# ZEROTH-ORDER POLICY GRADIENT FOR REINFORCEMENT LEARNING FROM HUMAN FEEDBACK WITHOUT REWARD INFERENCE

**Qining Zhang & Lei Ying**
Department of Electrical Engineering and Computer Science
University of Michigan – Ann Arbor
Ann Arbor, MI 48105, USA
{qiningz, leiying}@umich.edu

## ABSTRACT

Reward inference (learning a reward model from human preferences) is a critical intermediate step in the *Reinforcement Learning from Human Feedback* (RLHF) pipeline for fine-tuning *Large Language Models* (LLMs). In practice, RLHF faces fundamental challenges such as distribution shift, reward model overfitting, and problem misspecification. An alternative approach is direct policy optimization without reward inference, such as *Direct Preference Optimization* (DPO), which provides a much simpler pipeline and has shown empirical success in LLM applications. However, DPO utilizes the closed-form expression between the optimal policy and the reward function, which is only suitable under the bandit setting or deterministic MDPs. This paper develops two RLHF algorithms without reward inference for *general* RL problems beyond bandits and deterministic MDPs, and *general* preference models beyond the Bradley-Terry model. The key idea is to estimate the local value function difference from human preferences and then approximate the policy gradient with a zeroth-order gradient approximator. For both algorithms, we establish polynomial convergence rates in terms of the number of policy gradient iterations, the number of trajectory samples, and human preference queries per iteration. Numerical experiments in stochastic environments validate the performance of our proposed algorithms, outperforming popular RLHF baselines such as DPO and PPO. Our paper shows there exist provably efficient methods to solve general RLHF problems without reward inference.

## 1 INTRODUCTION

In the past decade, we have witnessed unprecedented success in applying *Reinforcement Learning* (RL) to many applications, such as video games (Knox & Stone, 2008; Warnell et al., 2018), recommendation and search (Zeng et al., 2016; Kohli et al., 2013), autonomous driving (Kiran et al., 2022). RL studies the interaction between decision-making agents and an evolving dynamic environment. At each time step, the agent takes a certain decision (action) given the current state, a reward signal to measure the quality of that decision is provided by the environment. The agent's goal is to learn a policy to maximize the cumulative reward, and the quality of the learned policy will depend on the per-step reward function. In classic RL, this reward function is usually handcrafted by domain experts to ensure it aligns with human interests. However, the problem of identifying a "good" reward function, also referred to as *Inverse Reinforcement Learning* (IRL), is non-trivial and one of the most fundamental problems in the history of RL (Ng & Russell, 2000). In recent years, *Reinforcement Learning from Human Feedback* (RLHF) that uses human preference feedback as a signal to recover a reward function has emerged to fine-tune *Large Language Models* (LLMs), which has delivered significant success (Christiano et al., 2017; Wu et al., 2021; Nakano et al., 2021; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022). RLHF follows the diagram shown in Fig. 1, which includes three major steps (Ouyang et al., 2022): (i) pre-train a policy neural network, (ii) collect sets of trajectory pairs and query human evaluators for preferences over trajectory pairs to train a reward model using maximum likelihood to align with human feedback, and (iii) use policy-optimization-

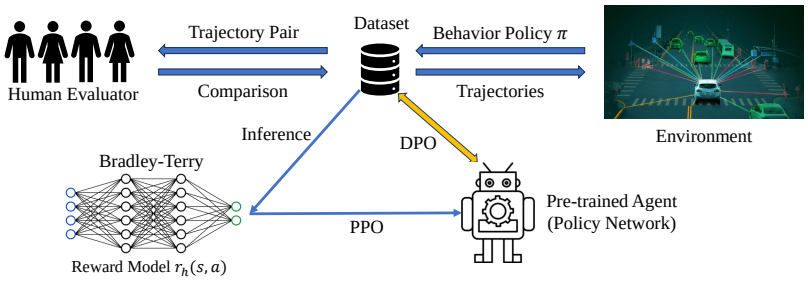



Figure 1: classic policy-based RLHF and DPO: classic RLHF such as PPO involves three steps: (i) policy pre-training, (ii) reward inference, and (iii) policy network training with reward model. DPO does not use a reward network but directly optimizes the policy network from human preferences.

based RL algorithms such as PPO (Schulman et al., 2017) to fine-tune the policy network with the reward signals generated by the reward model. The reward inference intermediate step, which trains the reward network, is crucial to obtaining a high-quality policy through RL in the final step.

**Drawbacks of Reward Inference.** To train a good reward model, i.e., to infer the underlying per-step reward function from human feedback (Christiano et al., 2017; Wang et al., 2023), the most common approach is to assume the feedback is generated based on a preference model such as the *Bradley-Terry* model (Bradley & Terry, 1952), and then maximize the log-likelihood of the collected trajectory comparison dataset accordingly over all possible (parameterized) reward functions. This procedure is indeed analyzed in most theoretical RLHF papers for both offline (Zhu et al., 2023; Zhan et al., 2024a) and online settings (Saha et al., 2023; Zhan et al., 2024b; Wu & Sun, 2024; Wang et al., 2023; Du et al., 2024). However, several challenges occur in practice for reward model training such as double problem misspecification, reward model evaluation without ground truth, distribution shift, and overfitting in joint reward model and policy training (Casper et al., 2023). These drawbacks are also reflected in the theoretical results, e.g., overfitting of the maximum likelihood estimator (MLE) in Zhu et al. (2024). Moreover, similar to the dilemma in IRL, the reward function that could explain human feedback is often not unique, especially when given limited training trajectories (Arora & Doshi, 2021; Ng & Russell, 2000). Some reward models may make it difficult for agents to learn a good RL policy.

**DPO.** To avoid the drawbacks of the reward inference in RLHF, Rafailov et al. (2023) proposed an algorithm called *Direct Preference Optimization* (DPO) which fine-tunes the LLM *directly* from human preferences. Based on the Bradley-Terry preference model and a closed-form expression of the optimal policy given a reference policy and the reward function, DPO constructs a loss function directly from human feedback for learning the optimal policy to avoid reward inference. This provides a much simpler pipeline and has great empirical performance (Rafailov et al., 2023; 2024a;b). However, the closed-form expression of the optimal policy that DPO builds on is only for non-parametric policies and its theoretical justification only works for the bandit setting (Rafailov et al., 2023) or RL problems with deterministic transitions (Rafailov et al., 2024b). It remains an open question how to solve general RLHF problems without reward inference.

**RLHF without Reward Inference.** Recently, value-based RLHF algorithms without global reward inference have been theoretically developed and analyzed (Xu et al., 2020; Zhang et al., 2024a) based on a dueling bandit approach (Bengs et al., 2021). The results, however, only hold for MDPs in tabular settings with finite state and action spaces. Chen et al. (2022) studied the function approximation regime, but their algorithm requires both the true preference model and the transition kernel to belong to a known function class, which is also far from practice. The result also depends on the function class complexity, which is usually large for most function approximators in practice. So far, no provable policy-based algorithm in this category has been developed.

This paper addresses the following important question:

> *Does there exist a provably efficient RLHF approach that does not require a reward model and works for general RL problems such as stochastic MDPs or infinite state and action spaces?*

### 1.1 MAIN CONTRIBUTIONS

DPO (Rafailov et al., 2023) establishes a direct connection between human preferences and RL based on the Bradley-Terry model and the optimal policy in closed form:

$$\pi^*(a|x) \propto \pi_{\text{ref}}(a|x) \exp\left(\frac{1}{\beta} r(x,a)\right), \tag{1}$$

where $r(x,a)$ is the reward in state $x$ with action $a$, $\pi_{\text{ref}}$ is a reference policy and $\pi^*$ is the optimal policy. Based on the direction connection, the policy optimization can be formulated as a direct matching between human preference and the optimal policy with a log-likelihood loss function. In a recent paper (Rafailov et al., 2024a), it has been further shown that DPO solves a KL-divergence-constrained policy optimization problem for the *deterministic* token-level MDP for LLMs, where the next state is deterministic given the current state and action. For general RL problems with parameterized policies, equation 1 does not hold, and it is often hard if not impossible to obtain a "global" function like it that connects the optimal policy and the reward (hence human feedback).

This paper exploits the "local" relation between human feedback and policy optimization. In particular, given a policy $\pi_{\boldsymbol{\theta}}$ and a perturbed version of the policy $\pi_{\boldsymbol{\theta}+\boldsymbol{v}}$ where $\boldsymbol{v}$ is a small perturbation vector, we use human feedback over the trajectories generated from both policies to inform the direction of a more preferred policy. Intuitively, if one trajectory is preferred over the other, the policy that generates this trajectory is likely to have a higher value. Then given a preference model such as the Bradley-Terry model, we can further estimate the value function differences of the two policies, $V(\pi_{\boldsymbol{\theta}+\boldsymbol{v}}) - V(\pi_{\boldsymbol{\theta}})$, where $V(\pi)$ is the value function associated with policy $\pi$. Finally, the value difference can be used as an estimator of policy gradient, $\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}})$, following the zeroth-order optimization approach (Nesterov & Spokoiny, 2017; Ghadimi & Lan, 2013) to improve the policy.

Based on this idea, this paper proposes two RLHF algorithms without reward inference: ***Z**eroth-Order **P**olicy **G**radient* (ZPG) and ***Z**eroth-Order **B**lock-**C**oordinate **P**olicy **G**radient* (ZBCPG), both from Human Feedback. ZBCPG differs from ZPG in its policy perturbation rule, which has lower computational complexity and allows parallel optimization since one can sample multiple perturbed policies to perform policy gradient and aggregate the estimated gradient. Under mild assumptions, both algorithms have the following rate of convergence to a stationary policy:

$$\mathcal{O}\left(\frac{Hd}{T} + \frac{d^2\sqrt{\log M}}{\sqrt{M}} + \frac{Hd\sqrt{d}}{\sqrt{N}}\right),$$

where $d$ is the dimension of policy network parameter $\boldsymbol{\theta}$, $H$ is the planning horizon, $T$ is the number of policy gradient steps, $N$ is the number of policy perturbations each step, and $M$ is the number of human queries for each pair of trajectories.

We remark that Tang et al. (2024b) proposes a similar approach towards utilizing human feedback and a zeroth-order gradient descent algorithm from ranking data. However, they assume an error-free ranking oracle over policies based on their value functions, which makes their problem a deterministic optimization problem and does not apply to trajectory preference data like in RLHF and DPO. This paper studies RLHF with trajectory preferences and quantifies the impacts of stochastic trajectories and human preferences on the rate of convergence of RLHF without reward inference.

## 2 PRELIMINARIES

**Episodic RL:** We consider an episodic RL problem $\mathcal{M} = (\mathbb{S}, \mathbb{A}, H, \boldsymbol{P}, \boldsymbol{\mu}_0)$, where $\mathbb{S}$ is the state space and $\mathbb{A}$ is the action space (both can be continuous), $H$ is the RL planning horizon, $\boldsymbol{P} = \{\boldsymbol{P}_h\}_{h=1}^H$ is the set of transition kernels, and $\boldsymbol{\mu}_0$ is the initial distribution. At the beginning of each episode, the agent will choose a policy $\pi$ represented by $H$ functions $\{\pi_h : \mathbb{S} \to \mathcal{P}(\mathbb{A})\}_{h=1}^H$, where $\mathcal{P}(\mathbb{A})$ denotes the set of all probability distributions over the action space. Then, an initial state $s_1$ is sampled from the initial distribution $\boldsymbol{\mu}_0$. At step $h$, the agent takes an action $a_h = \pi_h(s_h)$ after observing state $s_h$. The environment then moves to the next state $s_{h+1}$ sampled from the distribution $\boldsymbol{P}_h(\cdot|s_h, a_h)$ without revealing any reward feedback. We use $\tau$ to denote a trajectory with planing horizon $H$, i.e., $\tau = \{(s_h, a_h)\}_{h=1}^H$.

**Trajectory Reward:** we assume the expected reward of each trajectory $\tau$ is a general function $r(\tau)$ which maps any trajectory to a value in $[0, H]$ (Zhang et al., 2024a). Without loss of generality,

we scale the average per-step reward into $[0, 1]$, and the return of the trajectory does not necessarily need to be the sum of per-step rewards. For any given policy $\pi$, we can formulate the initial value function $V_1^\pi(s)$ as the expected reward of trajectories starting from $s$ with policy $\pi$:

$$V_1^\pi(s) = \mathbb{E}_\pi \left[ r(\tau) | \, s_1 = s \right] = \mathbb{E} \left[ r(\tau) | \, s_1 = s, \{a_1, \cdots, a_H\} \sim \pi \right].$$

The goal of the RL problem is to find a policy to maximize $V(\pi) = \mathbb{E}_{s \sim \boldsymbol{\mu}_0}[V_1^\pi(s)]$.

**Policy Parameterization:** to address the large state space $\mathbb{S}$ and action space $\mathbb{A}$ in most RL problems, we assume access to a parameterized policy network $\mathsf{N}_{\boldsymbol{\theta}} : \mathbb{S} \times [H] \to \mathcal{P}(\mathbb{A})$ which takes a state and a decision-making step as input, and then outputs the probability distribution of the next action. Here $\boldsymbol{\theta} \in \mathbb{R}^d$ is the policy network parameter vector. Each parameter $\boldsymbol{\theta}$ through the policy network will induce a policy which we slightly abuse the notation and use $\pi_{\boldsymbol{\theta}}$ to denote.

**Human Feedback:** The agent has access to human feedback that provides a preference based on the rewards of two trajectories. In each episode, the agent can choose two trajectories $\tau_0$ and $\tau_1$ to query human preference: one-bit feedback $o \in \{0, 1\}$. We assume the preference $o$ is generated according to a known preference model where the probability of the outcome between two trajectories is determined by the difference in their rewards. Since the difference is not necessarily a value inside the unit interval, the preference model uses a *link* function $\sigma : \mathbb{R} \to [0, 1]$ to map these differences of rewards to actual probabilities, i.e.,

$$\mathbb{P}(\tau_1 \succ \tau_0) = \sigma(r(\tau_1) - r(\tau_0)), \tag{2}$$

where $\tau_1 \succ \tau_0$ is the event that the human feedback prefers $\tau_1$ over $\tau_0$. The human feedback $o$, therefore, is a random sample from a Bernoulli distribution with $\mathbb{P}(o = 1) = \mathbb{P}(\tau_1 \succ \tau_0)$. The notion of link function comes from the dueling bandit literature to model preference with latent utility between arms, e.g. see Bengs et al. (2021, Section 3.2). This preference model has been used in dueling bandits (Bengs et al., 2021; Yue & Joachims, 2009; Kumagai, 2017; Ailon et al., 2014) as well as RLHF (Wang et al., 2023). One can see that one specific link function $\sigma$ will define a specific preference model (Azari et al., 2012), i.e., replacing $\sigma(\cdot)$ with a logistic function, we recover the Bradley-Terry model (Bradley & Terry, 1952), which is commonly used in RLHF for both practical (Christiano et al., 2017; Ouyang et al., 2022; Rafailov et al., 2023) and theoretical works (Du et al., 2024; Zhan et al., 2024a;b). On the other hand, let $\sigma(\cdot)$ be the cumulative distribution function (CDF) of standard normal distribution, we obtain the *Probit* model (Thurstone, 1927) which has wide application in the study of social choice theory, economy, and psychology (Train, 2009; Greene, 2000). A detailed discussion on other preference models is provided in the appendix. The following assumption on the link function is standard in both dueling bandits (Bengs et al., 2021) and preference-based RL (Wang et al., 2023). One can easily verify that both the Bradley-Terry and Probit models satisfy this assumption.

**Assumption 1** *The link function $\sigma(\cdot)$ in the preference model in equation 2 is bounded within $[0, 1]$ and strictly monotonically increasing on $[-H, H]$ with $\sigma(0) = 1/2$.*

**Problem and Notations:** Our goal is to find parameter $\boldsymbol{\theta}$ that maximizes the value function, i.e., $\max_{\boldsymbol{\theta} \in \mathbb{R}^d} V(\pi_{\boldsymbol{\theta}})$. For a scalar $a$, we use $\mathrm{trim}[a|\Delta]$ to represent $\min\{\max\{a, \Delta\}, 1 - \Delta\}$. For a vector $\boldsymbol{v}$, $\mathrm{trim}[\boldsymbol{v}|\Delta]$ represents the vector after applying the $\mathrm{trim}$ operator to each element respectively. Let $\boldsymbol{e}_i \in \mathbb{R}^d$ represent the unit vector with all zero elements but 1 on the $i$-th coordinate.

## 3 Zeroth-Order Policy Gradient Algorithms for RLHF

In this section, we propose two RLHF algorithms without reward inference, motivated by the relation between preference and zeroth-order gradient. We first present ZPG, a stochastic gradient descent algorithm, and then ZBCPG, a stochastic block-coordinate descent algorithm.

### 3.1 ZPG: Zeroth-Order Policy Gradient from Human Feedback

Our first algorithm ZPG consists of the following five steps in each policy gradient iteration:

- From the current policy $\pi_{\boldsymbol{\theta}_t}$, it first obtained a perturbed policy $\pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}$ (line 2-3).
- Sample $N$ pairs of trajectories under the two policies $\pi_{\boldsymbol{\theta}_t}$ and $\pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}$ (lines 5-6).

**Algorithm 1:** Zeroth-Order Policy Gradient from Human Feedback

**Parameters:** initial parameter $\boldsymbol{\theta}_0$, learning rate $\alpha$, trim size $\Delta$, perturbation distance $\mu$.

1 **for** $t = 1 : T$ **do**
2 sample $\boldsymbol{v}_t$ uniformly from a unit sphere $\mathbb{S}^{d-1} = \left\{ \boldsymbol{v} \in \mathbb{R}^d \middle| \|\boldsymbol{v}\|_2 = 1 \right\}$;
3 obtain a perturbed policy $\pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}$;
4 **for** $n = 1 : N$ **do**
5 sample trajectory $\tau_{n,0} \sim \pi_{\boldsymbol{\theta}_t}$;
6 sample trajectory $\tau_{n,1} \sim \pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}$;
7 query $M$ human evaluators with $(\tau_{n,1}, \tau_{n,0})$ and obtain feedback $[o_{n,1}, \cdots, o_{n,M}]$;
8 estimate preference probability $p_{t,n} = \mathrm{trim}\left[ \sum_{m=1}^{M} \frac{o_{n,m}}{M} \middle| \Delta \right]$;
9 estimate the policy gradient: $\hat{\boldsymbol{g}}_t = \frac{d}{\mu} \frac{\sum_{n=1}^{N} \sigma^{-1}(p_{t,n})}{N} \boldsymbol{v}_t$;
10 update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \hat{\boldsymbol{g}}_t$;

- For each trajectory pair, say $(\tau_{n,1}, \tau_{n,0})$, obtain $M$ independent human preferences (line 7) and estimate the probability that $\tau_{n,1}$ is preferred over $\tau_{n,0}$ (line 8), denoted by $p_{t,n}$.
- Use the $N$ estimates $\{p_{t,n}\}_n$ and link function $\sigma(\cdot)$ to estimate the gradient $\hat{\boldsymbol{g}}_t$ (line 9).
- Update the current policy to a new policy $\boldsymbol{\theta}_{t+1}$ using gradient ascent (line 10).

The pseudo-code is presented in Alg. 1. As we mentioned earlier, our approach uses human feedback in a way different from both the classic reward inference in RLHF and DPO. The reward inference uses human preferences to recover the *global* reward function and DPO relates the human preference generation mechanism to the optimal policy. We view the human feedback as local information that points to the direction of a more preferred policy, i.e., the policy gradient direction. Some online RLHF algorithms such as online DPO also exploit similar local estimation viewpoints, i.e., using new trajectories of current policy to locally improve the estimation of DPO loss and then proceed. However, we want to emphasize that the relation between the DPO loss and the optimal policy is still global, and it is limited to deterministic MDPs, which shows the novelty of our approach. The algorithm we propose has two key components: (i) a value function difference estimator from human preference and (ii) a policy gradient estimator from value function difference.

**Policy Gradient Approximation:** At each iteration of the algorithm, it first samples a $d$ dimensional vector $\boldsymbol{v}_t$ from a unit sphere and then perturbs the policy network parameter $\boldsymbol{\theta}_t$ along the direction of this sampled vector. Then, it uses the inner for loop to construct an estimation of the value function difference between the original policy and the perturbed policy, i.e., $V(\pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}) - V(\pi_{\boldsymbol{\theta}_t})$. We then plug it into the zeroth-order stochastic gradient descent (SGD) algorithm proposed in (Ghadimi & Lan, 2013) to construct a zeroth-order approximation to the policy gradient, i.e.,

$$\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}_t}) \approx \mathbb{E}_{\boldsymbol{v}_t} \left[ \frac{d}{\mu} \left( V(\pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}) - V(\pi_{\boldsymbol{\theta}_t}) \right) \boldsymbol{v}_t \right].$$

We remark that the random vector $\boldsymbol{v}_t$ for each iteration can also be drawn from a normal distribution (Nesterov & Spokoiny, 2017), but the unit sphere is more numerically stable.

**Value Function Inference:** The inner for loop of Alg. 1 aims to estimate the value function difference between the perturbed policy $\pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}$ and current policy $\pi_{\boldsymbol{\theta}_t}$. The algorithm samples multiple trajectory pairs with both policies and for each pair, it queries humans multiple times to obtain pairwise preferences $[o_{n,1}, \cdots, o_{n,M}]$. It then uses the preferences to construct a robust estimator $p_{t,n}$ to approximate the probability of comparison $\mathbb{P}(\tau_{n,1} \succ \tau_{n,0})$, which is further converted to an estimate of the value function difference based on the preference model in equation 2 as follows:

$$V(\pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}) - V(\pi_{\boldsymbol{\theta}_t}) \approx \frac{1}{N} \sum_{n=1}^{N} \sigma^{-1}(p_{t,n}). \tag{3}$$

Querying humans multiple times ensures an accurate estimation of the reward gap between two trajectories. The reward gap of two trajectories is a random sample of the value function difference, so we sample multiple trajectories to ensure the average trajectory reward gap converges to the

**Algorithm 2:** Zeroth-Order Block-Coordinate Policy Gradient from Human Feedback

**Parameters:** initial parameter $\boldsymbol{\theta}_0$, learning rate $\alpha$, trim size $\Delta$, perturbation distance $\mu$, coordinate batch size $K$.

**for** $t = 1 : T$ **do**
- sample a set of $K$ coordinates $\boldsymbol{i}_t = [i_{t,1}, i_{t,2}, \cdots, i_{t,K}]$ from $\{1, 2, \cdots, d\}$;
- sample a set $\boldsymbol{\lambda}_t = [\lambda_{t,1}, \lambda_{t,2}, \cdots, \lambda_{t,K}]$ where each $\lambda_{t,j}$ is uniformly sampled from $\{-1, 1\}$;
- construct the perturbation vector: $\boldsymbol{v}_t = \frac{1}{\sqrt{K}} \sum_{j=1}^{K} \lambda_{t,j} \boldsymbol{e}_{i_{t,j}}$;
- **for** $n = 1 : N$ **do**
  - sample trajectory $\tau_{n,0} \sim \pi_{\boldsymbol{\theta}_t}$;
  - sample trajectory $\tau_{n,1} \sim \pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}$;
  - query $M$ human evaluators with $(\tau_{n,1}, \tau_{n,0})$ and obtain feedback $[o_{n,1}, \cdots, o_{n,M}]$;
  - estimate preference probability $p_{t,n} = \mathrm{trim}\left[\sum_{m=1}^{M} \frac{o_{n,m}}{M} \middle| \Delta\right]$;
- estimate the policy gradient: $\hat{\boldsymbol{g}}_t = \frac{d}{\mu} \frac{\sum_{n=1}^{N} \sigma^{-1}(p_{t,n})}{N} \boldsymbol{v}_t$;
- update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \hat{\boldsymbol{g}}_t$;

value function difference. To ensure finite variance after applying $\sigma^{-1}(\cdot)$ function, we trim $p_{t,n}$ with a small constant which can be set to $\min\{\sigma(-H), 1 - \sigma(H)\}$ in this case.

### 3.2 ZBCPG: Zeroth-Order Block-Coordinate Policy Gradient from Human Feedback

In high-dimensional optimization problems, it is usually memory and operation inefficient to approximate the full gradient and update all the parameters in the policy network at the same iteration step (Malladi et al., 2023; Zhang et al., 2024b), which motivates parameter efficient fine-tuning (PEFT). The stochastic (block) coordinate descent approach naturally arises because of its ease of implementation, low memory requirements, and adaptability to distributed settings (Nesterov, 2012; Lu & Xiao, 2015). The same advantage also applies to RLHF when the number of parameters in the policy network is too large. Therefore, we propose ZBCPG, a block coordinate version of ZPG, which is summarized in Alg. 2. The key difference between ZBCPG and ZPG is the choice of the perturbation direction, where we use Rademacher noise instead of the normal perturbation in ZPG.

**Zeroth-Order Block Coordinate Gradient Approximation:** Instead of sampling from a sphere, which perturbs all parameters of the policy network, ZBCPG separates the sampling procedure into two simple parts: first sample a minibatch of coordinates and then sample a zero-centered Bernoulli random variable for each coordinate, which still results in a valid gradient estimator.

$$\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}_t}) \approx \mathbb{E}_{\boldsymbol{i}_t, \boldsymbol{\lambda}_t} \left[ \frac{d}{K\mu} \left( V(\pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}) - V(\pi_{\boldsymbol{\theta}_t}) \right) \boldsymbol{v}_t \right].$$

The block-coordinate approach allows us to (i) perturb a subset of parameters at each iteration, e.g., a specific layer of the policy network for fine-tuning, and (ii) have a parallel implementation where we have multiple gradient estimators $\hat{\boldsymbol{g}}_t$ when updating the policy. We will later show that both algorithms have similar provable convergence guarantees but the analysis of ZBCPG is more challenging due to the perturbation mechanism.

## 4 Theoretical Analysis: Rate of Convergence

In this section, we provide theoretical performance guarantees for both ZPG and ZBCPG. We first provide technical assumptions on the preference generation model, the policy network, and the value function landscape, which is necessary for deriving theoretical insights.

### 4.1 ASSUMPTIONS

To infer the local reward difference from human preference probability through link function $\sigma(\cdot)$, we impose the following assumption which is satisfied by the Bradley-Terry model. A slightly weaker assumption is also adopted by Wang et al. (2023) and justified as a minimal requirement to learn the optimal policy. We use $\Delta = \min\{\sigma(-H), 1-\sigma(H)\}$ as the trim constant.

**Assumption 2** *The inverse link function $\sigma^{-1}(\cdot)$ is L-Lipchitz continuous on $[\Delta, 1-\Delta]$.*

We further require the landscape of the value function and the policy network to be "regular", and impose the following assumption, which is a standard assumption used in nonconvex optimization literature (Liu et al., 2019; Bernstein et al., 2018; Reddi et al., 2018).

**Assumption 3** *The value function $V(\pi_{\boldsymbol{\theta}})$ for the policy network parameters $\boldsymbol{\theta}$ is L-smooth on $\mathbb{R}^d$.*

Since a trajectory reward is bounded in $[0, H]$, $V(\pi_{\boldsymbol{\theta}^*}) < \infty$, where $\boldsymbol{\theta}^*$ is the global optimal solution. For simplicity, we assume $L$ is the constant upper bound for both assumptions.

### 4.2 CONVERGENCE RATE AND SAMPLE COMPLEXITY

In this section, we present the theoretical guarantees for both ZPG and ZBCPG under all three assumptions mentioned in previous sections. We aim to learn an $\epsilon$-stationary policy $\pi_{\boldsymbol{\theta}}$ with $\|\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}})\|_2^2 \leq \epsilon$, and study the convergence rate and sample complexity.

**Theorem 1** *Choose the perturbation distance to be $\mu^2 = \Theta\left(\max\left\{\frac{1}{\sqrt{M}}, \frac{H}{\sqrt{dN}}\right\}\right)$ and learning rate to be $\alpha = \Theta(d^{-1})$. If $M = \Omega(H^2)$ and we randomly pick $\boldsymbol{\theta}_R$ uniformly from the trajectory $\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{T-1}\}$, then the convergence rate of ZPG satisfies:*

$$\mathbb{E}\left[\|\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}_R})\|_2^2\right] = \mathcal{O}\left(\frac{Hd}{T} + \frac{d^2\sqrt{\log M}}{\sqrt{M}} + \frac{Hd\sqrt{d}}{\sqrt{N}}\right).$$

**Theorem 2** *Choose the perturbation distance to be $\mu^2 = \Theta\left(\max\left\{\frac{1}{\sqrt{M}}, \frac{H}{\sqrt{dN}}\right\}\right)$ and learning rate to be $\alpha = \Theta(d^{-1})$. If $M = \Omega(H^2)$ and we randomly pick $\boldsymbol{\theta}_R$ uniformly from the trajectory $\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{T-1}\}$, then the convergence rate of ZBCPG satisfies:*

$$\mathbb{E}\left[\|\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}_R})\|_2^2\right] = \mathcal{O}\left(\frac{Hd}{T} + \frac{d^2\sqrt{\log M}}{\sqrt{M}} + \frac{Hd\sqrt{d}}{\sqrt{N}}\right).$$

The complete proof of both theorems is presented in the appendix. Here we first provide insights into the choice of hyper-parameters and convergence rate results in both theorems, and then we discuss the challenges and technical novelties of our proof.

**Insights behind the Convergence Rate:** Both ZPG and ZBCPG have the same rate of convergence which consists of three components: the zeroth-order gradient descent rate, the preference estimation error, and the value function approximation error

$$\underbrace{\frac{Hd}{T}}_{\text{Zeroth-Order Gradient Descent}} + \underbrace{\frac{d^2\sqrt{\log M}}{\sqrt{M}}}_{\text{Preference Estimation}} + \underbrace{\frac{Hd\sqrt{d}}{\sqrt{N}}}_{\text{Value Function Approximation}}.$$

The second represents the error that occurs when using multiple human preferences $[o_{n,1}, \cdots, o_{n,M}]$ to approximate the population-level human preference probability for given two trajectories, i.e., $\mathbb{P}(\tau_{n,1} \succ \tau_{n,0})$. This error will further result in a bias term after being plugged into the inverse link function $\sigma^{-1}(\cdot)$ to construct an estimation of the value function difference. The third term comes from the variance of using multiple trajectory rewards to approximate the value function of a policy. The first term represents the error resulting from zeroth-order stochastic gradient descent or blocked coordinate descent, which matches the state-of-the-art analysis result $\mathcal{O}(d/T)$ for

non-convex smooth function optimization (Nesterov & Spokoiny, 2017). Even though the final convergence rates are the same and we both use constant learning rates, how we choose the perturbation distance to obtain the rate differs from (Ghadimi & Lan, 2013). Specifically, they chose a small perturbation distance with $\mu^2 = \mathcal{O}(d/T)$ to make sure the zeroth order approximation error is of lower order. However, this choice will not work for us, because our gradient estimate is biased due to the non-linear nature of the link function in preference estimation. If we choose the perturbation distance $\mu$ to be too small, the preference estimation error will be amplified by $d/\mu$ due to the formula of zeroth-order approximation $\hat{\boldsymbol{g}}_t$. This phenomenon adds complication to our theoretical analysis. Our method is to use a moderate perturbation distance $\mu$. Moreover, this moderate perturbation distance also balances the preference estimation and the value function approximation errors.

Based on the theorems, we have the following corollary that characterizes the sample complexity.

**Corollary 1** *To learn an $\epsilon$-stationary policy, the required number of human preference queries of ZPG and ZBCPG with proper hyper-parameters satisfies*

$$TMN = \mathcal{O}\left(\frac{d^8 H^3}{\epsilon^5} \log\left(\frac{d}{\epsilon}\right)\right).$$

### 4.3 Technical Challenges and Proof Novelties

In this section, we first overview the proof of zeroth-order stochastic gradient descent used in (Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017; Gao et al., 2018; Liu et al., 2019) from a Lyapunov drift optimization perspective. We then show the major technical difficulties in applying such a framework to analyze both ZPG and ZBCPG, i.e., the gradient estimator is biased due to stochastic human preference. Then, we demonstrate our novel analysis techniques to resolve them.

**Classic Proof of Zeroth-Order Optimization:** to illustrate the procedure of the analysis of zeroth order gradient estimate, we suppose we can query $V(\pi_{\boldsymbol{\theta}})$ for any $\boldsymbol{\theta}$. This procedure makes use of the randomized smoothing function $V_\mu(\boldsymbol{\theta})$ (Ghadimi & Lan, 2013; Gao et al., 2018) as

$$V_\mu(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{v}'}\left[V(\pi_{\boldsymbol{\theta}+\mu\boldsymbol{v}'})\right],$$

where the random vector $\boldsymbol{v}'$ follows a uniform distribution over the unit Euclidean ball. It is shown in (Gao et al., 2018) that the zeroth-order gradient estimator used in ZPG, constructed from sampling $\boldsymbol{v}_t$ uniformly over a sphere, is an unbiased estimator of the smoothing function gradient, i.e.,

$$\nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{v}}\left[\frac{d}{\mu}\left(V(\pi_{\boldsymbol{\theta}+\mu\boldsymbol{v}}) - V(\pi_{\boldsymbol{\theta}_t})\right)\boldsymbol{v}\right],$$

where $\boldsymbol{v}$ is sampled from a unit sphere. Clearly, the gradient of the smoothing function is not equal to the original value function, as well as the function itself, but it can be shown that they are close as long as $\mu$ is small (Liu et al., 2018b):

$$|V_\mu(\pi_{\boldsymbol{\theta}}) - V(\pi_{\boldsymbol{\theta}})| = \mathcal{O}\left(\mu^2\right); \quad \|\nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}}) - \nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}})\|_2 = \mathcal{O}\left(\mu d\right). \tag{4}$$

The standard proof uses the randomized smoothing function $V_\mu(\pi_{\boldsymbol{\theta}})$ as the Lyapunov function and then bounds the drift given the stochastic gradient descent update rule when $\alpha = \Theta(1/d)$. Neglecting problem-independent constants, we have:

$$\begin{aligned}
&V_\mu(\pi_{\boldsymbol{\theta}_t}) - V_\mu(\pi_{\boldsymbol{\theta}_{t+1}}) \\
&\quad \leq -\alpha \underbrace{\|\nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}_t})\|_2^2}_{\text{Drift}} + \alpha \underbrace{\langle \nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}_t}), \nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}_t}) - \hat{\boldsymbol{g}}_t\rangle}_{\text{1st Order: GradBias}} + \alpha^2 \underbrace{\|\hat{\boldsymbol{g}}_t - \nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}_t})\|_2^2}_{\text{2nd Order: GradVar}\approx\mu^2 d^2}.
\end{aligned}$$

Note the gradient estimator $\hat{\boldsymbol{g}}_t$ is unbiased and bounded, and the gradient of $V_\mu(\pi_{\boldsymbol{\theta}})$ is close to $V(\pi_{\boldsymbol{\theta}})$, taking a conditional expectation over the filtration before time $t$ will result in:

$$\begin{aligned}
&\mathbb{E}[V_\mu(\pi_{\boldsymbol{\theta}_t})|\mathcal{F}_t] - \mathbb{E}[V_\mu(\pi_{\boldsymbol{\theta}_{t+1}})|\mathcal{F}_t] \\
&\quad \leq -\alpha\|\nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}_t})\|_2^2 + \alpha\langle \nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}_t}), \mathbb{E}[\nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}_t}) - \hat{\boldsymbol{g}}_t]\rangle + \alpha^2\mu^2 d^2 \\
&\quad \leq -\alpha\|\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}_t})\|_2^2 + \alpha\mu^2 d^2 + \alpha^2\mu^2 d^2,
\end{aligned}$$

where the last step uses equation 4 and the fact that the gradient is unbiased. Let us choose a small learning rate $\alpha = \Theta(1/d)$ and take an expectation with a telescoping sum to obtain:

$$\underbrace{\frac{\mathbb{E}[V_\mu(\pi_{\boldsymbol{\theta}_0})] - \mathbb{E}[V_\mu(\pi_{\boldsymbol{\theta}_T})]}{T}}_{\mathcal{O}(H/T)} \lesssim -\alpha\, \underbrace{\mathbb{E}\left[\frac{\sum_{t=1}^{T} \|\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}_t})\|_2^2}{T}\right]}_{\mathsf{Target}} + \alpha\mu^2 d^2.$$

A little manipulation will lead to the following bound, which can be made small when $\mu \approx \sqrt{1/dT}$.

$$\mathsf{Target} = \mathcal{O}\left(\frac{H}{T\alpha} + \mu^2 d^2\right) = \mathcal{O}\left(\frac{Hd}{T} + \mu^2 d^2\right) = \mathcal{O}\left(\frac{Hd}{T}\right).$$

**Amplified Gradient Biases for ZPG:** If we directly apply the steps above to ZPG, we immediately run into the issue that our gradient estimator $\hat{\boldsymbol{g}}_t$ in expectation is biased even compared to smoothing function gradient due to preference estimation. Moreover, the second-order gradient variance will be larger since we used trajectory reward to estimate the value function. From concentration, we will obtain an error bound of using preference to estimate the value function difference as:

$$\left|\mathbb{E}\left[\sigma^{-1}(p_{t,n})\right] - (V(\pi_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}) - V(\pi_{\boldsymbol{\theta}_t}))\right| \leq \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{M}}\right),$$

where $\tilde{\mathcal{O}}$ hides logarithmic terms. This bias term will be amplified by $d/\mu$ and then added to the gradient estimation bias in the first-order drift term if plugged into the analysis:

$$\mathbb{E}[V_\mu(\pi_{\boldsymbol{\theta}_t})|\mathcal{F}_t] - \mathbb{E}[V_\mu(\pi_{\boldsymbol{\theta}_{t+1}})|\mathcal{F}_t] \leq -\underbrace{\alpha\|\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}_t})\|_2^2 + \alpha\mu^2 d^2}_{\text{Same Drift as Before}} + \alpha\, \underbrace{\frac{d\|\nabla_{\boldsymbol{\theta}} V_\mu(\pi_{\boldsymbol{\theta}_t})\|_2}{\mu\sqrt{M}}}_{\text{Additional Bias}}.$$

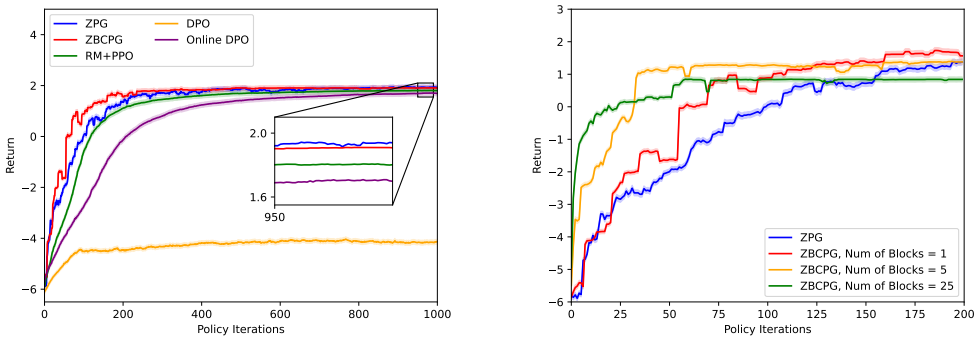
Using the same perturbation distance as before, the additional bias will lead to an $\tilde{\mathcal{O}}(\sqrt{T/M})$ term in the final bound, which is small only when $M$ is much larger than $T$ and is much looser compared with ours. For example, letting $M = T^2$, the above bound is $\tilde{\mathcal{O}}(1/\sqrt{T})$ while ours is $\tilde{\mathcal{O}}(1/T)$.

Our approach to avoid this term in the final result is to make use of the gradient value $\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}_t})$ in the first-order term to cancel out the additional bias on certain occasions. Specifically, we divide the trajectory of $\boldsymbol{\theta}_t$ into two sets, one with a relatively large gradient and one with a relatively small gradient. For $\boldsymbol{\theta}_t$ with a large gradient, we use a part of the negative drift to cancel out the additional bias, since the negative drift is the square of the gradient $\nabla_{\boldsymbol{\theta}} V(\pi_{\boldsymbol{\theta}_t})$ which is even larger. For $\boldsymbol{\theta}_t$ with a small gradient, we know the bias term will be small and thus can provide a refined drift bound. Combining this analysis with a slightly larger perturbation distance $\mu$, we will be able to balance the additional bias with gradient variance to cancel out the $\tilde{\mathcal{O}}(\sqrt{T/M})$ term and obtain the final result.

**Implicit Smoothing Function for ZBCPG:** Due to the choice of blocked perturbation vector sampling procedure, it is difficult to obtain the exact analytical expression of the smoothing function $V_\mu(\pi_{\boldsymbol{\theta}})$ whose gradient is the expectation of gradient estimation $\hat{\boldsymbol{g}}_t$ for ZBCPG. This prohibits us from continuing to use $V_\mu(\pi_{\boldsymbol{\theta}})$ as the Lyapunov function, as it is hard to analyze the gradient bias and the variance without an explicit target format. However, if we rethink the reason for introducing the smoothed function in zeroth-order optimization, we hope the gradient of the smoothed function will be unbiased to cancel out the first-order positive drift. However, this is already not true in the analysis of ZPG since we have gradient estimation bias from human feedback, but it is small enough on average to be controlled. If the gradient difference between the smoothed function $V_\mu(\pi_{\boldsymbol{\theta}})$ and the vanilla value function $V(\pi_{\boldsymbol{\theta}})$ is smaller than this additional bias, then we can use the original value function $V(\pi_{\boldsymbol{\theta}})$ as the Lyapunov function at the cost of an additional bias besides preference estimation. Fortunately, this can be achieved through a carefully chosen perturbation distance $\mu$ to balance these two types of errors.

## 5 EXPERIMENTS

In this section, we study the empirical performance of ZPG and ZBCPG in a stochastic GridWorld environment, where details can be found in Appendix. C. In our environment, the actions chosen




(a) performance (b) parallelization

Figure 2: GridWorld with Bradley-Terry Feedback: (a) the trajectory return of ZPG, ZBCPG, and RLHF baselines, and (b) the return of ZBCPG with different parallelization levels. All results are averaged over $10^5$ repetitions of policy evaluation and shaded areas indicate confidence intervals.

| Algorithm | ZPG (Ours) | ZBCPG (Ours) | RM+PPO | DPO | Online DPO |
|---|---|---|---|---|---|
| Return | $\mathbf{1.94} \pm 0.09$ | $\mathbf{1.91} \pm 0.09$ | $1.80 \pm 0.09$ | $-4.13 \pm 0.09$ | $1.71 \pm 0.09$ |

Table 1: Last Iterate Policy Average Return with Bradley-Terry Feedback.

by the agent may be reversed with certain probabilities due to imperfect control or environmental disturbances such as wind or turbulence. The agent can query human evaluators for preference over two trajectories and the human feedback is generated from the Bradley-Terry model with a logistic link function. We consider three baselines: (1) RM+PPO (Ouyang et al., 2022), (2) DPO for token-level MDP (Rafailov et al., 2024b), and (3) Online DPO for token-level MDP (Dong et al., 2024; Guo et al., 2024). All algorithms collect $N = 1000$ trajectory pairs between policy updates and $M = 1000$ human experts evaluate each pair. The trajectory return for each iteration is compared in Fig. 2(a) and the return of the final policy is reported in Tab. 1. Both ZPG and ZBCPG perform better than the three baselines in both convergence rates and the quality of the last iterate policy. Compared to PPO, our algorithms converge to a better policy partially since the reward model is inaccurate and the agent is not able to learn the optimal policy from it. It is also observed that vanilla DPO has a much worse performance than our proposed algorithms. This may result from two reasons: first, the DPO loss is valid only in deterministic MDP, and second, DPO is constrained to the neighborhood of the sub-optimal reference policy. The online DPO algorithm improves over vanilla DPO but still has inferior performance due to the inherent model error of the DPO loss. This also shows the fundamental difference between stochastic and deterministic MDPs, and the need to design RLHF algorithms for general RL problems. Moreover, online DPO converges much slower, partly because the DPO loss landscape becomes flat and hard to optimize when the weight of the KL constraint is small for better exploration. In Fig. 2(b), we also compare ZPG to distributed implementations of ZBCPG, where the panel of human evaluators is also separated into small groups for parallelization. It is shown that as the number of blocks increases, ZBCPG converges faster to a stationary policy. However, the number of human queries per pair of trajectories in each parallelization also decreases, which introduces a larger gradient bias and leads to a sub-optimal policy. Therefore, the trade-off between computation parallelization and accuracy should be taken carefully.

## 6 Conclusion

In this paper, we proposed two RLHF algorithms without reward inference based on a zeroth-order policy gradient called ZPG and ZBCPG. Both algorithms are shown to have a provable polynomial sample complexity to learn a stationary policy under mild conditions and exhibit nice empirical performances in environments with stochastic transitions, outperforming popular RLHF baselines.

## ACKNOWLEDGMENTS

The work of Qining Zhang and Lei Ying is supported in part by NSF under grants 2112471, 2134081, 2207548, 2240981, and 2331780.

## REFERENCES

Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 856–864, Bejing, China, 22–24 Jun 2014. PMLR.

Riad Akrour, Marc Schoenauer, and Michele Sebag. Preference-based policy learning. In *Machine Learning and Knowledge Discovery in Databases*, pp. 12–27, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23780-5.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2021.103500.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4447–4455. PMLR, 02–04 May 2024.

Hossein Azari, David Parks, and Lirong Xia. Random utility theory for social choice. *Advances in Neural Information Processing Systems*, 25, 2012.

Viktor Bengs, Robert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hullermeier. Preference-based online learning with dueling bandits: A survey. *Journal of Machine Learning Research*, 22(7): 1–108, 2021.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 560–569. PMLR, 10–15 Jul 2018.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444, 14643510.

Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, Weiwei Cheng, and Eyke Hüllermeier. Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm. *Machine learning*, 97:327–351, 2014.

Hanqin Cai, Yuchen Lou, Daniel Mckenzie, and Wotao Yin. A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1193–1203. PMLR, 18–24 Jul 2021.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando Ramirez, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *Trans. Machine Learning Research (TMLR)*, 2023.

Souradip Chakraborty, Amrit Bedi, Alec Koppel, Huazheng Wang, Dinesh Manocha, Mengdi Wang, and Furong Huang. PARL: A unified framework for policy alignment in reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

Xiaoyu Chen, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3773–3793. PMLR, 17–23 Jul 2022.

Paul Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth Stanley, and Jeff Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in neural information processing systems*, 31, 2018.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

Yihan Du, Anna Winnicki, Gal Dalal, Shie Mannor, and R. Srikant. Exploration-driven policy optimization in RLHF: Theoretical insights on efficient data utilization. In *Forty-first International Conference on Machine Learning*, 2024.

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

Xiang Gao, Bo Jiang, and Shuzhong Zhang. On the information-adaptive variants of the admm: an iteration complexity perspective. *Journal of Scientific Computing*, 76:327–363, 2018.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

WH Greene. Modeling ordered choices: A primer, 2010.

William H Greene. Econometric analysis 4th edition. *International edition, New Jersey: Prentice Hall*, pp. 201–215, 2000.

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

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In Paolo Frasconi, Niels Landwehr, Giuseppe Manco, and Jilles Vreeken (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 795–811, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46128-1.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

Chinmaya Kausik, Mirco Mutti, Aldo Pacchiano, and Ambuj Tewari. A framework for partially observed reward-states in rlhf. *arXiv preprint arXiv:2402.03282*, 2024.

Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2022. doi: 10.1109/TITS.2021.3054625.

Bradley Knox and Peter Stone. Tamer: Training an agent manually via evaluative reinforcement. In *2008 7th IEEE International Conference on Development and Learning*, pp. 292–297, 2008. doi: 10.1109/DEVLRN.2008.4640845.

Pushmeet Kohli, Mahyar Salek, and Greg Stoddard. A fast bandit algorithm for recommendation to users with heterogenous tastes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27 (1):1135–1141, Jun. 2013. doi: 10.1609/aaai.v27i1.8463.

Dingwen Kong and Lin Yang. Provably feedback-efficient reinforcement learning via active reward learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11063–11078. Curran Associates, Inc., 2022.

Wataru Kumagai. Regret analysis for continuous dueling bandit. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Zihao Li, Zhuoran Yang, and Mengdi Wang. Reinforcement learning with human feedback: Learning dynamic choices via pessimism. *arXiv preprint arXiv:2305.18438*, 2023.

Sijia Liu, Jie Chen, Pin-Yu Chen, and Alfred Hero. Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 288–297. PMLR, 09–11 Apr 2018a.

Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018b.

Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong. signSGD via zeroth-order oracle. In *International Conference on Learning Representations*, 2019.

Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152:615–642, 2015.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53038–53075. Curran Associates, Inc., 2023.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Yu. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. doi: 10.1137/100802001.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17(2):527–566, apr 2017. ISSN 1615-3375. doi: 10.1007/s10208-015-9296-2.

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

Ellen Novoseller, Yibing Wei, Yanan Sui, Yisong Yue, and Joel Burdick. Dueling posterior sampling for preference-based reinforcement learning. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pp. 1029–1038. PMLR, 03–06 Aug 2020.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. *arXiv preprint arXiv:2406.02900*, 2024a.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From $r$ to $q^*$: Your language model is secretly a q-function. In *First Conference on Language Modeling*, 2024b.

Ingo Rechenberg. Evolutionsstrategie. *Optimierung technischer Systeme nach Prinzipien derbiologischen Evolution*, 1973.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Aadirupa Saha, Aldo Pacchiano, and Jonathan Lee. Dueling rl: Reinforcement learning with trajectory preferences. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 6263–6289. PMLR, 25–27 Apr 2023.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Marc Schoenauer, Riad Akrour, Michele Sebag, and Jean-Christophe Souplet. Programming by feedback. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1503–1511, Bejing, China, 22–24 Jun 2014. PMLR.

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

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

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Remi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Avila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. In *Forty-first International Conference on Machine Learning*, 2024a.

Zhiwei Tang, Dmitry Rybin, and Tsung-Hui Chang. Zeroth-order optimization meets human feedback: Provable learning via ranking oracles. In *The Twelfth International Conference on Learning Representations*, 2024b.

LL Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273, 1927.

Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

Yuanhao Wang, Qinghua Liu, and Chi Jin. Is rlhf more difficult than standard rl? *arXiv preprint arXiv:2306.14111*, 2023.

Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11485.

Aaron Wilson, Alan Fern, and Prasad Tadepalli. A bayesian approach for policy learning from trajectory preference queries. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Christian Wirth, Johannes Fürnkranz, and Gerhard Neumann. Model-free preference-based reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016. doi: 10.1609/aaai.v30i1.10269.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Jeff Wu, Long Ouyang, Daniel Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021.

Runzhe Wu and Wen Sun. Making RL with preference-based feedback efficient via randomization. In *The Twelfth International Conference on Learning Representations*, 2024.

Tengyang Xie, Dylan J Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Awadallah, and Alexander Rakhlin. Exploratory preference optimization: Harnessing implicit q*-approximation for sample-efficient rlhf. *arXiv preprint arXiv:2405.21046*, 2024.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. In *Forty-first International Conference on Machine Learning*, 2024.

Yichong Xu, Ruosong Wang, Lin Yang, Aarti Singh, and Artur Dubrawski. Preference-based reinforcement learning with finite-time guarantees. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18784–18794. Curran Associates, Inc., 2020.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 1201–1208, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553527.

Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 2025–2034, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939878.

Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason D. Lee, and Wen Sun. Provable offline preference-based reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024a.

Wenhao Zhan, Masatoshi Uehara, Wen Sun, and Jason D. Lee. Provable reward-agnostic preference-based reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024b.

Qining Zhang, Honghao Wei, and Lei Ying. Reinforcement learning from human feedback without reward inference: Model-free algorithm and instance-dependent analysis. *Reinforcement Learning Journal*, 3:1236–1251, 2024a.

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D. Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark. In *Forty-first International Conference on Machine Learning*, 2024b.

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

Banghua Zhu, Michael Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 43037–43067. PMLR, 23–29 Jul 2023.

Banghua Zhu, Michael Jordan, and Jiantao Jiao. Iterative data smoothing: Mitigating reward overfitting and overoptimization in RLHF. In *Forty-first International Conference on Machine Learning*, 2024.

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