feedback at the token level. Moreover, some studies employ indirect methods for collecting feedback. For example, [85] uses human interactions on translated eBay titles to find more preferred translations.

Further, it is also possible to provide computational feedback, for example, from automated metrics [10], forms of synthetic feedback [20, 79], web descriptions [1, 62], LLM generated feeedback [110, 170, 205], which might, in turn, be generated based on certain human requisites or instructions [11, 87, 179]. However, these methods still use little to no human feedback and may have several unexplored limitations such as instability and lack of robustness [3, 57, 171] and are not the focus of this survey. We refer readers to Fernandes et al. [47] for discussion on different type of feedback used in Natural Language Generation.

6.3.3 Collection of High-Quality Human Feedback. Collecting high-quality human feedback is a challenging task that has been the focus of extensive research. The quality of feedback is pivotal; subpar or noisy feedback can significantly hamper the performance of the final trained model. For example, for summarization tasks, [222] discovered that their model predominantly extracted verbatim lines from the document. This was later attributed to low-quality feedback by [177]. Similarly, the size of the feedback is also crucial. For example, despite employing similar methodologies, [97] identified a need for a "greater amount of feedback" for the methods in [186] to be effective, as the intended objective was not even observed in the latter work.

The provision of clear and unambiguous instructions to the labelers is a fundamental requirement [123, 222]. Failure to do so can not only result in low-quality feedback but also introduce systematic bias in the collected feedback and, consequently, the model [139]. Typically, labelers are provided with a comprehensive set of instructions, including guidelines for handling edge cases [12]. [53] even provides a tutorial to the selected few labelers.

Researchers typically screen labelers to ensure they possess the necessary skills to provide feedback. For instance, in the case of translation tasks, bilingual labelers with native proficiency in both languages are preferred [86]. Additionally, a minimum educational qualification is generally preferred. For example, [177] requires labelers to have at least a high-school degree, whereas [123], [53] and [12] require a minimum undergraduate and master's degree respectively. The end goal also influences the selection of labelers. For instance, creating a harmless and helpful chatbot necessitates a diverse group of labelers with varying backgrounds and demographics [12, 136] as otherwise this may result in implicit biases in the model [141]. For instance, currently deployed LMs have been shown to reflect views more aligned with western audiences [43] and may have systematic political biases [160], partly owing to the lack of annotators from diverse demographic groups.

However, despite screening, there may be low agreement among the annotators themselves, or even between researchers and annotators [85]. The labelers are further screened based on two standard criteria (1) inter-annotator agreement, i.e., the agreement between different annotators on the same example, and (2) expert-annotator agreement, i.e., the agreement between annotators and experts [86]. Specifically, the former metric ensures that the labelers are consistent in their feedback, and the latter metric is used to keep only those labelers that have a high agreement with experts. Reference [117] creates a group of super-raters who have a high agreement with experts, and the group is expanded upon iteratively. Even after filtering, some methods ensure a hands-on relationship with labellers [177] and have also created Slack groups for discussing any bugs, issues, or edge cases [12].

6.4 Supervised Fine-Tuning: Limitations and Role

Upon the collection of high-quality feedback, the subsequent step is to assimilate this feedback to train the model. The most direct method to achieve this is to perform SFT of the LM based on the collected feedback. Specifically, human feedback is gathered in the form of expert outputs on input prompts, also referred to as human demonstrations. These human demonstrations can be perceived as positive example outputs to prompts that should be generated by the LM. The model is then fine-tuned on these demonstrations using the same pretraining objective, and this process in RL terminology is often termed behavior cloning [123].

Additionally, when dealing with preference data, the model can be directly fine-tuned on preferred feedback. However, this approach exhibits limitations by not accounting for negative feedback–outputs that the model should avoid generating. This is crucial for training robust models that can handle adversarial situations and identify and rectify errors. To tackle this limitation, alternative methods that incorporate both positive and negative feedback have been developed, as discussed in Section 6.10.

In addition to human demonstrations, existing public instances from NLP datasets can be used as instruction tuning demonstrations [190]. This usually involves creating new instruction-tuning datasets by adding task instructions to existing examples from the dataset [2]. In another field of work, prompts from the initial iterations of GPT-3 [23] served to real customers through Web API were used to fine-tune the model on expert (human) demonstrations provided by contracted labelers [136].

Limitations of Supervised Finetuning. While finetuning on supervised data enhances the model beyond its pretrained version in following instructions and intended tasks, it suffers from numerous limitations. For instance, it does not penalize the model for hallucinating or permit it to learn from neutral or negative feedback. This can lead to harmful and unintended behavior, making it easier to prompt such models to elicit them [50, 142]. Furthermore, behavior cloning is likely to perform poorly in out-of-distribution prompts [145]. These limitations may stem from the fact that during behavior cloning, the model is not allowed to *explore* the vast space of possible actions, i.e., the model is not allowed to generate outputs that are not present in the demonstrations and, in turn, get feedback for them.

SFT as Initial Policy in RLHF models. Despite its caveats, SFT plays a pivotal role in RLHF as it provides a robust initial policy, which allows RLHF methods to work well. From an RL perspective, learning algorithms such as the widely used PPO in training sequence-to-sequence models, struggle to improve from poor initializations, especially when the action space is large, as in the case of text generation. This is because these methods use model-based exploration, which is ineffective when the transition probabilities over many actions are similar [128] i.e., different text outputs have similar probabilities of generation.

Furthermore, as we discuss in Section 6.6, usually a KL penalty is applied to ensure the output text generated by our RL-tuned model is close to the initial model. Thus, during RL training, it is preferable to start with an initial model that already generates decent-quality text. Empirical studies have demonstrated that starting with fine-tuning on high-quality human demonstrations results in significant improvements over starting with PLMs [136, 177]. For instance, InstructGPT collects API customer and labeler written prompts and outputs to fine-tune their model before initiating with the RLHF training [136].

Glaese et al. [53] have also shown that starting with a prompted model (dialogue) instead of fine-tuning on label demonstrations is possible. However, they start with a large model (70B parameters) and do not perform a comparative study starting with fine-tuning on human demonstrations. Thus, it cannot be definitively concluded that starting with a prompted model is equivalent to fine-tuning on human demonstrations. Moreover, prompting has the limitation of using up a major portion of the context length of the model, which, apart from the computational burden, can also be crucial for some tasks because of limited context length. Reference [8] proposes using context distillation by training the model to generate output similar to its prompted counterpart using KL divergence loss. They find similar performance to the prompted model, and the method has been used in their subsequent works [12].

In conclusion, while SFT can be utilized independently of the RLHF pipeline, it still suffers from several significant limitations. However, it still serves as an integral step in the RLHF pipeline, providing a robust initial policy crucial for subsequent RL training.

6.5 Reward Modeling

Reward as a Proxy for Human Feedback. After the collection of human feedback, the next challenge is training the LM effectively. Although SFT offers a straightforward method, its effectiveness is limited by the volume of human feedback. In contrast, RLHF introduces an RM to emulate human feedback, thereby acting as a stand-in for the true reward function, i.e., the actual human feedback. This RM, usually much smaller than the LM, facilitates fine-tuning the LM using feedback generated by it on new model outputs, avoiding the need for additional costly human annotation. In practice, using an RM over SFT has been found more data-efficient [153].

Training a Reward Model. The RM is a fine-tuned LM that assigns a scalar reward score to an input-output pair. The last embedding layer is replaced with a single projection layer that outputs this scalar reward. While the RM can learn from various types of feedback, recent studies highlight the simplicity and effectiveness of preference-based feedback [12, 136]. This approach involves fine-tuning the initialized RM to predict the preference between two trajectories (output text) given the same input prompt or context. The reward is typically modeled as a **Bradley-Terry-Luce** (**BTL**) model [21], where the probability of preferring one trajectory over another is a function of the difference in their reward scores. Mathematically, this can be represented as

$$\Pr((o \succ o', c) \mid \phi) = \sigma[R_\phi(c, o) - R_\phi(c, o')] \tag{10}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, o and o' represent the two trajectories, and their rewards are represented as $R_\phi(c, o)$ and $R_\phi(c, o')$, respectively. This form of reward modeling has been found to provide smoother rewards and is less noisy [30]. A similar method can then be used for ranking between k trajectories (k > 2), where the reward is modeled as a **Plackett–Luce** (**PL**) model [109, 144]. Moreover, [219] provides theoretical proof of convergence guarantees under the Maximum Likelihood estimate of both BTL and PL models.

The size and initialization of the RM are critical determinants of its performance. While smaller RMs are easier to train, scaling laws suggest that larger models yield better agreement with actual

human preferences [8]. However, Ouyang et al. [136] found that training very large RMs can be unstable and result in overfitting. Instead, they report good performance even when using an RM that is 30 times smaller than the policy model.

Regarding initialization, multiple methods have been proposed. While Ziegler et al. [222] fine-tune a PLM on preference data collected on model-generated outputs, Ouyang et al. [136] train a GPT-3 based RM on publicly available datasets. However, only a slight advantage was found over using PLMs or supervised-fine-tuned models. Leveraging publicly available preference datasets (such as ranked answers from StackOverflow), as suggested by Askell et al. [8], notably enhances RM performance, especially for smaller models and datasets.

Challenges in Reward Modeling. The RM is initially trained on a selected set of input prompts and corresponding initial model outputs. As the model training progresses, it is crucial for the RM to generalize to new model outputs and potentially new input prompts. We refer readers to Section 5.1 for a deeper theoretical exploration of this aspect.

Regarding the generalization capabilities of RMs, Ouyang et al. [136] present findings that demonstrate high generalization to held-out test labelers. This capability is of paramount importance since a majority of the inputs encountered during LM training would be out-of-distribution w.r.t. the RM training phase. Generalization capability depends on various factors such as the dataset's size, the amount of noise in the feedback dataset, and the characteristics of the pretrained RM.

Moreover, the robustness and calibration of the RMs with respect to actual human preferences are essential for their effectiveness. A well-calibrated RM should accurately predict the probability of a human preferring one output over another. Bai et al. [12] discovered that when training solely on a helpfulness feedback dataset, their model exhibits strong calibration. However, when trained on a mixture of helpfulness and harmlessness datasets, the model is underconfident in its predictions. To assess robustness, a common practice involves evaluating the policy model trained using the RM.

To assess robustness, a common practice involves evaluating the policy model trained using the RM. Interestingly, Bai et al. [12] discerned that smaller RMs and higher rewards correlate with decreased robustness. This phenomenon arises from the RM's initial training on model outputs with naturally low rewards. To address this distribution shift, an approach involving iterated training of the RM is proposed (see Section 6.6). In summation, the discussion underscores that the trained RM on preferences is an imperfect proxy of human feedback, especially in out-of-domain cases.

Moving Beyond Scalar Rewards. Apart from providing a single scalar reward at the end of a trajectory (complete text output), several methods model a more fine-grained approach. References [84, 186] provide a segment-level reward during training, a method also known as process supervision. Interestingly, while [186] did not find any major downstream performance improvement with their method, [97] used similar methodology but instead trained larger models on a larger feedback dataset coupled with evaluation on a more difficult task found segment-level feedback to be significantly more useful. Reference [163] uses language feedback from another LLM that implicitly acts like an RM for the training of the policy model.

While ideally, as discussed in Section 4, the RM provides a dual-purpose reward taking into account both the task information (e.g., summarization task) and the task-specific evaluation (a condescending summary is rewarded less than a neutral summary). However, diversifying the approach, some strategies involve the use of multiple RMs, each specializing in distinct characteristics or specific tasks. References [154, 197] demonstrate the efficacy of training separate RMs for specific attributes such as coherency and factuality. Similarly, [53] introduces two RMs–one for preference and another for rule violation in dialogue generation. They found using two models over one to be more effective, likely because of a smaller feedback dataset. Further, since the preference-based RM provides a delayed reward (reward provided at the end of the whole trajectory), the A2C

algorithm, when used for sequence modeling [10] proposes potential-based reward shaping, where intermediate generations are also rewarded.

In conclusion, the reward modeling process is a critical component of RLHF which involves the training of a model to emulate human feedback, thereby acting as a surrogate for the true reward function. The size, initialization, and generalization capabilities of the RM are all crucial factors that influence its performance. The RM must be robust, well-calibrated, and additionally can provide more fine-grained feedback to the policy model training.

6.6 RLHF Finetuning of Language Models

The trained RM is utilized for finetuning the LM. Framing the task as RL, with the LM as the policy, algorithms such as PPO and **Advantage Actor-Critic** (**A2C**) [10, 165] are used to update the parameters of the LM such that the generated outputs maximize the obtained reward. These are gradient-based methods, called policy-gradient algorithms, that directly update the parameters of the policy using the evaluative reward feedback.

6.6.1 Training Procedure. The pre-trained/SFT LM is prompted with contexts/prompts from a prompting dataset. The prompting dataset may or may not be identical to the one used for collecting human demonstrations in the SFT phase [136]. The model outputs, along with the inputs, are passed to the RM that generates a scalar output indicating the reward for this input-output pair. The reward is used as evaluative feedback to update the parameters of the LM using suitable RL algorithms that result in increasing the likelihood of the generation of more rewarding outputs. We next discuss a few commonly used RL algorithms for the process.

6.6.2 Training Algorithms. The commonly used policy-gradient algorithms for aligning LLMs using RLHF are PPO and A2C [10, 165]. Both fall under the category of actor-critic algorithms. These algorithms consist of two main components: the critic learns the expected cumulative reward for an input-output pair, called the value function, and the actor is the LLM policy that gets updated based on the cumulative reward estimates obtained from the critic. The reward values are obtained from the previously trained RM, which is kept frozen during the RL training. As the LLM encounters more interactions and collects more reward feedback, it uses the data to update the value function and the LLM policy parameters. The training objective aims to update the parameters of the policy to increase the expected cumulative reward of the LLM policy. A2C and A3C [119] use an estimate of the *advantage* of taking an action instead of the action-value function for that action as a way of incurring lesser variance in policy gradient estimation. PPO additionally constrains the policy update at each iteration from straying too far by using a clipped objective [165]. This helps provide additional stability to the training. Training LLMs at a large scale requires an immense engineering effort, and practical implementations of these algorithms require domain-specific variations. While major progress has been made toward efficient training and inference of LLMs [66, 88, 121, 122, 178, 199, 200, 206, 209], there is still a lot of scope for improvement in the sample efficiency and stability of training algorithms for RLHF. Recent work has addressed these challenges with different variants of these algorithms tackling different aspects ranging from practical implementation issues such as high memory usage [159], changes specific for NLP [153, 196], training instability [216]. One notable example is GRPO [167], which eliminates the need for a separate value model, which group multiple rollouts for a single input prompt and uses the mean of rewards as a baseline. The method has shown remarkable performance in verifiable domains such as math and code [34]. While it reduces memory requirements, it increases computational cost due to need of multiple online rollouts. We refer readers to [192] for a comprehensive survey of policy gradient algorithms, and further compare several modifications to these conventional algorithms in Section 6.10.

6.6.3 Improving Training Stability. Imperfections in the RM reduce the effectivity of the training algorithms, as the value functions learned, and in turn thus the gradient updates, become inaccurate. Thus, using the aforementioned algorithms with the learned RM may lead the LM to exploit the imperfections and generate nonsensical text, often called "reward overoptimization". This can be mitigated with appropriate regularization during training. As the pre-trained or SFT model (policy) is already a highly capable LLM, Jaques et al. [71] propose using a copy of the initial model to regularize training. The aim is to ensure that even as the policy parameters are updated to maximize reward, the outputs of the updated policy do not stray too far from the initial policy. In particular, an additional regularization term of the **Kullback–Leibler** (**KL**) divergence between the policy being trained and the initial policy is added to the RL training objective in the form of a reward penalty, commonly called the KL penalty. Theoretically, the addition of this KL penalty has been shown to be similar to performing Bayesian inference [83] on the model. A hyperparameter β controls the weight of this KL penalty regularization during training. Further, it is common to compare different variants of RL algorithms at a fixed KL distance from the initial model, with the aim of maximizing the reward with the lowest possible KL divergence.

6.6.4 Iterated RLHF. As training progresses, the RM can become miscalibrated with human preferences at higher rewards [12]. This is because the RM was trained on outputs from the initial model, which inherently have low-valued rewards. Consequently, several methods [12, 177] have employed an iterative training approach, where new outputs are generated by the updated policy, which are then annotated by humans for feedback. The RM is then retrained based on this new human feedback, followed by training of the policy model. This process, referred to as Iterated-RLHF or Online-RLHF, is repeated for several iterations. Although effective, this procedure is naturally expensive and time-consuming.

Limitations in Current RLHF practices. Despite the impressive results achieved by RLHF in practice, it is an unstable training process [28]. Moreover, it is highly sensitive to hyperparameters, necessitating a significant amount of hyperparameter tuning [149, 210]. Furthermore, the generalization capabilities of RLHF and other issues, such as underperformance on metrics not captured by the RM, warrant further investigation. A comprehensive examination of these aspects is discussed in Section 6.8.

6.7 Practical Applications of RLHF

Several recent works have effectively used RLHF in the post-training pipeline of improving frontier LLMs. For instance, Llama-2 [184] used RLHF for improving factuality, safety, and helpfulness for typical chat-based prompts. For instance, Llama-2-7B's performance on TruthfulQA [100] improved from 33% to 57% with RLHF and toxicity generation decreased from 22% to near 0 percentage on ToxiGen benchmark [63]. Similar procedure was used in Llama-3 [55] for a larger variety of tasks such as general english, coding, reasoning, multilinguality, tool use highlighting the large scale potential of RLHF. Interestingly, both Llama-3 and Tulu-2 [69] utilized some variant of DPO [149] citing significantly lower computational cost while maintaining similar performance. At the same time Tulu-3 [89] used PPO additionally used rule-based rewards in verifiable domains (such as math, instruction following) instead of RMs. Similarly DeepSeek [35] and Qwen [146] series of LLMs have shown remarkable performance improvements using RLHF for a variety of capabilities, including long-text generation, structured data analysis. More recently RL has been extensively used for improving code and math performance in verifiable domains [34, 64, 89, 134, 204]

6.8 Limitations of RLHF Models

Fine-tuning models using RLHF showcase a remarkable ability to align with human preferences and generalize to new scenarios and is more sample-efficient than SFT. Nonetheless, these models

exhibit characteristics and behaviors that warrant careful consideration, prompting the need for further exploration and refinement.

Alignment Capabilities. One intriguing property, referred to as the Alignment Tax, was identified by [136]. The phenomenon reveals that RLHF-trained chat models sometimes perform poorly compared to initial policy in downstream tasks, suggesting a cost linked to aligning human preferences. To mitigate this, they propose incorporating the pre-training objective into RLHF-finetuning, which substantially reduces the Alignment Tax. Moreover, [12] indicates that larger models tend to exhibit lower alignment tax. Reference [12] also observed that RLHF models better align with human preferences as the scales of both the RM and policy model increase. Further, several works have tried mitigating the alignment tax using model merging techniques [101, 130] and using experience replay [210]. Further, Li et al. [94] showed DPO exhibits lower alignment tax than PPO. It is noteworthy, however, that a similar scaling effect could be seen in instruction-finetuned SFT models. A comprehensive comparison of the scaling effects on RLHF versus SFT models is currently lacking in the literature and would make for an intriguing future study.

Generalization Capabilities. RLHF models have exhibited impressive generalization capabilities beyond their training data, including generalization on new prompts and human feedback. For instance, [136] demonstrates RLHF-tuned models answering coding questions and following instructions in multiple languages despite being finetuned only in English and with limited code-related prompts. This suggests that the majority of an LM's capabilities are acquired during pre-training, and RLHF merely aligns these capabilities to elicit desired behavior. However, this generalization can be a double-edged sword, potentially leading to undesirable outcomes, especially when the feedback signal is sparse. For instance, the initial LLaMA2 Chat Model,[4] when prompted "How to kill a process?" refused to answer, drawing ethical concerns, though the intended answer was about terminating a computer process. This behavior likely stems from the model's extended generalization from examples that trained it to reject violent queries. The example further highlights the problems of imperfect rewards leading to misgeneralization, as discussed in Section 5.1. Further, a distributional shift between prompts used for RM finetuning and RLHF training can result in the policy model misaligning with human preferences [12]. Further, during RL training, outputs are sampled from the LM, which is evaluated using the RM. However, deviations in parameters used for sampling outputs from the model during inference from those in training can yield poor results [153].

Diversity and Biases of RLHF model outputs. Another characteristic of RLHF models is their low entropy in output distribution [12], which challenges generating diverse responses [80]. This holds true for both seen and unseen datasets. To address this, entropy regularization techniques are introduced [71, 93] to amplify diversity in the action space, albeit not always resolving the issue [151]. While not conclusive, [12] found that while RLHF models exhibit better sentiment toward all classes, they display similar biases to underlying LLMs when sampling with temperature < 1 (i.e., with low diversity samples). This could be attributed to their lower entropy. Furthermore, while pre-trained models often generate probabilities that are well-calibrated, RLHF models may lose this calibration. For instance, [133] found that for pre-trained GPT-4, the probability of generating an answer is often directly proportional to the probability of it being correct. However, in the case of RLHF models, the distribution is skewed toward more likely answers.

Improving safety and reducing biases in RLHF remains a key challenge. Various approaches have been proposed, such as SAFE RLHF with dual objectives for helpfulness and harmlessness [33], Equilibrate RLHF using fine-grained data-centric alignment [183], and the COBRA framework for

[4]https://huggingface.co/meta-llama/Llama-2-70b-chat-hf

mitigating biases by aggregating RMs [60]. Additionally, RLHF models may inadvertently learn to mislead users by optimizing for perceived over genuine rewards [191]. Addressing ongoing issues like reward hacking and persistent biases is crucial for improving fairness and robustness [24].

Reward Misgeneralization. Reward misgeneralization is a significant challenge where the RM fails to accurately represent true human preferences, often learning spurious correlations instead of underlying human intent [137]. This can occur when the RM overfits to noisy or idiosyncratic preferences [211], or erroneously associates features like response length [174] or specific stylistic choices with quality, leading the policy to exploit these shallow heuristics. Ambiguous preferences and unidentifiable reward structures can exacerbate misgeneralization [143], and surprisingly, overly accurate RMs can paradoxically worsen downstream performance [26]. Techniques like information-theoretic reward modeling aim to improve generalization [118], but ensuring robust RM generalization as the policy explores new outputs remains difficult.

Robustness and Safety. It is imperative to note that the RM is merely an imperfect proxy for real human preferences/feedback. Due to the lack of calibration and robustness of RMs [12], over-optimizing against the RM can render it an ineffective measure (Goodhart's Law). This phenomenon, known as Reward Overoptimization, has been studied in the context of LMs by [32, 52].

Further, training RLHF models in practice is very difficult for practitioners owing to unstable training [28], hyperparameter sensitivity [149, 210], loading multiple models leading to high memory usage [159]. As a result, there have been significant efforts to simplify the training process by learning directly from the available feedback using simpler SFT objectives, as we discuss in Section 6.10.

While RLHF substantially enhances the performance of LLMs and aligns them with human preferences, it is not without its limitations. These include, but are not limited to, issues such as text hallucination [115], bias and toxicity [37, 48, 58], and the generation of harmful text when probed [142, 189]. Despite significant improvements, these models are not fully aligned with human preferences, underscoring the need for continued research in this field.

6.9 Enriching Reward Signals in Reinforcement Learning

Challenges with Sparse Rewards in Traditional RL. RL has conventionally employed delayed and sparse rewards, where agents receive scalar feedback at the end of a trajectory or episode [181]. While this approach is straightforward to implement and aligns with the task objective, it is not without its drawbacks. Sparse rewards can lead to sample-inefficient learning due to extensive exploration requirements [17].They may result in reward hacking, where agents exploit unintended strategies to maximize rewards without solving the intended task [68]. Underspecified rewards, which do not fully capture the desired behavior, can also yield suboptimal or degenerate solutions [59].

Enriching Reward Signals. To mitigate the limitations of sparse rewards, researchers have explored various methods for providing richer feedback in environments with inherently sparse rewards. These approaches include reward shaping, where the original reward signal is augmented with additional feedback [56, 125]; intrinsic motivation, which encourages exploration and learning through internal rewards based on novelty, curiosity, or learning progress [17, 135, 140]; and multi-objective optimization with multiple reward signals [157, 158]. Hierarchical RL, which decomposes complex tasks into simpler subtasks with their own reward structures, has also been investigated [16, 40]. Moreover, richer forms of feedback, such as learning from corrections [14, 70], demonstrations [156], and language feedback [49, 112], have proven beneficial.

Implications for RLHF in LLMs. Current RLHF pipelines for LLMs primarily rely on sparse rewards provided at the end of an episode, with RMs trained using sparse preference-based feedback. Similar

challenges observed in traditional RL have also been identified in RLHF-tuned LLMs. Some progress has been made in learning from feedback for multi-objective optimization [154], language feedback [162], corrective feedback [110, 170], and denser rewards [197]. Future research should explore the integration of these techniques to address the unique challenges in training LLMs with RLHF, potentially improving generalization and robustness.

6.10 Moving Beyond Conventional RL Training

While RLHF has been very successful, it still results in unstable training [28], is hyperparameter sensitive [149, 210], has high memory usage [159] making it difficult for practitioners to actually use it. As a result, there have been significant efforts to simplify the training process by learning directly from the available feedback using simpler SFT objectives.

6.10.1 Alternatives to RL Using Reward Model. Once, an RM is trained, it is not necessary to perform the RLHF-based training. Instead, an alternate approach during inference is to sample multiple outputs from the LLM and rank them using the RM [31, 123]. This is also called best-on-n sampling or rejection sampling. If sampling multiple outputs, it is important to ensure diversity of outputs by adjusting the sampling parameters (such as higher temperature). This approach is often considered as either a baseline or augmented with RLHF-trained models for better inference-time results.

Further, various works [41, 176, 210] use the trained RM to rank multiple responses and use the signal from the ranked responses to train the policy model, without using an elaborate RL algorithm. In another line of work, RAD [36] uses weighted-decoding of tokens at inference, based on a separately trained RM.

6.10.2 Alternatives to RL without Explicit Reward Models. In this section, we discuss alternative methods to align LMs with human feedback that do not rely on RMs. While RLHF-PPO has shown promising results, it suffers from sensitivity to hyperparameters, the need for training additional models, and potential misalignment of the RM [138, 149, 174, 197, 220]. To address these issues, recent research has explored various techniques that directly incorporate human feedback into the training process, without relying on additional RMs.

A straightforward approach is SFT on positive demonstrations from human feedback, such as instruction-finetuned models [27, 136, 217]. However, this method does not utilize negative feedback, which is crucial for training robust models that can handle adversarial situations and identify and correct errors.

Recent works, such as [104, 213], provide both positive and negative demonstrations/feedback and maximize the likelihood of generating positive/preferred output. These methods have shown better performance than RLHF methods on summarization and dialogue tasks. Zhao et al. [214] demonstrate that **Sequence Likelihood calibration** (**SLiC**) [215] can be used to train models on off-policy offline data collected for different models, resulting in better performance than RLHF-based methods on summarization tasks. SLiC uses a ranking calibration loss that contrasts positive and negative sequences while motivating the model to predict the positive class. Further, RSO [106] improves policy learning in SLiC by using statistical rejection sampling from the policy.

Azar et al. [9], Rafailov et al. [149] further reformulate the objective encoded in the RLHF PPO algorithm, and train the model directly on the new objective, without the need for a separate RM. This follows the intuition, that the policy model can be implicitly used as an RM for training itself based on the collected feedback. However, the results are preliminary, and extending to out-of-distribution prompts may not be possible without the introduction of an explicit RM. Nonethless, subsequent works have proposed several modifications to the DPO objective. KTO [44] extends this framework by introducing a prospect-theoretic loss, arguing that it better captures human preferences and improves alignment by incorporating cognitive biases, such as loss aversion,

directly into the optimization. SimPO [116] further simplifies the training procedure by removing the dependence on a reference policy altogether, instead relying on a margin-based loss defined only over the model's outputs, thus reducing memory requirements and showing performance improvements.

Another line of research focuses on refining model-generated responses using human-encoded principles or feedback. References [11, 87] propose a framework where a list of human-encoded principles (Constitution) guide the model to critique its generations and self-refine the responses. The model is then fine-tuned on the refined responses. Self-Align [179] follows a similar procedure but further removes the need to start with an RLHF-finetuned model. They fine-tune the pretrained LLaMA [185] base model using less than 300 lines of human feedback (in the form of constitutional principles) and achieve performance comparable to state-of-the-art models in terms of helpfulness and harmlessness.

Another direction of work learns to generate or select good feedback for model outputs and apply it to refine LM outputs. Reference [162] takes a similar refinement approach but utilizes available summarization feedback. The initial model is conditioned on input, feedback, and output, generating multiple refinements. The model is then fine-tuned on refinements with the highest similarity to human feedback. Reference [105] aligns human moral values by modeling DP (dynamic-programming) based edits from unaligned source text to target aligned text. The model is then fine-tuned on the refinements generated by the edits, using RL for the second part of the process. Reference [202] fine-tunes a dialogue model using multi-modal feedback with the DIRECTOR method [6], which models both negative and positive sequence labeling directly in the LM head.

In summary, these alternative methods generate new data based on feedback or guidelines and then use it to fine-tune the model. These approaches reduce the reliance on RMs and have shown promising results in some tasks, making them a viable alternative to RLHF-PPO. While these models are easier to train and help in alleviating many drawbacks of RLHF, the evaluation performed has been performed only on specific domains, and constrained settings. Moreover, other in-depth analysis such as sample efficiency and properties exhibited by these models, especially on out-of-distribution data needs to be explored further.

6.11 Comparison to Related Surveys

Our work builds upon and differentiates from prior surveys in several key ways. Foundational surveys on RL [7, 72] provide broad overviews of classical and deep RL, respectively, but do not focus on human feedback or LLMs. Recent literature surveys have also focussed on integrating (human) feedback specifically for LMs [47, 188], but are limited to feedback without providing a comprehensive survey of complete RLHF pipelines and alternatives. Concurrent to our work, Kaufmann et al. [75] provide an extensive overview of RLHF but do not critically examine foundational statistical assumptions or limitations in depth. In contrast, our work explicitly analyzes the theoretical underpinnings of RLHF, particularly focusing on RM assumptions, generalization issues, and feedback sparsity, providing deeper and broader insights.

7 Discussion and Conclusion

In this work, we explore the fundamental aspects of RLHF, aiming to clarify its mechanisms and limitations. We highlight the underlying assumptions necessary for RLHF and examine the impact of different implementation choices, shedding light on the workings of this approach. Our analysis naturally focuses on the RMs, which constitute the core component of RLHF. We introduce the concept of *oracular rewards*, which represent the oracular reward signals that RMs should approximate. The challenges encountered in learning these reward functions highlight both the practical and fundamental limitations of RLHF, as thoroughly analyzed by Casper et al. [24].

Our comprehensive review of the existing literature traces the development of RLHF from its inception to the recent advancements. We cover various aspects: the types of feedback, the details and variations of training algorithms, and alternative methods for achieving alignment without using RL. In related work, Kaufmann et al. [75] extensively survey RLHF, highlighting its evolution from preference-based learning.

Despite the numerous variations of RLHF, the core principle of learning from evaluative feedback remains unchanged. This form of learning is naturally suited to RL, while the specifics of agent formulation, the nature of reward feedback, and environment definition continue to evolve. We anticipate the reduction of reliance on human (or AI) feedback by using existing knowledge sources to construct rewards, which is one of the most promising directions for future efforts to enhance the impact of RLHF. Additionally, improving reward encoding mechanisms to better reflect the diversity of human preferences is an important area for further research.

As RLHF continues to advance and reach its full potential, supported by research in these areas, the use of LLMs is also expanding. Until we fully understand the implications of RLHF, it is crucial to develop robust methods for quantifying uncertainty in the outputs generated by an LLM. Such techniques would enable us to identify and address low confidence outputs, which is especially important in safety-critical applications. Ultimately, understanding its implications becomes paramount as advancements in RLHF increasingly influence industries and economies. Thus, research in this field is critical in shaping the future of large-scale language modeling and its societal impact.

Acknowledgments

We thank Khanh Nguyen for extensive and insightful feedback on earlier versions of the draft. We also thank Wenlong Zhao, Tuhina Tripathi, and Abhiman Neelakanteswara for their help with improving the clarity of the manuscript.

References

[1] Pranjal Aggarwal, A. Deshpande, and Karthik Narasimhan. 2023. SemSup-XC: Semantic supervision for zero and few-shot extreme classification. In *Proceedings of the International Conference on Machine Learning*. Retrieved from https://api.semanticscholar.org/CorpusID:256274863

[2] Anirudh Ajith, Chris Pan, Mengzhou Xia, Ameet Deshpande, and Karthik Narasimhan. 2024. InstructEval: Systematic evaluation of instruction selection methods. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 4336–4350. https://doi.org/10.18653/v1/2024.findings-naacl.270

[3] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. 2023. Self-Consuming generative models go MAD. arXiv:2307.01850 [cs.LG]. Retrieved from https://arxiv.org/abs/2307.01850

[4] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. 2017. Hindsight experience replay. *Advances in Neural Information Processing Systems* 30 (2017), 5048–5058.

[5] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Tachard Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Z. Chen, et al. 2023. PaLM 2 Technical Report.

[6] Kushal Arora, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. 2022. Director: Generator-classifiers for supervised language modeling. In *Proceedings of the AACL*.

[7] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38. https://doi.org/10.1109/MSP.2017.2743240

[8] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, T. J. Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *ArXiv abs/2112.00861* (2021).

[9] Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. arXiv:2310.12036 [cs.AI]. Retrieved from https://arxiv.org/abs/2310.12036

[10] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. arXiv:1607.07086. Retrieved from https://arxiv.org/abs/1607.07086

[11] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, John Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: harmlessness from AI feedback. *ArXiv abs/2212.08073* (2022).

[12] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv abs/2204.05862* (2022).

[13] Alexandre Bailly, Corentin Blanc, Élie Francis, Thierry Guillotin, Fadi Jamal, Béchara Wakim, and Pascal Roy. 2022. Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine* 213 (2022), 106504.

[14] Andrea V. Bajcsy, Dylan P. Losey, Marcia Kilchenman O'Malley, and Anca D. Dragan. 2017. Learning robot objectives from physical human interaction. In *Proceedings of the Conference on Robot Learning*. Retrieved from https://api.semanticscholar.org/CorpusID:28406224

[15] Peter Barnett, Rachel Freedman, Justin Svegliato, and Stuart Russell. 2023. Active reward learning from multiple teachers. arXiv:2303.00894. Retrieved from https://arxiv.org/abs/2303.00894

[16] Andrew G. Barto and Sridhar Mahadevan. 2003. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems* 13, 4 (2003), 41–77. https://api.semanticscholar.org/CorpusID:386824

[17] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. 2016. Unifying count-based exploration and intrinsic motivation. In *Proceedings of the NIPS*. Retrieved from https://api.semanticscholar.org/CorpusID:8310565

[18] Dimitri Bertsekas and John N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific.

[19] Christopher Bishop. 2006. Pattern recognition and machine learning. *Springer Google Schola* 2 (2006), 531–537.

[20] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. 2023. Training diffusion models with reinforcement learning. *ArXiv abs/2305.13301* (2023).

[21] Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. *The Method of Paired Comparisons. Biometrika* 39, 3/4 (1952), 324.

[22] Daniel S. Brown and Scott Niekum. 2019. Deep Bayesian reward learning from preferences. arXiv:1912.04472. Retrieved from https://arxiv.org/abs/1912.04472

[23] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *ArXiv abs/2005.14165* (2020).

[24] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. arXiv:2307.15217 [cs.AI].

[25] Angelica Chen. 2023. Improving code generation by training with natural language feedback. arXiv:2303.16749. Retrieved from https://arxiv.org/abs/2303.16749. https://api.semanticscholar.org/CorpusID:257804798

[26] Yanjun Chen, Dawei Zhu, Yirong Sun, Xinghao Chen, Wei Zhang, and Xiaoyu Shen. 2024. The accuracy paradox in RLHF: When better reward models don't yield better language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.), Association for Computational Linguistics, Miami, Florida, USA, 2980–2989. DOI: https://doi.org/10.18653/v1/2024.emnlp-main.174

[27] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. *Blog post*, March 30 (2023). https://lmsys.org/blog/2023-03-30-vicuna/

[28] Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2019. On the weaknesses of reinforcement learning for neural machine translation. arXiv:1907.01752. Retrieved from https://arxiv.org/abs/1907.01752

[29] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling language modeling with pathways. *ArXiv abs/2204.02311* (2022).

[30] Paul Francis Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. arXiv:1706.03741. Retrieved from https://arxiv.org/abs/1706.03741

[31] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. arXiv:2110.14168. Retrieved from https://arxiv.org/abs/2110.14168

[32] Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2023. Reward model ensembles help mitigate overoptimization. arXiv:2310.02743 [cs.LG]. Retrieved from https://arxiv.org/abs/2310.02743

[33] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe RLHF: Safe reinforcement learning from human feedback. arXiv:2310.12773 [cs.AI]. Retrieved from https://arxiv.org/abs/2310.12773

[34] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, et al. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv:2501.12948 [cs.CL]. https://arxiv.org/abs/2501.12948

[35] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, et al. 2025. DeepSeek-V3 Technical Report. arXiv:2412.19437 [cs.CL]. https://arxiv.org/abs/2412.19437. arXiv:2310.09520 [cs.CL].

[36] Haikang Deng and Colin Raffel. 2023. Reward-Augmented decoding: Efficient controlled text generation with a unidirectional reward model. arXiv:2310.09520 [cs.CL]. Retrieved from https://arxiv.org/abs/2310.09520

[37] A. Deshpande, Vishvak Murahari, Tanmay Rajpurohit, A. Kalyan, and Karthik Narasimhan. 2023. Toxicity in ChatGPT: Analyzing persona-assigned language models. arXiv:2304.05335. Retrieved from https://arxiv.org/abs/2304.05335. https://api.semanticscholar.org/CorpusID:258060002

[38] Ameet Deshpande, Tanmay Rajpurohit, Karthik Narasimhan, and Ashwin Kalyan. 2023. Anthropomorphization of AI: Opportunities and risks. arXiv:2305.14784. Retrieved from https://arxiv.org/abs/2305.14784

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805. Retrieved from https://arxiv.org/abs/1810.04805

[40] Thomas G. Dietterich. 1999. Hierarchical reinforcement learning with the MAXQ value function decomposition. arXiv:9905014. Retrieved from https://arxiv.org/abs/9905014. https://api.semanticscholar.org/CorpusID:57341

[41] Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and T. Zhang. 2023. RAFT: Reward rAnked FineTuning for generative foundation model alignment. arXiv:2304.06767. Retrieved from https://arxiv.org/abs/2304.06767

[42] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2021. GLaM: Efficient scaling of language models with mixture-of-experts. *ArXiv abs/2112.06905* (2021).

[43] Esin Durmus, Karina Nyugen, Thomas Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, et al. 2023. Towards measuring the representation of subjective global opinions in language models. arXiv:2306.16388. Retrieved from https://arxiv.org/abs/2306.16388. https://api.semanticscholar.org/CorpusID:259275051

[44] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: Model alignment as prospect theoretic optimization. arXiv:2402.01306 [cs.LG]. Retrieved from https://arxiv.org/abs/2402.01306

[45] Xiang Fan, Yiwei Lyu, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2022. Nano: Nested human-in-the-loop reward learning for few-shot language model control. arXiv:2211.05750. Retrieved from https://arxiv.org/abs/2211.05750

[46] Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and André F. T. Martins. 2022. Quality-aware decoding for neural machine translation. arXiv:2205.00978. Retrieved from https://arxiv.org/abs/2205.00978

[47] Patrick Fernandes, Aman Madaan, Emmy Liu, António Farinhas, Pedro Henrique Martins, Amanda Bertsch, José G. C. de Souza, Shuyan Zhou, Tongshuang Sherry Wu, Graham Neubig, et al. 2023. Bridging the gap: A survey on integrating (human) feedback for natural language generation. arXiv:2305.00955. Retrieved from https://arxiv.org/abs/2305.00955. https://api.semanticscholar.org/CorpusID:258426970

[48] Emilio Ferrara. 2023. Should ChatGPT be biased? Challenges and risks of bias in large language models. arXiv:2304.03738. Retrieved from https://arxiv.org/abs/2304.03738. https://api.semanticscholar.org/CorpusID:258041203

[49] Daniel Fried, Jacob Andreas, and Dan Klein. 2017. Unified pragmatic models for generating and following instructions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. Retrieved from https://api.semanticscholar.org/CorpusID:21015570

[50] Deep Ganguli, Liane Lovitt, John Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Benjamin Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *ArXiv abs/2209.07858* (2022). https://api.semanticscholar.org/CorpusID:252355458

[51] Ge Gao, Hung-Ting Chen, Yoav Artzi, and Eunsol Choi. 2023. Continually improving extractive QA via human feedback. arXiv:2305.12473. Retrieved from https://arxiv.org/abs/2305.12473

[52] Leo Gao, John Schulman, and Jacob Hilton. 2022. Scaling laws for reward model overoptimization. arXiv:2210.10760. Retrieved from https://arxiv.org/abs/2210.10760

[53] Amelia Glaese, Nathan McAleese, Maja Trkebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. 2022. Improving alignment of dialogue agents via targeted human judgements. *ArXiv abs/2209.14375* (2022).

[54] Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the LAW@ACL*.

[55] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The Llama 3 herd of models. arXiv:2407.21783 [cs.AI]. https://arxiv.org/abs/2407.21783

[56] Marek Grześ. 2017. Reward shaping in episodic reinforcement learning. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (São Paulo, Brazil) (AAMAS'17)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 565–573.

[57] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary LLMs. arXiv:2305.15717 [cs.CL]. Retrieved from https://arxiv.org/abs/2305.15717

[58] Shashank Gupta, Vaishnavi Shrivastava, Ameet Deshpande, Ashwin Kalyan, Peter Clark, Ashish Sabharwal, and Tushar Khot. 2023. Bias runs deep: Implicit reasoning biases in persona-assigned LLMs. arXiv:2311.04892. Retrieved from https://arxiv.org/abs/2311.04892.

[59] Dylan Hadfield-Menell, Smitha Milli, P. Abbeel, Stuart J. Russell, and Anca D. Dragan. 2017. Inverse reward design. arXiv:1711.02827. Retrieved from https://arxiv.org/abs/1711.02827. https://api.semanticscholar.org/CorpusID:3805733

[60] Zafaryab Haider, Md Hafizur Rahman, Vijay Devabhaktuni, Shane Moeykens, and Prabuddha Chakraborty. 2025. A framework for mitigating malicious RLHF feedback in LLM training using consensus based reward. *Scientific Reports* 15, 1 (2025), 9177. DOI: https://doi.org/10.1038/s41598-025-92889-7

[61] Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazaré, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot!. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[62] Austin W. Hanjie, A. Deshpande, and Karthik Narasimhan. 2022. SemSup: Semantic supervision for simple and scalable zero-shot generalization. Retrieved from https://api.semanticscholar.org/CorpusID:255595954

[63] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. arXiv:2203.09509 [cs.CL]. Retrieved from https://arxiv.org/abs/2203.09509

[64] Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. arXiv:2403.04642 [cs.LG]. Retrieved from https://arxiv.org/abs/2403.04642

[65] Danny Hernandez, Tom B. Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, T. J. Henighan, Tristan Hume, et al. 2022. Scaling laws and interpretability of learning from repeated data. *ArXiv abs/2205.10487* (2022).

[66] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. arXiv:1503.02531. Retrieved from https://arxiv.org/abs/1503.02531

[67] 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. 2022. training compute-optimal large language models. *ArXiv abs/2203.15556* (2022).

[68] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. 2018. Reward learning from human preferences and demonstrations in Atari. arXiv:1811.06521. Retrieved from https://arxiv.org/abs/1811.06521. https://api.semanticscholar.org/CorpusID:53424488

[69] Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing LM adaptation with Tulu 2. arXiv:2311.10702 [cs.CL]. Retrieved from https://arxiv.org/abs/2311.10702

[70] Ashesh Jain, Shikhar Sharma, Thorsten Joachims, and Ashutosh Saxena. 2015. Learning preferences for manipulation tasks from online coactive feedback. *The International Journal of Robotics Research* 34, 10 (2015), 1296–1313. https://doi.org/10.1177/0278364915581193

[71] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Àgata Lapedriza, Noah J. Jones, Shixiang Shane Gu, and Rosalind W. Picard. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. arXiv:1907.00456. Retrieved from https://arxiv.org/abs/1907.00456

[72] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4 (1996), 237–285. Retrieved from https://api.semanticscholar.org/CorpusID:1708582

[73] Adam Tauman Kalai and Santosh S Vempala. 2023. Calibrated language models must hallucinate. arXiv:2311.14648. Retrieved from https://arxiv.org/abs/2311.14648

[74] Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv:2001.08361. Retrieved from https://arxiv.org/abs/2001.08361

[75] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. 2023. A survey of reinforcement learning from human feedback. arXiv:2312.14925. Retrieved from https://arxiv.org/abs/2312.14925

[76] Yaser Keneshloo, Tian Shi, Naren Ramakrishnan, and Chandan K. Reddy. 2018. Deep reinforcement learning for sequence-to-sequence models. *IEEE Transactions on Neural Networks and Learning Systems* 31 (2018), 2469–2489.

[77] Urvashi Khandelwal, Kevin Clark, Dan Jurafsky, and Lukasz Kaiser. 2019. Sample efficient text summarization using a single pre-trained transformer. arXiv:1905.08836. Retrieved from https://arxiv.org/abs/1905.08836

[78] Samuel Kiegeland and Julia Kreutzer. 2021. Revisiting the weaknesses of reinforcement learning for neural machine translation. arXiv:2106.08942. Retrieved from https://arxiv.org/abs/2106.08942

[79] Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Min Yoo, and Minjoon Seo. 2023. Aligning large language models through synthetic feedback. arXiv:2305.13735. Retrieved from https://arxiv.org/abs/2305.13735

[80] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the effects of RLHF on LLM generalisation and diversity. arXiv:2310.06452 [cs.LG]. Retrieved from https://arxiv.org/abs/2310.06452

[81] W. Bradley Knox, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro Allievi. 2022. Models of human preference for learning reward functions. arXiv:2206.02231. Retrieved from https://arxiv.org/abs/2206.02231

[82] W. B. Knox and P. Stone. 2008. TAMER: Training an agent manually via evaluative reinforcement. In *Proceedings of the 2008 7th IEEE International Conference on Development and Learning*. 292–297.

[83] Tomasz Korbak, Ethan Perez, and Christopher L. Buckley. 2022. RL with KL penalties is better viewed as Bayesian inference. arXiv:2205.11275. Retrieved from https://arxiv.org/abs/2205.11275

[84] Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Sam Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. arXiv:2302.08582. Retrieved from https://arxiv.org/abs/2302.08582

[85] Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can neural machine translation be improved with user feedback? arXiv:1804.05958. Retrieved from https://arxiv.org/abs/1804.05958

[86] Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. 2018. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. arXiv:1805.10627. Retrieved from https://arxiv.org/abs/1805.10627

[87] Sandipan Kundu, Yuntao Bai, Saurav Kadavath, Amanda Askell, Andrew Callahan, Anna Chen, Anna Goldie, Avital Balwit, Azalia Mirhoseini, Brayden McLean, et al. 2023. Specific versus general principles for constitutional AI. arXiv:2310.13798 [cs.CL].

[88] François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M. Rush. 2021. Block pruning for faster transformers. arXiv:2109.04838. Retrieved from https://arxiv.org/abs/2109.04838

[89] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2025. Tulu 3: Pushing frontiers in open language model post-training. arXiv:2411.15124 [cs.CL]. Retrieved from https://arxiv.org/abs/2411.15124

[90] Carolin (Haas) Lawrence and Stefan Riezler. 2018. Improving a neural semantic parser by counterfactual learning from human bandit feedback. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[91] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating training data makes language models better. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[92] Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? End-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 2443–2453. DOI: https://doi.org/10.18653/v1/D17-1259

[93] Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016. Dialogue learning with human-in-the-loop. arXiv:1611.09823. Retrieved from https://arxiv.org/abs/1611.09823

[94] Shengzhi Li, Rongyu Lin, and Shichao Pei. 2024. Multi-modal preference alignment remedies degradation of visual instruction tuning on language models. arXiv2402.10884. Retrieved from https://arxiv.org/abs/2402.10884

[95] Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2017. Paraphrase generation with deep reinforcement learning. arXiv:1711.00279. Retrieved from https://arxiv.org/abs/1711.00279

[96] Zichao Li, Prakhar Sharma, Xing Han Lu, Jackie Chi Kit Cheung, and Siva Reddy. 2022. Using interactive feedback to improve the accuracy and explainability of question answering systems post-deployment. arXiv:2204.03025. Retrieved from https://arxiv.org/abs/2204.03025. https://api.semanticscholar.org/CorpusID:248006299

[97] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. arXiv:2305.20050. Retrieved from https://arxiv.org/abs/2305.20050

[98] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology* 22, 140 (1932), 55–55.

[99] Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[100] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. arXiv:2109.07958 [cs.CL]. Retrieved from https://arxiv.org/abs/2109.07958

[101] Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, et al. 2024. Mitigating the alignment tax of RLHF. arXiv:2309.06256 [cs.LG]. Retrieved from https://arxiv.org/abs/2309.06256

[102] Bing Liu, Gökhan Tür, Dilek Z. Hakkani-Tür, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

[103] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv:1603.08023. Retrieved from https://arxiv.org/abs/1603.08023

[104] Hao Liu, Carmelo Sferrazza, and P. Abbeel. 2023. Chain of hindsight aligns language models with feedback. arXiv:2302.02676. Retrieved from https://arxiv.org/abs/2302.02676

[105] Ruibo Liu, Chenyan Jia, Ge Zhang, Ziyu Zhuang, Tony X. Liu, and Soroush Vosoughi. 2023. Second thoughts are best: Learning to re-align with human values from text edits. arXiv:2301.00355. Retrieved from https://arxiv.org/abs/2301.00355

[106] Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2023. Statistical rejection sampling improves preference optimization. arXiv:2309.06657 [cs.CL]. Retrieved from https://arxiv.org/abs/2309.06657

[107] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Found. Trends Inf. Retr.* 3, 3 (March 2009), 225–331. https://doi.org/10.1561/1500000016

[108] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692 [cs.CL]. Retrieved from https://arxiv.org/abs/1907.11692

[109] R. Duncan Luce. 1979. Individual choice behavior: A theoretical analysis.

[110] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-Refine: Iterative refinement with self-feedback. arXiv:2303.17651. Retrieved from https://arxiv.org/abs/2303.17651

[111] Andrei Andreevich Markov. 1954. The theory of algorithms. *Trudy Matematicheskogo Instituta Imeni VA Steklova* 42 (1954), 3–375.

[112] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. In *Proceedings of the International Conference on Machine Learning*. Retrieved from https://api.semanticscholar.org/CorpusID:2408319

[113] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On faithfulness and factuality in abstractive summarization. arXiv:2005.00661. Retrieved from https://arxiv.org/abs/2005.00661

[114] Daniel McFadden. 1981. Econometric models of probabilistic choice.

[115] Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. arXiv:2305.14552. Retrieved from https://arxiv.org/abs/2305.14552. https://api.semanticscholar.org/CorpusID:258865517

[116] Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. arXiv:2405.14734 [cs.CL]. Retrieved from https://arxiv.org/abs/2405.14734

[117] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nathan McAleese. 2022. Teaching language models to support answers with verified quotes. *ArXiv abs/2203.11147* (2022).

[118] Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. 2024. InfoRM: Mitigating reward hacking in RLHF via information-theoretic reward modeling. arXiv:2402.09345 [cs.LG]. Retrieved from https://arxiv.org/abs/2402.09345

[119] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1928–1937.

[120] Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. 2023. Nash learning from human feedback. arXiv:2312.00886. Retrieved from https://arxiv.org/abs/2312.00886. https://api.semanticscholar.org/CorpusID:265609682

[121] Vishvak Murahari, Ameet Deshpande, Carlos E. Jimenez, Izhak Shafran, Mingqiu Wang, Yuan Cao, and Karthik Narasimhan. 2023. MUX-PLMs: Pre-training language models with data multiplexing. arXiv:2302.12441. Retrieved from https://arxiv.org/abs/2302.12441

[122] Vishvak Murahari, Carlos E. Jimenez, Runzhe Yang, and Karthik R. Narasimhan. 2022. DataMUX: Data multiplexing for neural networks. In *Proceedings of the 36th Conference on Neural Information Processing Systems*. Retrieved from https://openreview.net/forum?id=UdgtTVTdswg

[123] Reiichiro Nakano, Jacob Hilton, S. Arun Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332. Retrieved from https://arxiv.org/abs/2112.09332

[124] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2019. Deep double descent: where bigger models and more data hurt. arXiv:191202292 [cs, stat]. (2019).

[125] A. Ng, Daishi Harada, and Stuart J. Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the International Conference on Machine Learning*. Retrieved from https://api.semanticscholar.org/CorpusID:5730166

[126] Richard Ngo, Lawrence Chan, and Sören Mindermann. 2022. The alignment problem from a deep learning perspective. arXiv:2209.00626. Retrieved from https://arxiv.org/abs/2209.00626

[127] Duy-Hung Nguyen, Nguyen-Viet-Dung Nghiem, Bao-Sinh Nguyen, Dung Tien Le, Shahab Sabahi, Minh Le Nguyen, and Hung Le. 2022. Make The most of prior data: A solution for interactive text summarization with preference feedback. In *NAACL-HLT*.

[128] Khanh Nguyen, Hal Daumé, and Jordan L. Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. *ArXiv abs/1707.07402* (2017).

[129] Khanh Nguyen, Dipendra Misra, Robert Schapire, Miro Dudík, and Patrick Shafto. 2021. Interactive learning from activity description. In *ICML*.

[130] Michael Noukhovitch, Samuel Lavoie, Florian Strub, and Aaron Courville. 2023. Language model alignment with elastic reset. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (New Orleans, LA, USA) (NIPS'23)*. Curran Associates Inc., Red Hook, NY, USA, Article 152, 23 pages.

[131] Marcus O'Connor. 1989. Models of human behaviour and confidence in judgement: A review. *International Journal of Forecasting* 5, 2 (1989), 159–169. https://doi.org/10.1016/0169-2070(89)90083-6

[132] OpenAI. 2022. ChatGPT. https://openai.com/blog/chatgpt. (2022).

[133] OpenAI. 2023. GPT-4 Technical Report. *ArXiv abs/2303.08774* (2023).

[134] OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, et al. 2024. OpenAI o1 System Card. arXiv:2412.16720 [cs.AI]. https://arxiv.org/abs/2412.16720

[135] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V. Hafner. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation* 11, 2 (2007), 265–286. https://doi.org/10.1109/TEVC.2006.890271

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

[137] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. arXiv:2201.03544 [cs.LG]. Retrieved from https://arxiv.org/abs/2201.03544

[138] Richard Yuanzhe Pang, Vishakh Padmakumar, Thibault Sellam, Ankur P. Parikh, and He He. 2022. Reward gaming in conditional text generation. arXiv:2211.08714. Retrieved from https://arxiv.org/abs/2211.08714

[139] Mihir Parmar, Swaroop Mishra, Mor Geva, and Chitta Baral. 2022. Don't blame the annotator: Bias already starts in the annotation instructions. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.

[140] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 488–489. Retrieved from https://api.semanticscholar.org/CorpusID:20045336

[141] Andi Peng, Besmira Nushi, Emre Kiciman, Kori Inkpen, and Ece Kamar. 2022. Investigations of performance and bias in human-AI teamwork in hiring. arXiv:2202.11812 [cs.HC]. Retrieved from https://arxiv.org/abs/2202.11812

[142] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nathan McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[143] Silviu Pitis. 2023. Failure modes of learning reward models for LLMs and other sequence models. In *Proceedings of the ICML 2023 Workshop The Many Facets of Preference-Based Learning*. Retrieved from https://openreview.net/forum?id=NjOoxFRZA4

[144] R. L. Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society Series C* 24, 2 (June 1975), 193–202. https://doi.org/10.2307/2346567

[145] Dean A. Pomerleau. 1988. Alvinn: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems* 1 (1988), 305–313.

[146] Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2025. Qwen2.5 technical report. arXiv:2412.15115 [cs.CL]. https://arxiv.org/abs/2412.15115

[147] Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

[148] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *ArXiv abs/2112.11446* (2021).

[149] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. arXiv:2305.18290. Retrieved from https://arxiv.org/abs/2305.18290

[150] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv:1910.10683. Retrieved from https://arxiv.org/abs/1910.10683

[151] Anton Raichuk, Piotr Stanczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, L'eonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, and Sylvain Gelly. 2021. What matters for on-policy deep actor-critic methods? A large-scale study. In *Proceedings of the International Conference on Learning Representations*. Retrieved from https://api.semanticscholar.org/CorpusID:233340556

[152] Deepak Ramachandran and Eyal Amir. 2007. Bayesian inverse reinforcement learning. In *Proceedings of the IJCAI*, Vol. 7. 2586–2591.

[153] Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. arXiv:2210.01241. Retrieved from https://arxiv.org/abs/2210.01241

[154] Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. 2023. Rewarded soups: Towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards.

[155] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. arXiv:1511.06732. Retrieved from https://arxiv.org/abs/1511.06732

[156] Desik Rengarajan, Gargi Nikhil Vaidya, Akshay Sarvesh, Dileep M. Kalathil, and Srinivas Shakkottai. 2022. Reinforcement learning with sparse rewards using guidance from offline demonstration. arXiv:2202.04628. Retrieved from https://arxiv.org/abs/2202.04628. https://api.semanticscholar.org/CorpusID:246679865

[157] Diederik M. Roijers. 2016. Multi-objective decision-theoretic planning. Retrieved from https://api.semanticscholar.org/CorpusID:124195290

[158] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. arXiv:1402.0590. Retrieved from https://arxiv.org/abs/1402.0590. https://api.semanticscholar.org/CorpusID:14478191

[159] Michael Santacroce, Yadong Lu, Han Yu, Yuanzhi Li, and Yelong Shen. 2023. Efficient RLHF: Reducing the memory usage of PPO. arXiv:2309.00754 [cs.LG]. Retrieved from https://arxiv.org/abs/2309.00754

[160] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? arXiv:2303.17548 [cs.CL]. Retrieved from https://arxiv.org/abs/2303.17548

[161] Rylan Schaeffer, Mikail Khona, Zachary Robertson, Akhilan Boopathy, Kateryna Pistunova, Jason W. Rocks, Ila Rani Fiete, and Oluwasanmi Koyejo. 2023. Double descent demystified: Identifying, interpreting & ablating the sources of a deep learning puzzle. arXiv:2303.14151. Retrieved from https://arxiv.org/abs/2303.14151

[162] J'er'emy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. Training language models with language feedback.

[163] J'er'emy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. arXiv:2303.16755. Retrieved from https://arxiv.org/abs/2303.16755

[164] Natalie Schluter. 2017. The limits of automatic summarisation according to ROUGE. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.

[165] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. arXiv:1707.06347. Retrieved from https://arxiv.org/abs/1707.06347

[166] Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[167] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv:2402.03300 [cs.CL]. Retrieved from https://arxiv.org/abs/2402.03300

[168] Shiqi Shen, Yong Cheng, Zhongjun He, W. He, Hua Wu, Maosong Sun, and Yang Liu. 2015. Minimum risk training for neural machine translation. arXiv:1512.02433. Retrieved from https://arxiv.org/abs/1512.02433

[169] Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2018. Toward diverse text generation with inverse reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

[170] Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. Retrieved from https://api.semanticscholar.org/CorpusID:258833055

[171] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. arXiv:2305.17493 [cs.LG]. Retrieved from https://arxiv.org/abs/2305.17493

[172] David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. 2021. Reward is enough. *Artificial Intelligence* 299 (2021), 103535. https://doi.org/10.1016/j.artint.2021.103535

[173] Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature* 620, 7972 (1 Aug 2023), 172–180. https://doi.org/10.1038/s41586-02306291-2

[174] Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in RLHF. arXiv:2310.03716 [cs.CL]. Retrieved from https://arxiv.org/abs/2310.03716

[175] Artem Sokolov, Stefan Riezler, and Tanguy Urvoy. 2016. Bandit structured prediction for learning from partial feedback in statistical machine translation. arXiv:1601.04468. Retrieved from https://arxiv.org/abs/1601.04468

[176] Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment. arXiv:2306.17492 [cs.CL]. Retrieved from https://arxiv.org/abs/2306.17492

[177] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan J. Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. arXiv:2009.01325. Retrieved from https://arxiv.org/abs/2009.01325

[178] Yushan Su, Vishvak Murahari, Karthik Narasimhan, and Kai Li. 2023. PruMUX: Augmenting data multiplexing with model compression. arXiv:2305.14706. Retrieved from https://arxiv.org/abs/2305.14706

[179] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David D. Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. arXiv:2305.03047. Retrieved from https://arxiv.org/abs/2305.03047

[180] R. S. Sutton. 2004. The reward hypothesis. Blog post, 1 February 2024. (2004). http://incompleteideas.net/rlai.cs.ualberta.ca/RLAI/rewardhypothesis.html

[181] Richard S. Sutton and Andrew G. Barto. 2005. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks* 16 (2005), 285–286.Retrieved from https://api.semanticscholar.org/CorpusID:9166388

[182] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

[183] Yingshui Tan, Yilei Jiang, Yanshi Li, Jiaheng Liu, Xingyuan Bu, Wenbo Su, Xiangyu Yue, Xiaoyong Zhu, and Bo Zheng. 2025. Equilibrate RLHF: Towards balancing helpfulness-safety trade-off in large language models. arXiv:2502.11555 [cs.AI]. Retrieved from https://arxiv.org/abs/2502.11555

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

[185] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. arXiv:2302.13971. Retrieved from https://arxiv.org/abs/2302.13971

[186] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, L. Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. arXiv:2211.14275. Retrieved from https://arxiv.org/abs/2211.14275

[187] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the NIPS*.

[188] Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. arXiv:2307.12966 [cs.CL]. Retrieved from https://arxiv.org/abs/2307.12966

[189] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does LLM safety training fail? arXiv:2307.02483. Retrieved from https://arxiv.org/abs/2307.02483. https://api.semanticscholar.org/CorpusID:259342528

[190] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. arXiv:2109.01652. Retrieved from https://arxiv.org/abs/2109.01652

[191] Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R. Bowman, He He, and Shi Feng. 2024. Language models learn to mislead humans via RLHF. arXiv:2409.12822 [cs.CL]. Retrieved from https://arxiv.org/abs/2409.12822

[192] Lilian Weng. 2018. Policy gradient algorithms. Blog post, April 8, 2018. lilianweng.github.io (2018). https://lilianweng.github.io/posts/2018-04-08policy-gradient/

[193] Lilian Weng. 2024. Reward hacking in reinforcement learning. Blog post, November 28, 2024. Lil'Log (2024). https://lilianweng.github.io/posts/202411-28-reward-hacking/

[194] Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nissan Stiennon, Ryan Lowe, Jan Leike, and Paul Francis Christiano. 2021. Recursively summarizing books with human feedback. arXiv:2109.10862. Retrieved from https://arxiv.org/abs/2109.10862

[195] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A large language model for finance. arXiv:2303.17564. Retrieved from https://arxiv.org/abs/2303.17564. https://api.semanticscholar.org/CorpusID:257833842

[196] Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2023. Pairwise proximal policy optimization: Harnessing relative feedback for LLM alignment. arXiv:2310.00212. Retrieved from https://arxiv.org/abs/2310.00212. https://api.semanticscholar.org/CorpusID:263334045

[197] Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hanna Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training.

[198] Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. arXiv:2306.01693. Retrieved from https://arxiv.org/abs/2306.01693

[199] Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. [n. d.]. Sheared LLaMA: Accelerating language model pre-training via structured pruning. ([n. d.]).

[200] Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

[201] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023. DoReMi: Optimizing data mixtures speeds up language model pretraining.

[202] Jing Xu, Megan Ung, Mojtaba Komeili, Kushal Arora, Y-Lan Boureau, and Jason Weston. 2022. Learning new skills after deployment: Improving open-domain internet-driven dialogue with human feedback. arXiv:2208.03270. Retrieved from https://arxiv.org/abs/2208.03270

[203] Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. 2023. Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology* 55, 1 (Aug. 2023), 90–112. https://doi.org/10.1111/bjet.13370

[204] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. arXiv:2505.09388 [cs.CL]. https://arxiv.org/abs/2505.09388

[205] Kevin Yang, Nanyun Peng, Yuandong Tian, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[206] Ziqing Yang, Yiming Cui, and Zhigang Chen. 2022. TextPruner: A model pruning toolkit for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Dublin, Ireland, 35–43. DOI: https://doi.org/10.18653/v1/2022.acl-demo.4

[207] Georgios N. Yannakakis and John Hallam. 2011. Ranking vs. preference: A comparative study of self-reporting. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*. Retrieved from https://api.semanticscholar.org/CorpusID:48790

[208] Sanghyun Yi, Rahul Goel, Chandra Khatri, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer Gabriel, and Dilek Z. Hakkani-Tür. 2019. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. In *Proceedings of the International Conference on Natural Language Generation*.

[209] Yichun Yin, Cheng Chen, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2021. AutoTinyBERT: Automatic hyper-parameter optimization for efficient pre-trained language models. 5146–5157.

[210] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Feiran Huang. 2023. RRHF: Rank responses to align language models with human feedback without tears. arXiv:2304.05302. Retrieved from https://arxiv.org/abs/2304.05302

[211] Jiazheng Zhang, Wenqing Jing, Zizhuo Zhang, Zhiheng Xi, Shihan Dou, Rongxiang Weng, Jiahuan Li, Jingang Wang, Mingxu Chai, Shibo Hong, et al. 2025. Two minds better than one: Collaborative reward modeling for LLM alignment. arXiv:2505.10597 [cs.LG]. Retrieved from https://arxiv.org/abs/2505.10597

[212] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. OPT: Open Pre-trained transformer language models. *ArXiv abs/2205.01068* (2022).

[213] Tianjun Zhang, Fangchen Liu, Justin Wong, P. Abbeel, and Joseph Gonzalez. 2023. The wisdom of hindsight makes language models better instruction followers. arXiv:2302.05206. Retrieved from https://arxiv.org/abs/2302.05206

[214] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. SLiC-HF: Sequence likelihood calibration with human feedback. arXiv:2305.10425. Retrieved from https://arxiv.org/abs/2305.10425

[215] Yao Zhao, Misha Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J. Liu. 2022. Calibrating sequence likelihood improves conditional language generation. arXiv:2210.00045. Retrieved from https://arxiv.org/abs/2210.00045

[216] Rui Zheng, Shihan Dou, Songyang Gao, Wei Shen, Bing Wang, Yan Liu, Senjie Jin, Qin Liu, Limao Xiong, Luyao Chen, et al. 2023. Secrets of RLHF in large language models Part I: PPO.

[217] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. LIMA: Less is more for alignment. arXiv:2305.11206 [cs.CL]. Retrieved from https://arxiv.org/abs/2305.11206

[218] Wangchunshu Zhou and Ke Xu. 2020. Learning to compare for better training and evaluation of open domain natural language generation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[219] Banghua Zhu, Jiantao Jiao, and M.I. Jordan. 2023. Principled reinforcement learning with human feedback from pairwise or K-wise comparisons. arXiv:2301.11270. Retrieved from https://arxiv.org/abs/2301.11270

[220] Banghua Zhu, Hiteshi Sharma, Felipe Vieira Frujeri, Shi Dong, Chenguang Zhu, M. I. Jordan, and Jiantao Jiao. 2023. Fine-tuning language models with advantage-induced policy alignment. arXiv:2306.02231. Retrieved from https://arxiv.org/abs/2306.02231

[221] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[222] Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. arXiv:1909.08593. Retrieved from https://arxiv.org/abs/1909.08593

Received 10 November 2024; revised 26 May 2025; accepted 1 June 2025