TOWARDS A THEORY OF INFERENCE-TIME ALIGNMENT WITH UNKNOWN REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative model alignment has received broad interest, and significant progress has been made in supervised fine-tuning and inference-time computation. Yet, alignment has remained poorly understood from a statistical learning perspective. We formulate inference-time alignment as a weak-to-strong learning problem, where a reference policy (weak model) is assumed to be fairly good and the goal is to produce a strong model that predicts a good response at test time with arbitrarily high probability. Our problem is formulated as learning from scratch — everything is learned from data rather than assuming access to a good reward estimate, and thus differs from the existing inference-time alignment theory. Our framework shares similarity with the recent work of Joshi et al. (2026), where for each prompt, there could be multiple good responses. Our definition of the alignment learnability follows the standard PAC learning principle. We introduce a novel combinatorial dimension of the reward class which we call the alignment dimension, and show that it completely characterizes the alignment learnability — a reward class is alignment learnable if and only if its alignment dimension is finite. The core of our learning procedure works by learning a pairwise comparator and then running a tournament over candidate responses. We believe that our results might shed light toward establishing a complete theoretical understanding of alignment.

1 INTRODUCTION

Alignment — aligning AI agents to behave in accordance with human intentions — is motivated by the observation that while modern generative models acquire impressive capabilities from scalable training, they may exhibit unintended or harmful behavior Amodei et al. (2016); Ouyang et al. (2022). Let us take language model alignment as an example.[1] Traditional methodology for language model alignment falls into two categories, namely, Reinforcement Learning from Human/AI Feedback (RLHF/RLAIF) Christiano et al. (2017); Ziegler et al. (2020); Bai et al. (2022b); Lee et al. (2024) and inference-time alignment Khanov et al. (2024); Sun et al. (2024); Huang et al. (2025a). RLHF/RLAIF methods usually involve two steps, where a parameterized reward function is first learned using the curated preference data annotated by human beings or AI, and is then used to optimize the model to produce responses that receive higher rewards. In contrast, inference-time alignment approaches work by selecting or steering the model's generated responses according to a reward signal at inference time, and are thus free from updating the model parameters. We defer a detailed discussion of these two paradigms to Appendix A.

Yet, despite its increasing popularity in both empirical and theoretical investigations, only a very few works have tried to model the alignment problem from a statistical learning perspective and explored its learn-

[1] Throughout the paper, we use language model alignment as an example for presentation. It is worth emphasizing that our theory captures inference-time alignment in general for other generative models.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ing performance guarantee. In fact, theoretical works on RLHF have largely focused on designing better preference-optimization objectives, motivated by the empirical observation of reward overoptimization (a.k.a. reward hacking) Gheshlaghi Azar et al. (2024); Liu et al. (2024); Huang et al. (2025b); Xie et al. (2025); or on deriving sample complexity guarantees under structured learning models, such as assuming linear rewards, log-linear policies, and contextual-bandit formulation Xiong et al. (2024); Nika et al. (2024); Li et al. (2024); Das et al. (2026). For inference-time alignment, existing theoretical researches have conditioned on a reward model satisfying prescribed accuracy or uncertainty assumptions Huang et al. (2025a); Beirami et al. (2025); Aminian et al. (2026); Sriraman & Block (2026), rather than studying the end-to-end statistical problem of learning the reward model from finite preference data and subsequently using it to align unseen test instances. Motivated by this, we ask a more basic statistical learning question:

Question 1. *Which reward classes are learnable for inference-time alignment, given access only to the preference data?*

In this work, we provide an answer to the above question by studying the problem of inference-time alignment from a statistical learning perspective. For simplicity, we call our problem alignment rather inference-time alignment whose formal definition can be found in Section 2. As a preliminary work, we restrict to the basic setting where the reward class contains binary reward functions, i.e., $\mathcal{R} \subseteq \{0,1\}^{\mathcal{X}\times\mathcal{Y}}$ and assume that the true reward function lies in class, i.e., $r^* \in \mathcal{R}$. Unlike the existing routes, we formulate the alignment problem as a boosting/weak-to-strong learning problem, where the data is generated from a reference/base policy π_{ref} whose performance is assumed to be fair, in a sense that it has a constant probability of generating a good response (with r^* value 1) for a given input prompt. Therefore, a reference policy can be thought of as a weak model in the context of weak-to-strong learning. In practice, π_{ref} is usually obtained from a pretrained language model or a supervised fine-tuning (SFT) model. For instance, InstructGPT uses the pretrained GPT-3 as the reference model Ouyang et al. (2022).[2] This is because pretrained models typically possess nice properties such as linguistic fluency and broad knowledge, but their pretraining does not pay much attention to following human instructions or preferences. On this account, practical language model alignment methods are usually designed to output selective responses generated from (some policy close to) π_{ref}. The goal of alignment is to boost the performance of the reference model using the training data with human-annotated preferences. We adopt the classical (ϵ, δ) standard of statistical learning to measure the performance, namely, with probability at least $1-\delta$, the probability that the learner (aligned model) "predicts" a response with r^* value 0 is at most ϵ. Thus, the aligned model can be viewed as a stronger model as the target of boosting. In particular, we consider inference-time alignment — the learner is expected to select a response with r^* value 1 from multiple π_{ref}-generated candidates with arbitrarily high probability at test time.

Our main contributions are summarized as follows. To the best of our knowledge, we are the first to develop a statistical learning framework for the general alignment problem, in particular inference-time alignment. Our problem is formulated as learning from scratch — everything is learned from data, which we call "inference-time alignment with unknown rewards". This distinguishes our work from existing inference-time alignment theoretical works since they assume having access to a good reward estimate (see Appendix A for a detailed discussion). Although superficially related to several existing statistical learning frameworks such as (i) multiclass learning — the learner outputs only one of the multiple labels for each test instance Natarajan & Tadepalli (1988); Natarajan (1989) and (ii) list learning — the learner outputs multiple plausible labels for each instance rather than just one Brukhim et al. (2022), we notice that our formulation differs from them — multiple labels might be correct, but the learner only needs to find one of them. In fact, our learning framework is closer to the recent works of Joshi et al. (2026); Pour et al. (2026) (see Section 1.1 for a detailed discussion), and might be of independent interest to the learning theory community. We propose a novel combinatorial dimension of the reward class, which we name the alignment dimension. We then

[2]Other common reference models include the LLaMA series Touvron et al. (2023), the Gemma series Team et al. (2024), the Qwen series Bai et al. (2023), etc.

provide a complete characterization of the learnability of the alignment problem, which we call the alignment learnability — any reward class is alignment learnable if and only if its alignment dimension is finite. We consider our work as an initial step towards establishing a statistical learning theory for the general alignment problem, potentially shedding light on developing more systematic principles for alignment in practice.

1.1 Related Work

The most related literature is Joshi et al. (2026). They studied the problem of language model alignment from a statistical learning perspective. They also assumed that the true reward function r^* lies in a low-cardinality reward class $\mathcal{R} \subseteq \{0,1\}^{\mathcal{X}\times\mathcal{Y}}$.[3] In their primary setting, the learner receives i.i.d. samples $\{x_i, y_i\}_{1\le i\le m}$ where each sample contains $x_i \sim \mathcal{D}$ and y_i being one of the good responses to the input x_i, i.e., $r^*(x_i, y_i) = 1$. Indeed, they assumed that each y_i is generated from a demonstration policy $\tilde{\pi}(\cdot|x_i)$ supported on the set of good responses given x_i. Denote $V_{r^*}(\pi) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(\cdot|x)}[r^*(x,y)]$. Their main result is that, with high probability,

$$\left|V_{r^*}(\hat{\pi}) - \sup_{\pi} V_{r^*}(\pi)\right| = O\left(\frac{\log(|\mathcal{R}|)}{m}\right).$$

Their learning algorithm utilizes the classical multiplicative-weights update by maintaining weights over all candidate reward functions and predicting according to their weighted aggregate. Moreover, they also studied a natural extension which they called the pass@k metric, where the learner still receives training data with a single correct response per sample, but at test time, the learned policy can output a tuple of responses whose reward is the best reward among the candidates. Under this extension, they improved the convergence rate from $O(\log(|\mathcal{R}|)/m)$ to $O(\log_{k+1}(|\mathcal{R}|)/m)$.

To clarify our contributions, we list the main differences between our work and theirs as follows. (i) In their model, the learner is required to generate a good response, while our alignment formulation requires the learner to select among supplied candidate responses, mainly for capturing the inference-time alignment. (ii) One of the main limitations of their results is that they are restricted to the case where the reward class has a finite cardinality. In contrast, our work provides a complete learnability guarantee for any (infinite) reward class. (iii) They have each training sample containing a single correct response, whereas we receive pairs of reference-policy samples together with their binary reward labels. (iv) Finally, their pass@k extension is the closest to our formulation. However, the success of a learning algorithm in their model only requires that the output tuple at test time contains a good response (which is very close to the list learning[4]), while our definition of success requires exactly picking out the correct one.

Pour et al. (2026) instead considered an online learning variant. They introduced three types of reward feedback models, namely, the mistake-unknown feedback model — after predicting, the learner only sees a correct response but is not told whether its prediction was correct or not, the mistake-known feedback model — the learner is told a correct response as well as the correctness of its prediction, and the set-valued feedback model — the learner sees the entire multiset of correct responses after prediction. They further introduced three generalized versions of the Littlestone dimension Littlestone (1988) to characterize the mistake bound of these three online learning frameworks, respectively.

2 Preliminaries

Throughout the paper, let $\mathcal{X}$ be an instance space and $\mathcal{Y}$ be an action space. Let $\mathcal{R} \subseteq \{0,1\}^{\mathcal{X}\times\mathcal{Y}}$ be a reward class containing reward functions $r : \mathcal{X}\times\mathcal{Y} \to \{0,1\}$. Let $\mathcal{U} = \mathcal{X}\times\mathcal{Y}\times\mathcal{Y}$. For any reward function

[3] They further extend their results to $\mathcal{R} \subseteq [0,1]^{\mathcal{X}\times\mathcal{Y}}$. For simplicity, we discuss here the case of binary rewards.

[4] List learning assumes exactly one correct label, but their model assumes multiple correct labels. In other words, their learner is expected to output a tuple that intersects with the true label set.

$r : \mathcal{X} \times \mathcal{Y} \mapsto \{0, 1\}$, define its induced set-valued acceptance function as $A_r : \mathcal{U} \mapsto \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$ satisfying $A_r(u) = \{j \in \{0, 1\} : r(x, y_j) = 1\}$ for any $U \in \mathcal{U}$. For a reward class $\mathcal{R}$, define its induced set-valued acceptance class as $\mathcal{A}(\mathcal{R}) = \{A_r : r \in \mathcal{R}\}$. Our learning problem can be described as follows.

Assume the realizable case where an unknown target reward function $r^* \in \mathcal{R}$ is to be learned. Let $\mathcal{D}_X$ be some unknown distribution over $\mathcal{X}$ and let $\gamma \in (0, 1)$. Let π_{ref} be a conditional distributional measure such that for $\mathcal{D}_X$-almost every $x \in \mathcal{X}$,

$$\pi_{\text{ref}}(y \in \mathcal{Y} : r^*(x, y) = 1|x) \geq \gamma.^{5}$$

It is worthwhile to emphasize that we only have oracle (query) access to π_{ref}, but have no distribution information about it. For a sample size $m \in \mathbb{N}$, we collect m i.i.d. samples, $S_m = \{(U_1, A_{r^*}(U_1)), \ldots, (U_m, A_{r^*}(U_m))\}$, according to the following. For every $i \in [m]$, $U_i = (X_i, Y_i^{(0)}, Y_i^{(1)}) \in \mathcal{U}$ is a random tuple where

$$X_i \sim \mathcal{D}_X \text{ and } Y_i^{(0)}, Y_i^{(1)} \overset{\text{i.i.d.}}{\sim} \pi_{\text{ref}}(\cdot|X_i). \tag{1}$$

In brief, each sample contains an instance X_i, a pair of labels $Y_i^{(0)}, Y_i^{(1)}$, and an acceptance set $A_{r^*}(U_i) \subseteq \{0, 1\}$ indicating which labels are accepted by the target reward r^*. For simplicity, we denote by $\mathcal{D}_{\text{pair}}$ the distribution over $\mathcal{U}$ defined in Equation (1). At test time, we are given a test prompt $X \sim \mathcal{D}_X$ accompanied with k labels $Y^{(1)}, \ldots, Y^{(k)}$ i.i.d. generated from $\pi_{\text{ref}}(\cdot|X)$ and we denote this as $Z \sim \mathcal{D}_k$, where $\mathcal{D}_k$ is the distribution on $\mathcal{X} \times \mathcal{Y}^k$ defined as above.

The goal is to design a learning algorithm $\mathbf{A}$ that takes S_m as input and outputs a hypothesis $\mathbf{A}(S_m) : \mathcal{X} \times \mathcal{Y}^k \to [k]$ that, given a test tuple $Z = (X, Y^{(1)}, \ldots, Y^{(k)})$, predicts an index $\mathbf{A}(S_m)(Z) = \hat{i}_m(Z) \in [k]$ such that the corresponding response $Y^{(\hat{i}_m(Z))}$ satisfying $r^*(X, Y^{(\hat{i}_m(Z))}) = 1$ with high probability. Note that this exactly matches the inference-time alignment since the learner selects a good response from candidates rather than predicts one. More formally, we define the learnability of our problem as follows.

Definition 1 (**Alignment learnability**). *A reward class $\mathcal{R}$ is called alignment learnable if there exists a learning algorithm $\mathcal{A}$ such that the following holds. For any $\gamma, \epsilon, \delta \in (0, 1)$, there exists $k(\gamma, \epsilon, \delta) < \infty$ such that, for any $k \geq k(\gamma, \epsilon, \delta)$, there exists $m(k, \epsilon, \delta) < \infty$ such that, for every target reward function $r^* \in \mathcal{R}$, every distribution $\mathcal{D}_X$ over $\mathcal{X}$, every reference policy π_{ref} satisfying $\Pr_{Y \sim \pi_{\text{ref}}(\cdot|x)}(r^*(x, Y) = 1) \geq \gamma$ for $\mathcal{D}_X$-almost every x, and every $m \geq m(k, \epsilon, \delta)$,*

$$\Pr_{Z \sim \mathcal{D}_k}\left[r^*(X, Y^{(\hat{i}_m(Z))}) = 1\right] \geq 1 - \epsilon,$$

with probability at least $1 - \delta$ over the randomness of S_m and the internal randomness of the learner.

Remark. *In conclusion, a reward class $\mathcal{R}$ is alignment learnable if it requires (i) a finite training sample size m; and (ii) a finite inference-time sampling round k. The assumption of π_{ref} implies that $k \geq \log(1/\epsilon)/\gamma$ suffices to guarantee that a good response exists in the test data with probability at least $1 - \epsilon$. The finiteness of the alignment dimension further ensures the finiteness of m.*

2.1 Main results

Our main result is a clean characterization of the alignment learnability as defined in Definition 1, thus completely answering Question 1. We introduce a novel combinatorial dimension, which we call the "alignment dimension", and show that the alignment dimension of a reward class characterizes its alignment learnability (Theorem 1).

[5] It is relatively strong to assume that π_{ref} is fairly good uniformly over the instance space $\mathcal{X}$. check Section 2.2 for further discussion.

Definition 2 (Alignment dimension). *Let $d \in \mathbb{N}$. We say that a set of d triples $\{u_1, \ldots, u_d\}$, where for every $i \in [d]$, $u_i = (x_i, y_i^{(0)}, y_i^{(1)}) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ are alignment shattered by $\mathcal{R}$ if, for every boolean vector $\boldsymbol{b} \in \{0,1\}^d$, there exists a reward function $r_{\boldsymbol{b}} \in \mathcal{R}$ satisfying that for every $i \in [d]$, $r_{\boldsymbol{b}}(x_i, y_i^{(b_i)}) = 1$ and $r_{\boldsymbol{b}}(x_i, y_i^{(1-b_i)}) = 0$. It is worth emphasizing that $x_1, \ldots, x_d$ are not required to be distinct. The alignment dimension of $\mathcal{R}$, denoted by $d_{\mathrm{ALN}}(\mathcal{R})$, is the largest $d \in \mathbb{N} \cup \{0\}$ for which such an alignment-shattered set exists. If arbitrarily large finite sets are alignment-shattered, we say that $d_{\mathrm{ALN}}(\mathcal{R}) = \infty$.*

Theorem 1. *$\mathcal{R}$ is alignment learnable if and only if $d_{\mathrm{ALN}}(\mathcal{R}) < \infty$.*

2.2 Remarks on assumption of reference policy

In our model, we insist that for $\gamma \in (0,1)$, for any prompt $x \in \mathcal{X}$, the reference model π_{ref} outputs a correct answer with probability at least γ. This assumption is impractical. Indeed, if we prompted "prove the Riemann Hypothesis", our reference model is assumed to give us a correct proof with non-trivial probability. Hence, it is more realistic to assume that the reference model only performs well on some subset of $\mathcal{X}$. We consider the following two loss metrics under the above weaker assumption of π_{ref}. For a prompt x where π_{ref} has less than γ probability of outputting a good response, (i) we do not consider that our learner makes a mistake no matter what it predicts. This has already been captured by our theory since we can restrict the instance space to exclude such prompts. (ii) We require the learner to output **IDK** ("I Don't Know!") if and only if no good response exists within $\tilde{\Omega}(1/\gamma)$ inference-time sampling rounds. In fact, we generally prefer the aligned model to say **IDK** rather than to hallucinate. However, it turns out that this problem is equivalent to learning the reward function in the worst case, and thus the learnability is characterized by the VC dimension (see Definition 3) of the reward class $\mathcal{R}$. To illustrate this, we give the following example.

Example 1. *Let $\mathcal{X}$ be an infinite space and $|\mathcal{Y}| = 2$. Let $\{z_1 = (x_1, y^{(0)}, y^{(1)}), z_2 = (x_2, y^{(0)}, y^{(1)}), \ldots\}$ be an infinite space. For any $\boldsymbol{b} \in \{0,1\}^{\mathbb{N}}$, define a reward function $r_{\boldsymbol{b}}$ according to $r_{\boldsymbol{b}}(x_i, y^{(0)}) = 0, r_{\boldsymbol{b}}(x_i, y^{(1)}) = b_i$, for any $i \in \mathbb{N}$. Define a reward class $\mathcal{R} = \{r_{\boldsymbol{b}}, \forall \boldsymbol{b} \in \{0,1\}^{\mathbb{N}}\}$, and $\pi_{\mathrm{ref}}(y^{(0)}|x) = \pi_{\mathrm{ref}}(y^{(1)}|x) = 1/2$.*

Note that whether the learner outputs $y^{(1)}$ or **IDK** is equivalent to the problem of whether $r^*(x_i, y^{(1)}) = 1$ or not. This requires to exactly learn r^* given samples from $\mathcal{X} \times \{y^{(1)}\}$. Note that the VC dimension and the alignment dimension of $\mathcal{R}$ is infinite and 0, respectively. Therefore, this task is strictly harder than our alignment formulation.

Organization. In Section 3, we provide several examples to rule out some natural learning strategies for our formulation of the inference-time alignment problem. Those examples also lead us to consider the one-inclusion-graph learning algorithm and further interpret the alignment problem as learning partial concept classes. Section 4 provides the main theorem and our learning strategy with high-level proof ideas. The complete proofs are provided in Appendix C.1 and Appendix C.2. Section 5 concludes this paper, and more detailed future directions are discussed in Appendix D.

3 Examples

In this section, we provide several concrete examples that (i) distinguish our alignment problem from the standard binary/multiclass classification problems; (ii) rule out certain candidates of the alignment learnability characterization as well as some natural learning algorithms; and (iii) guide us towards considering the one-inclusion-graph algorithm. For brevity, we postpone the detailed explanation to Appendix B.

3.1 ALIGNMENT IS NOT A STANDARD CLASSIFICATION PROBLEM

Our first example shows that our alignment problem differs intrinsically from the standard binary/multi-class classification problem, thus ruling out several candidates for the characterization of the alignment learnability. Before proceeding to the example, we present the formal definition of the classic VC dimension (for binary classification). The discussion about multiclass classification appears in Appendix B.1.

Definition 3 (VC dimension Vapnik & Chervonenkis (1971)). *We say that a set $\{z_1 = (x_1, y_1), \ldots, z_n = (x_n, y_n)\}$ is shattered by $\mathcal{R}$, if for every $\boldsymbol{b} \in \{0,1\}^n$, there exists $r_{\boldsymbol{b}} \in \mathcal{R}$, such that $r_{\boldsymbol{b}}(z_i) = b_i$ for every $i \in [n]$. The VC dimension of $\mathcal{R}$ is defined as the largest $d \in \mathbb{N} \cup \{0\}$ such that there exists a set $\{z_1, \ldots, z_d\}$ shattered by $\mathcal{R}$. If no largest finite d exists, we say that $\mathcal{R}$ has an infinite VC dimension.*

Example 2. *Let $\mathcal{X}$ be an infinite space and $|\mathcal{Y}| = 2$. Let $\{z_1 = (x_1, y^{(0)}, y^{(1)}), z_2 = (x_2, y^{(0)}, y^{(1)}), \ldots\}$ be an infinite space. For any $\boldsymbol{b} \in \{0,1\}^{\mathbb{N}}$, define a reward function $r_{\boldsymbol{b}}$ according to $r_{\boldsymbol{b}}(x_i, y^{(0)}) = 1, r_{\boldsymbol{b}}(x_i, y^{(1)}) = b_i$, for any $i \in \mathbb{N}$. Define a reward class $\mathcal{R} = \{r_{\boldsymbol{b}}, \forall \boldsymbol{b} \in \{0,1\}^{\mathbb{N}}\}$.*

A natural idea is to first learn a reward function by solving a binary classification problem where the fundamental Empirical Risk Minimization (ERM) rule achieves a nearly optimal guarantee, and then apply the Best-of-N or rejection sampling at test time according to the learned reward function. This idea suggests that the VC dimension of the reward class $\mathcal{R}$ might be the characterization of the alignment learnability (achieved by ERM). However, Example 2 refutes this conjecture by providing an instance of $\mathcal{R}$ that has an infinite VC dimension, yet is alignment learnable *without any samples*. Due to the space limitation, we leave a detailed discussion to Appendix B.1.

3.2 LEARN-AND-OPTIMIZE PRINCIPLE FAILS

Recall that Example 2 has ruled out the strategy of learning a reward function by training an ERM on data and then optimizing at test-time based on the learned reward function. More generally, it falls into a broad class of algorithms following the "learn-and-optimize" principle, that is, to first learn a reward function (not necessarily in the reward class) only based on the training data (independent of the test instance) and then select the best response (under the evaluation of the learned reward function) at test time (e.g., doing Best-of-N or rejection sampling). Our next example shows that any learning strategy following the "learn-and-optimize" principle fails to guarantee alignment learnability.

Example 3. *Let $p \in \mathbb{N}$. Let $\{z_1 = (x_1, y^{(1)}, y^{(2)}, \ldots, y^{(2p+1)}), z_2 = (x_2, y^{(1)}, y^{(2)}, \ldots, y^{(2p+1)}), \ldots\}$ be an infinite space. Define $\mathcal{R} = \{r_{\boldsymbol{s}}, \forall \boldsymbol{s} \in [2p+1]^{\mathbb{N}}\}$, where for any $i \in \mathbb{N}$ and any $j \in [2p+1]$, $r_{\boldsymbol{s}}(x_i, y^{(j)}) = 1$, if $j \le s_i$ and $r_{\boldsymbol{s}}(x_i, y^{(j)}) = 0$, if $j > s_i$.*

Let $\boldsymbol{s} = \{s_i\}_{i \in \mathbb{N}}$ such that s_i is i.i.d. sampled from $\mathrm{Unif}([2p])$, where $\mathrm{Unif}([2p])$ stands for the uniform distribution on $[2p]$. Let $r^* = r_{\boldsymbol{s}}$ and $\pi_{\mathrm{ref}}(y^{(s_i)}|x_i) = \pi_{\mathrm{ref}}(y^{(s_i+1)}|x_i) = 1/2$. For any test prompt x_i that does not show up in the training data S_m, assume that the sampling round k is sufficiently large such that at least one good response exists in the candidate pool. Now, if the candidate pool contains two distinct responses, for any learned binary reward function $\hat{r}$, $\mathbb{E}_{\boldsymbol{s}}[\Pr_{Y \sim \pi_{\mathrm{ref}}(\cdot|x_i)}(\hat{r}(x_i, Y) \neq r^*(x_i, Y))] \ge 1/2$. Therefore, the learn-and-optimize principle fails here. However, $d_{\mathrm{ALN}}(\mathcal{R}) = 0$ and the alignment learnability is achievable for Example 3 by utilizing test-time sampling (see Appendix B.2 for details).

We emphasize that Example 3 does not rule out realizing the alignment learnability by the learn-and-optimize principle when a real-valued reward function (which might encode rank information) can be learned for optimization at test time, which we leave open for future work.

3.3 PESSIMISM PRINCIPLE FAILS

Pessimism — intentionally constructing a conservative, lower-bound estimate of the reward function when uncertainty arises from data scarcity — has been shown to provide provable benefits in offline RL Jin et al. (2021); Xie et al. (2021) and reward function learning Zhu et al. (2023); Zhan et al. (2024).

More formally, a pessimism principle can be interpreted as running the so-called *Closure* algorithm Auer & Ortner (2007) on a test instance-dependent version space. The following example shows that the pessimism principle yields a reward function that assigns reward value 0 to both $(x, y^{(0)})$ and $(x, y^{(1)})$, and thus cannot return a good response at test time. The detailed explanation is deferred to Appendix B.3

Example 4. *Let $\mathcal{X}$ be an infinite space and $|\mathcal{Y}| = 2$. Let $\{z_1 = (x_1, y^{(0)}, y^{(1)}), z_2 = (x_2, y^{(0)}, y^{(1)}), \ldots\}$ be an infinite space. Define $\mathcal{R} = \{r_{\boldsymbol{b},i}, \forall \boldsymbol{b} \in \{0,1\}^{\mathbb{N}}, \forall i \in \mathbb{N}\}$, where $r_{\boldsymbol{b},i}$ is defined according to*

$$\begin{cases} r_{\boldsymbol{b},i}(x_j, y^{(0)}) = 1, & r_{\boldsymbol{b},i}(x_j, y^{(1)}) = b_j, & j \neq i \\ r_{\boldsymbol{b},i}(x_j, y^{(0)}) = b_j, & r_{\boldsymbol{b},i}(x_j, y^{(1)}) = 1, & j = i \end{cases}, \quad \forall j \in \mathbb{N}.$$

Given the failure of the ERM and pessimism principles, tie-breaking (at the test point) seems to matter a lot, guiding us towards considering the one-inclusion graph algorithm.

4 PROOF OF THEOREM 1

In this section, we prove the learnability characterization stated in Theorem 1. The main idea is to reduce the alignment problem to learning a partial binary comparison class. The learner first learns to distinguish a good response from a bad response when they are presented as a pair. At test time, it compares every pair of candidate responses and picks one of the most-winning candidates. We prove the upper bound in Section 4.1-Section 4.3. We then prove the matching necessity result in Section 4.4 by constructing a hard family of independent pairwise preferences.

4.1 LEARNING THE PAIRWISE COMPARISON CLASS

For any reward function $r \in \mathcal{R}$, define its induced partial pairwise comparison function $g_r : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to \{0, 1, *\}$ according to

$$g_r(x, y^{(0)}, y^{(1)}) = \begin{cases} 0, & r(x, y^{(0)}) = 1 \text{ and } r(x, y^{(1)}) = 0, \\ 1, & r(x, y^{(0)}) = 0 \text{ and } r(x, y^{(1)}) = 1, \\ *, & r(x, y^{(0)}) = r(x, y^{(1)}). \end{cases}$$

Let $\mathcal{G}(\mathcal{R}) = \{g_r : r \in \mathcal{R}\}$ be the induced partial comparison class. It is easy to notice that the VC dimension of the partial class $\mathcal{G}(\mathcal{R})$ is equivalent to the alignment dimension of the reward class $\mathcal{R}$, $d_{\mathrm{ALN}}(\mathcal{R})$. It is easy to see that we can generate $g_{r^*}(u)$ from our observed label, $A_{r^*}(u)$.

We first recall a standard consequence of the PAC learnability of partial concept classes. We state it in a form that controls the error under the ambient distribution $\mathcal{D}_{\mathrm{pair}}$, rather than only under the distribution conditioned on observing a non-$*$ label.

Lemma 1. *Let $\mathcal{H} \subseteq \{0, 1, *\}^{\mathcal{U}}$ be a partial concept class with $\mathrm{VC}(\mathcal{H}) = d < \infty$, and let $h^* \in \mathcal{H}$. Suppose that $U_1, \ldots, U_m$ are generated i.i.d. from a distribution $\mathcal{D}$ over $\mathcal{U}$ and that the partial labels $h^*(U_1), \ldots, h^*(U_m)$ are observed. There exists a learning algorithm $\mathbb{A}_{\mathrm{OIG}}$ which returns a total predictor $\hat{h}_m : \mathcal{U} \to \{0, 1\}$ such that, for any $\eta, \delta \in (0, 1)$, $m \geq (C/\eta)[(d \vee 1)\log^2(e(d \vee 1)/\eta) + \log(2/\delta)]$ for a universal constant $C > 0$ implies $\Pr_{U \sim \mathcal{D}}(h^*(U) \in \{0, 1\} \text{ and } \hat{h}_m(U) \neq h^*(U)) \leq \eta$ with probability at least $1 - \delta$ over the training sample and the internal randomness of the learner. Here, $d \vee 1 = \max\{d, 1\}$.*

Training:

- For every $i \in [m]$, set $U_i = (X_i, Y_i^{(0)}, Y_i^{(1)})$, which is sampled from $\mathcal{D}_{\text{pair}}$ and determine the partial label $g^*(U_i) \in \{0, 1, *\}$.
- Discard the examples with label $*$ and run the partial-class OIG learner to obtain a pairwise predictor $\hat{g}_m : \mathcal{U} \to \{0, 1\}$.

Inference:

- Let $Z = (X, Y^{(1)}, \ldots, Y^{(k)})$ be a test instance and initialize $W_j = 0$ for every $j \in [k]$.
- For every $1 \le a < b \le k$, compute $\hat{g}_m(X, Y^{(a)}, Y^{(b)})$.
  - If the prediction is 0, update $W_a \leftarrow W_a + 1$.
  - If the prediction is 1, update $W_b \leftarrow W_b + 1$.
- Return $Y^{(\hat{j})}$, where $\hat{j}$ is chosen uniformly from $\arg\max_{j \in [k]} W_j$.

Figure 1: The pairwise tournament learning procedure used in Theorem 2.

The proof strategy is to bound the probability of two different failure events. One is that the partial concept learner makes a mistake, and the other is that too many samples have label $*$ and are discarded, but at test time $h^*(U) \in \{0, 1\}$. The first bound comes from the work of Alon et al. (2021) and the second bound comes from the multiplicative Chernoff bound. For brevity, the complete proof is in Appendix C.1.

4.2 The pairwise tournament learning procedure

We now present the learning procedure. The training stage learns the partial pairwise comparison class using Lemma 1. At test time, the learner compares every pair of candidate responses. Each pairwise winner receives one point, and the learner returns a response with the largest number of wins. Our learning strategy is formally presented in Figure 1.

The next lemma explains why the tournament converts accurate pair comparisons into a correct selection.

Lemma 2. *Fix a test tuple $Z = (X, Y^{(1)}, \ldots, Y^{(k)})$ satisfying that there exists $j \in [k]$, such that $r^*(X, Y^{(j)}) = 1$. Suppose that for every pair containing exactly one good response, $\hat{g}_m$ selects correctly. Then every maximizer of the score W_j makes $r^*(X, Y^{(j)}) = 1$.*

Proof. Let $G = |\{j \in [k] : r^*(X, Y^{(j)}) = 1\}|$ and $B = k - G$. Every good response defeats all B bad responses, and therefore has score at least B. On the other hand, every bad response loses to all G good responses and can defeat at most the other $B - 1$ bad responses. Hence, every bad response has score at most $B - 1$. It follows that every maximizer of the score is good. □

4.3 Upper bound

We are now ready to prove the sufficiency direction of Theorem 1, which is stated as Theorem 2 formally.

Theorem 2. *Suppose that $d_{\text{ALN}}(\mathcal{R}) = d < \infty$. For any $\gamma, \epsilon, \delta \in (0, 1)$, choose $k \ge \max\{2, \lceil \log(2/\epsilon)/\gamma \rceil\}$. There exists a universal constant $C > 0$ such that the pairwise tournament procedure satisfies*

$$\Pr_{Z \sim \mathcal{D}_k} \left(r^*(X, Y^{(\hat{i}_m(Z))}) = 1 \right) \ge 1 - \epsilon$$

with probability at least $1 - \delta$, provided that $m \ge Ck^2((d \vee 1)\log^2((d \vee 1)k/\epsilon) + \log(1/\delta))/\epsilon$.

The high-level idea of this proof is as follows. There are two bad events in this learning scenario, one is that there is no good responses contained in the test data. The other is that our learned predictor makes at least one mistake on the comparisons between every pair of responses, which contains exactly one good responses conditioned on there is at least one good response in test data. We can bound the probability of both events. The bound on the first event comes from the assumption on π_{ref} and the bound on the second event comes from Lemma 1. When both of the events does not happen, Lemma 2 shows that our selection must be good. The complete proof is in Appendix C.1.

4.4 Lower bound

We next prove that the finiteness of the alignment dimension is necessary. The lower bound uses the fact that a shattered family contains many independent pairwise preferences, including preferences associated with different pairs sharing the same prompt.

Theorem 3. *Suppose that $d_{\mathrm{ALN}}(\mathcal{R}) \geq d$. Fix any learning algorithm, any training sample size $m \in \mathbb{N}$, and any test data with k responses, where $k \in \mathbb{N}$. If $d \geq 8mk$, there exist a target reward function $r^* \in \mathcal{R}$, a distribution $\mathcal{D}_X$, and a reference policy π_{ref} satisfying the assumption of π_{ref} with $\gamma = 1/2$, such that*

$$\Pr_{S_m}\left(\Pr_{Z\sim\mathcal{D}_k}\left(r^*(X, Y^{(\hat{i}_m(Z))}) = 0\right) \geq \frac{1}{8}\right) \geq \frac{2}{7}.$$

The proof relies on the probabilistic method. By uniformly randomly choosing the target concept, we can show that the expected value of $\mathbb{E}_{S_m}[\Pr_{Z\sim\mathcal{D}_k}(r^*(X, Y^{(\hat{i}_m(Z))}) = 0)]$ is large. Therefore, there must exist a target reward function such that $\mathbb{E}_{S_m}[\Pr_{Z\sim\mathcal{D}_k}(r^*(X, Y^{(\hat{i}_m(Z))}) = 0)]$ is large. Then by the similar reasoning of Markov inequality, we can show this theorem. The complete proof is in Appendix C.2.

5 Discussion

We formulate the problem of inference-time alignment into a statistical learning framework, where the goal is to boost the prediction performance of a reference policy by leveraging the preference data. We define the notion of *alignment learnability* in the sense of weak-to-strong learning: given a reference policy (weak model), is there a finite sample size such that i.i.d. training data can be used to produce a strong model that predicts good responses with arbitrarily high probability at test time? We seek a characterization of alignment learnability — asking which reward classes are alignment learnable. As an initial work, we restrict to the realizable case as well as binary rewards. We introduce a novel and natural combinatorial dimension of the reward class called the alignment dimension, which provides a complete characterization of the alignment learnability — a reward class is alignment learnable if and only if its alignment dimension is finite.

We consider our work as an initial step towards establishing a general statistical learning theory for alignment. Since we just figured out its learnability, a natural follow-up question is to quantify the optimal sample complexity required to achieve the $(\gamma, \epsilon, \delta)$ alignment learnability, namely, the optimal dependencies in terms of γ, ϵ and δ. Moreover, we notice that our learning model differs from existing frameworks such as multiclass learning and list learning, and itself might be of independent interest to the learning theory community. Indeed, various extensions such as online learning and active learning could be interesting to study. We defer to Appendix D a discussion of several interesting problems closely related to our formulation.

AI disclosure

In this work, we used generative AI tools for assisting in the writing of proofs. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, pro-

vide critical ingredients for proving mathematical claims, and proposing or refining hypotheses, generating synthetic data sets, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, and interpreting results are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Noga Alon, Steve Hanneke, Ron Holzman, and Shay Moran. A theory of PAC learnability of partial concept classes. In *IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 658–671, 2021. URL https://ieeexplore.ieee.org/abstract/document/9719837?

Gholamali Aminian, Idan Shenfeld, Amir Reza Asadi, Ahmad Beirami, and Youssef Mroueh. Best-of-N through the smoothing lens: KL divergence and regret analysis. In *International Conference on Learning Representations*, volume 2026, pp. 105057–105090, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/abb7833fc3f38e785e3f9777c74610e5-Paper-Conference.pdf.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016. URL https://arxiv.org/pdf/1606.06565.pdf.

Peter Auer and Ronald Ortner. A new PAC bound for intersection-closed concept classes. *Machine Learning*, 66(2):151–163, 2007. URL https://link.springer.com/article/10.1007/s10994-006-8638-3.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. URL https://arxiv.org/abs/2309.16609.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022a. URL https://arxiv.org/abs/2204.05862.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI feedback, 2022b. URL https://arxiv.org/abs/2212.08073.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the Best-of-N alignment policy. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 3580–3602. PMLR, 2025. URL https://proceedings.mlr.press/v267/beirami25a.html.

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. URL http://www.jstor.org/stable/2334029.

Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 943–955, 2022. doi: 10.1109/FOCS54457.2022.00093.

Paul F 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, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *Proceedings of The 27th Conference on Learning Theory*, volume 35, pp. 287–316. PMLR, 2014. URL https://proceedings.mlr.press/v35/daniely14b.html.

Nirjhar Das, Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury. Active preference optimization for sample efficient rlhf. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pp. 96–112. Springer Nature Switzerland, 2026. URL https://link.springer.com/chapter/10.1007/978-3-032-06096-9_6.

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, pp. 4447–4455. PMLR, 2024. URL https://proceedings.mlr.press/v238/gheshlaghi-azar24a.html.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. Is Best-of-N the best of them? Coverage, scaling, and optimality in inference-time alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 25075–25126. PMLR, 2025a. URL https://proceedings.mlr.press/v267/huang25c.html.

Audrey Huang, Wenhao Zhan, Tengyang Xie, Jason Lee, Wen Sun, Akshay Krishnamurthy, and Dylan Foster. Correcting the mythos of KL-regularization: Direct alignment without overoptimization via chi-squared preference optimization. In *International Conference on Learning Representations*, volume 2025, pp. 92647–92701, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e785d64baa2cde06c69d9b7a14636ef2-Paper-Conference.pdf.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline RL? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5084–5096. PMLR, 2021. URL https://proceedings.mlr.press/v139/jin21e.html.

Nirmit Joshi, Gene Li, Siddharth Bhandari, Shiva Kasiviswanathan, Cong Ma, and Nathan Srebro. Learning to answer from correct demonstrations. In *International Conference on Learning Representations*, volume 2026, pp. 44352–44384, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/4975dfc24b3fd21d069842c90dd07c91-Paper-Conference.pdf.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. Args: Alignment as reward-guided search. In *International Conference on Learning Representations*, volume 2024, pp. 17755–17773, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4d1344317478ad99ff5f4e414aeab689-Paper-Conference.pdf.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 26874–26901. PMLR, 2024. URL https://proceedings.mlr.press/v235/lee24t.html.

Zihao Li, Xiang Ji, Minshuo Chen, and Mengdi Wang. Policy evaluation for reinforcement learning from human feedback: A sample complexity analysis. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238, pp. 2737–2745. PMLR, 2024. URL https://proceedings.mlr.press/v238/li24l.html.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988. URL https://link.springer.com/article/10.1023/A%3A1022869011914?

Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. Provably mitigating overoptimization in RLHF: Your SFT loss is implicitly an adversarial regularizer. In *Advances in Neural Information Processing Systems*, volume 37, pp. 138663–138697, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/fa69e968b7319fd42524febd41475fb3-Paper-Conference.pdf.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, volume 37, pp. 124198–124235, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e099c1c9699814af0be873a175361713-Paper-Conference.pdf.

Tom M. Mitchell. Version spaces: a candidate elimination approach to rule learning. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, pp. 305–310, 1977. URL https://dl.acm.org/doi/abs/10.5555/1624435.1624501.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 36577–36590. PMLR, 2024. URL https://proceedings.mlr.press/v235/muldrew24a.html.

Balas K Natarajan. On learning sets and functions. *Machine Learning*, 4(1):67–97, 1989. URL https://link.springer.com/article/10.1007/BF00114804.

B.K. Natarajan and P. Tadepalli. Two new frameworks for learning. In *Machine Learning Proceedings 1988*, pp. 402–415. 1988. URL https://www.sciencedirect.com/science/article/pii/B9780934613644500463.

Andi Nika, Debmalya Mandal, Parameswaran Kamalaruban, Georgios Tzannetos, Goran Radanovic, and Adish Singla. Reward model learning vs. direct policy optimization: A comparative analysis of learning from human preferences. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 38145–38186. PMLR, 2024. URL https://proceedings.mlr.press/v235/nika24a.html.

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, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Alireza F. Pour, Farnam Mansouri, and Shai Ben-David. Learning with multiple correct answers – regret bounds under different feedback models, 2026. URL https://arxiv.org/abs/2602.09402.

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, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Ved Sriraman and Adam Block. Revisiting the (sub)optimality of Best-of-N for inference-time alignment. In *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336, pp. 5980–6028. PMLR, 2026. URL https://proceedings.mlr.press/v336/sriraman26a.html.

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, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. Fast Best-of-N decoding via speculative rejection. In *Advances in Neural Information Processing Systems*, volume 37, pp. 32630–32652, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/3950f6bf5c2eb7435ecf58eaa85cc8c2-Paper-Conference.pdf.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob

Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on Gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. doi: 10.1137/1116025.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. In *International Conference on Learning Representations*, volume 2026, pp. 49450–49483, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/517f9b9c227b9dd51dba4560f37165ed-Paper-Conference.pdf.

Yue Wu, Zhiqing Sun, Rina Hughes, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. In *International Conference on Learning Representations*, volume 2025, pp. 91558–91582, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e48fa1c4f08fd1ae35d5df8352c3106d-Paper-Conference.pdf.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 6683–6694, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/34f98c7c5d7063181da890ea8d25265a-Paper.pdf.

Tengyang Xie, Dylan Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed H Awadallah, and Alexander Rakhlin. Exploratory preference optimization: Harnessing implicit $q*$ -approximation for sample-efficient RLHF. In *International Conference on Learning Representations*, volume 2025, pp. 43632–43669, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6c71ddecdefd911d2f993461972d8b39-Paper-Conference.pdf.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 54715–54754. PMLR, 2024. URL https://proceedings.mlr.press/v235/xiong24a.html.

Yuancheng Xu, Udari Sehwag, Alec Koppel, Sicheng Zhu, Bang An, Furong Huang, and Sumitra Ganesh. Genarm: Reward guided generation with autoregressive reward model for test-time alignment. In *International Conference on Learning Representations*, volume 2025, pp. 73970–73990, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/b7a55b1fa8f2500299e1af3562a1191d-Paper-Conference.pdf.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 57905–57923. PMLR, 2024. URL https://proceedings.mlr.press/v235/yuan24d.html.

Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason Lee, and Wen Sun. Provable offline preference-based reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 29559–29584, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/7efe88bb4138d602e56637cfcf713654-Paper-Conference.pdf.

Heyang Zhao, Chenlu Ye, Quanquan Gu, and Tong Zhang. Sharp analysis for KL-regularized contextual bandits and RLHF. In *Advances in Neural Information Processing Systems*, volume 38, pp. 107964–108002, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/9b04b2d8c488ed1e943401dcdd927216-Paper-Conference.pdf.

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, pp. 43037–43067. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhu23f.html.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL https://arxiv.org/abs/1909.08593.

A More Related Work

RLHF/RLAIF. Reinforcement Learning from Human/AI Feedback (RLHF/RLAIF) framework is summarized as follows. Given an input/prompt x and an action/response y, the evaluation of y on x is assumed to be generated by some latent reward model $r^*(x, y)$, which lies in a reward class $\mathcal{R}$. For a pair of responses (y^w, y^l) on x, human preferences are captured by a theoretical preference model called the Bradley-Terry model Bradley & Terry (1952). Specifically, given r^* as the true reward function, the probability that a response y^w is preferred over another response y^l on the prompt x is stipulated as

$$\mathbb{P}_{r^*}\left(y^w \succ y^l \middle| x\right) = \frac{\exp\left\{r^*(x, y^w)\right\}}{\exp\left\{r^*(x, y^w)\right\} + \exp\left\{r^*(x, y^l)\right\}} =: \sigma\left(r^*(x, y^w) - r^*(x, y^l)\right),$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function. RLHF/RLAIF methods usually consist of two steps — reward maximization followed by policy searching via KL-divergence-regularized reinforcement learning. The training preference dataset $\mathcal{S}_n := \{(x_i, y_i^w, y_i^l)\}_{i=1}^n$ is generated according to $y^w, y^l \sim \pi_{\text{ref}}(\cdot|x)$ where π_{ref} is called the reference policy, and the preferences $y^w \succ y^l$ are annotated by human beings/AI. Reward maximization refers to searching for a reward function $r \in \mathcal{R}$ to maximize the following likelihood objective

$$\mathcal{L}\left(r, \mathcal{S}_n\right) = \mathbb{E}_{(x, y^w, y^l) \sim \mathcal{S}_n}\left[\log \sigma\left(r(x, y^w) - r(x, y^l)\right)\right].$$

The learned reward function $\hat{r}$ will be used in the next step of RL fine-tuning — searching for a policy (conditional distribution) π from some policy hypothesis class Π to maximize the following KL-divergence-regularized RL objective

$$\max_{\pi \in \Pi}\left\{\mathbb{E}_{x \sim \mathcal{S}_n, y \sim \pi(\cdot|x)}\left[\hat{r}(x, y) - \beta \cdot \mathbb{D}_{\text{KL}}\left(\pi(y|x) \,||\, \pi_{\text{ref}}(y|x)\right)\right]\right\},$$

where $\beta > 0$ is a hyperparameter to control the weight of the KL-regularizer.

This objective of RLHF/RLAIF has been extensively studied in the past few years Christiano et al. (2017); Ziegler et al. (2020); Stiennon et al. (2020); Bai et al. (2022b;a); Lee et al. (2024). Early works mainly focused on designing efficient RL algorithms such as the Proximal Policy Optimization (PPO) Schulman et al. (2017) to learn a policy based on a well-estimated reward function. Motivated by the fact that the traditional RLHF pipeline is complex, computationally heavyweight and often unstable, Rafailov et al. (2023) proposed the celebrated Direct Preference Optimization (DPO) algorithm, which directly optimize a policy to adhere to human preferences in a single stage of policy training, without explicit reward modeling. Since then, substantial research effort has shifted toward developing such direct optimization algorithms for various preference learning settings such as iterative/on-policy preference learning Yuan et al. (2024); Xiong et al. (2024); Wu et al. (2025), active preference learning Muldrew et al. (2024); Das et al. (2026), online/exploratory preference learning Xie et al. (2025), and preference learning without using a reference model Meng et al. (2024). More recently, motivated by the fact that many reasoning tasks admit rewards that can be automatically checked by programmatic verifiers rather than relying on expensive human-provided preference signals (e.g., verifying the correctness on mathematical problems and code execution), Reinforcement Learning with Verifiable Rewards (RLVR) has been proposed as a variant of RLHF Lambert et al. (2025); Shao et al. (2024); Wen et al. (2026). Unlike in general RLHF where rewards are typically real-valued, RLVR adopts binary rewards — assigning 1 when the generated response passes an automatic verifier and 0 otherwise — which exactly matches our formulation. Therefore, our formulation captures exactly inference-time alignment with verifiable rewards.

Inference-time alignment. Instead of fine-tuning the language model using the learned reward function as in RLHF, an inference-time alignment algorithm keeps the reference policy π_{ref} fixed and uses the reward model only at test time, such as to do rejection sampling Sun et al. (2024), to guide decoding token by token Xu et al. (2025), to implement planning and search Yao et al. (2023), etc. For instance, a canonical

inference-time alignment algorithm is the Best-of-N (BoN) heuristic Sun et al. (2024); Huang et al. (2025a); Beirami et al. (2025). Specifically, given an imperfect reward model $\hat{r}$ and a new test input x, BoN works by independently sampling N candidate responses $y^{(1)}, \ldots, y^{(N)}$ from $\pi_{\text{ref}}(\cdot|x)$, and returns the one with the highest reward of $\hat{r}$, i.e.,

$$\hat{y}_{\text{BoN}}(x) = y^{(j)}, \quad \text{where} \quad j \in \arg\max_{1 \le i \le N} \hat{r}(x, y^{(i)}).$$

The imperfect reward model $\hat{r}$ can be an open-source reward model or a reward model trained on data separately. As not the main focus, theoretical works on inference-time alignment usually do not involve analyzing the performance of estimating the true reward function r^* by $\hat{r}$, but rather quantify this estimation error and leave it in bounds. For instance, the estimation quality of $\hat{r}$ can be measured by the expected squared error with respect to r^* under the reference policy π_{ref}, i.e.,

$$\mathbb{E}_{y \sim \pi_{\text{ref}}(\cdot|x)} \left[\left(\hat{r}(x, y) - r^*(x, y) \right)^2 \right].$$

In contrast, our learnability result takes both the reward estimation and the test-time computation into consideration since our problem is to learn from scratch — everything should be learned from data without any additional assumptions. Therefore, our theory explains the most general version of the inference-time alignment problem. We refer the readers to the work of Huang et al. (2025a) for a detailed review of inference-time alignment research.

B Detailed Explanations of Examples

To help the reader understand the explanation, we restated all of our examples in this section.

B.1 Alignment is not a standard classification problem

Example 5 (Restatement of Example 2)**.** *Let $\mathcal{X}$ be an infinite space and $|\mathcal{Y}| = 2$. Let $\{z_1 = (x_1, y^{(0)}, y^{(1)}), z_2 = (x_2, y^{(0)}, y^{(1)}), \ldots\}$ be an infinite space. For any $\boldsymbol{b} \in \{0,1\}^{\mathbb{N}}$, define a reward function $r_{\boldsymbol{b}}$ according to $r_{\boldsymbol{b}}(x_i, y^{(0)}) = 1, r_{\boldsymbol{b}}(x_i, y^{(1)}) = b_i$, for any $i \in \mathbb{N}$. Define a reward class $\mathcal{R} = \{r_{\boldsymbol{b}}, \forall \boldsymbol{b} \in \{0,1\}^{\mathbb{N}}\}$.*

Example 2 provides an instance of $\mathcal{R}$ that has an infinite VC dimension, yet is alignment learnable *without any samples*. To see this, note that

- $\mathcal{R}$ has an infinite VC dimension since the (infinite) set $\{(x_1, y^{(1)}), (x_2, y^{(1)}), \ldots\}$ is shattered by $\mathcal{R}$.
- $\mathcal{R}$ has an alignment dimension $d_{\text{ALN}}(\mathcal{R}) = 0$ since any reward function assigns 1 to $(x_i, y^{(0)})$ for any $i \in \mathbb{N}$.
- $\mathcal{R}$ is alignment learnable *without any training samples* using the following learning strategy. For any test prompt x, sample $k = \Theta((1/\gamma)\log(1/\epsilon))$ responses from $\pi_{\text{ref}}(\cdot|x)$ at test time. (This ensures that a good response exists with high probability in the candidate pool. See Section 4.3 for a rigorous reasoning.) Then, return $y^{(0)}$ if any of the k candidates is $y^{(0)}$, and return $y^{(1)}$ otherwise. This is because, if $y^{(0)}$ exists, it is always acceptable since all reward functions (including the target) agree that it is good. Otherwise, if $y^{(0)}$ does not exist, since at least one good response is in the pool, $y^{(1)}$ must be a good response.
- The ERM algorithm fails to learn $\mathcal{R}$ according to Definition 1. Let $r^* = r_{\mathbf{0}}$ and $\pi_{\text{ref}}(y^{(0)}|x_i) = \pi_{\text{ref}}(y^{(1)}|x_i) = 1/2$ for any $i \in \mathbb{N}$. Indeed, for any finite sample size m, the worst-case ERM predicts a reward function $\hat{r}_m$ that satisfies $\hat{r}_m(x_i, y^{(1)}) = 1$ for any x_i that does not show up in

the training sample. Therefore, optimizing at test time using $\hat{r}_m$ predicts $y^{(1)}$ for a given x and is always wrong.

Then we show that the same example also refutes the conjecture that alignment is equivalent to multiclass classification. To see this, we first formally define the DS dimension, which characterizes the learnability of multiclass learning.

Definition 4 (**DS dimension** Daniely & Shalev-Shwartz (2014)). *Let $u_i = (x_i, y_i^{(0)}, y_i^{(1)})$. We say that a set $\{u_1, \ldots, u_n\}$ is DS-shattered by $\mathcal{A}(\mathcal{R})$, if there exists a finite and non-empty subset $\mathcal{A}'(\mathcal{R}) \subseteq \mathcal{A}(\mathcal{R})$, for every $A_r \in \mathcal{A}'(\mathcal{R})$ and for every $i \in [n]$, there exists $A_{r'} \in \mathcal{A}'(\mathcal{R})$ such that $A_{r'}(u_i) \neq A_r(u_i)$ and for every $j \neq i$, $A_{r'}(u_j) = A_r(u_j)$.*

The DS dimension of $\mathcal{A}(\mathcal{R})$ is defined as the largest $d \in \mathbb{N} \cup \{0\}$ such that there exists a set $\{u_1, \ldots, u_d\}$ DS-shattered by $\mathcal{A}(\mathcal{R})$. If no largest finite d exists, we say that $\mathcal{A}(\mathcal{R})$ has an infinite DS dimension.

Another natural approach is to consider the alignment problem as learning a set-valued multiclass classification with a finite label space of size 4, where each multiclass classifier $A_r \in \mathcal{A}(\mathcal{R}) : \mathcal{Z}_k \to \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$. This means, equivalently, that for multiple responses to each prompt, we learn exactly which of them are good. It implies that the characterization of the alignment learnability might be the DS dimension of $\mathcal{A}(\mathcal{R})$. However, for the reward class $\mathcal{R}$ in Example 2, its induced set-valued acceptance class $\mathcal{A}(\mathcal{R})$ has an infinite DS dimension since the (infinite) set $\{u_1, u_2, \ldots\}$ is DS-shattered by $\mathcal{A}(\mathcal{R})$. Therefore, the alignment problem also differs in nature from multiclass classification.

B.2 Learn-and-optimize principle fails

Example 6 (Restatement of Example 3). *Let $p \in \mathbb{N}$. Let $\{z_1 = (x_1, y^{(1)}, y^{(2)}, \ldots, y^{(2p+1)}), z_2 = (x_2, y^{(1)}, y^{(2)}, \ldots, y^{(2p+1)}), \ldots\}$ be an infinite space. Define $\mathcal{R} = \{r_{\boldsymbol{s}}, \forall \boldsymbol{s} \in [2p+1]^{\mathbb{N}}\}$, where for any $i \in \mathbb{N}$ and any $j \in [2p+1]$, $r_{\boldsymbol{s}}(x_i, y^{(j)}) = 1$, if $j \leq s_i$ and $r_{\boldsymbol{s}}(x_i, y^{(j)}) = 0$, if $j > s_i$.*

Let $\boldsymbol{s} = \{s_i\}_{i \in \mathbb{N}}$ such that s_i is i.i.d. sampled from $\mathrm{Unif}([2p])$, where $\mathrm{Unif}([2p])$ stands for the uniform distribution on $[2p]$. Let $r^* = r_{\boldsymbol{s}}$ and $\pi_{\mathrm{ref}}(y^{(s_i)}|x_i) = \pi_{\mathrm{ref}}(y^{(s_i+1)}|x_i) = 1/2$.

Indeed, we can show that having access to the test instance and a pool of $k = \Theta((1/\gamma)\log(1/\epsilon))$ candidate responses, a pessimism principle realizes the alignment learnability of the reward class defined in Example 3. To see this, note that when the candidate pool contains only one unique response, it must be a good one since π_{ref} is assumed to generate at least one good response within k times of sampling. When at least two distinct responses are observed, the pessimism principle will only assign reward value 1 to the response $y^{(j)}$ in the pool that has the smallest index $j \in [2p+1]$, and assign reward value 0 to all the other responses. In other words, only the smallest-indexed response, which is guaranteed to be good, will be accepted at test time.

B.3 Pessimism principle fails

As discussed before, a pessimism principle can be interpreted as running the so-called *Closure* algorithm on a test instance-dependent version space. We slightly extend the definition of the set-valued function A_r as $A_r : \mathcal{Z}_k \mapsto 2^{[k]}$ satisfying $A_r(z) = \{j \in [k] : r(x, y_j) = 1\}$ here. Let $S_m = \{Z_i, A_{r^*}(Z_i)\}_{i=1}^m$ denote the training data. We adapt the classical notion of version space Mitchell (1977) to define $V_{S_m}(\mathcal{R}) = \{r \in \mathcal{R} : A_r(Z_i) = A_{r^*}(Z_i), \forall i \in [m]\}$. Let $z = (x, y^{(1)}, \ldots, y^{(k)})$ be a test instance. Define the "test-aware version space" as $\hat{\mathcal{R}}_m = \{r \in V_{S_m}(\mathcal{R}) : \exists i \in [k] \text{ s.t. } i \in A_r(z)\}$. Now, we attribute the pessimism principle to running the Closure algorithm on $\hat{\mathcal{R}}_m$, i.e., to return a reward function that computes

the pointwise minimum reward value:

$$\hat{r}_{\text{closure}}(x, y) = \min\left\{r(x, y) : r \in \hat{\mathcal{R}}_m\right\}, \quad \forall (x, y) \in \mathcal{X} \times \mathcal{Y}.$$

Then, using $\hat{r}_{\text{closure}}$ to optimize and select at test time. Equivalently, the learning procedure selects, for the test instance $z = (x, y^{(1)}, \ldots, y^{(k)})$, a response that lies in the intersection of all acceptance sets induced by reward functions in $\hat{\mathcal{R}}_m$, that is, select $y^{(\hat{i}_m)}$ with $\hat{i}_m \in [k] \cap (\cap_{r \in \hat{\mathcal{R}}_m} A_r(z))$. Our Example 4, however, suggests that the pessimism principle does not always guarantee the alignment learnability since $\cap_{r \in \hat{\mathcal{R}}_m} A_r(z) = \emptyset$ might happen.

Example 7 (Restatement of Example 4)**.** *Let $\mathcal{X}$ be an infinite space and $|\mathcal{Y}| = 2$. Let $\{z_1 = (x_1, y^{(0)}, y^{(1)}), z_2 = (x_2, y^{(0)}, y^{(1)}), \ldots\}$ be an infinite space. Define $\mathcal{R} = \{r_{\boldsymbol{b},i}, \forall \boldsymbol{b} \in \{0,1\}^{\mathbb{N}}, \forall i \in \mathbb{N}\}$, where $r_{\boldsymbol{b},i}$ is defined according to*

$$\begin{cases} r_{\boldsymbol{b},i}(x_j, y^{(0)}) = 1, & r_{\boldsymbol{b},i}(x_j, y^{(1)}) = b_j, & j \neq i \\ r_{\boldsymbol{b},i}(x_j, y^{(0)}) = b_j, & r_{\boldsymbol{b},i}(x_j, y^{(1)}) = 1, & j = i \end{cases}, \quad \forall j \in \mathbb{N}.$$

As an extension of Example 2, Example 4 provides a reward class with an infinite VC dimension and an alignment dimension $d_{\text{ALN}}(\mathcal{R}) = 1$. Given a finite training sample, if the test instance does not show up in the training data and the candidate pool contains both $y^{(0)}$ and $y^{(1)}$, $\hat{\mathcal{R}}_m$ contains reward functions that realize both $(1, 0)$ and $(0, 1)$ patterns of rewards for $((x, y^{(0)}), (x, y^{(1)}))$. Therefore, the pessimism principle yields a reward function that assigns reward value 0 to both $(x, y^{(0)})$ and $(x, y^{(1)})$, and thus cannot return a good response at test time.

C Omitted Proofs

C.1 Upper Bound

In this section, we provide the detailed proofs of the upper bound results.

Lemma 3 (**Lemma 1 restated**)**.** *Let $\mathcal{H} \subseteq \{0, 1, *\}^{\mathcal{U}}$ be a partial concept class with $\text{VC}(\mathcal{H}) = d < \infty$, and let $h^* \in \mathcal{H}$. Suppose that $U_1, \ldots, U_m$ are generated i.i.d. from a distribution $\mathcal{D}$ over $\mathcal{U}$ and that the partial labels $h^*(U_1), \ldots, h^*(U_m)$ are observed. There exists a learning algorithm $\mathbb{A}_{\text{OIG}}$ which returns a total predictor $\hat{h}_m : \mathcal{U} \to \{0, 1\}$ such that, for any $\eta, \delta \in (0, 1)$,*

$$m \geq \frac{C}{\eta}\left[(d \vee 1)\log^2\left(\frac{e(d \vee 1)}{\eta}\right) + \log\left(\frac{2}{\delta}\right)\right] \tag{2}$$

for a universal constant $C > 0$ implies

$$\Pr_{U \sim \mathcal{D}}\left(h^*(U) \in \{0, 1\} \text{ and } \hat{h}_m(U) \neq h^*(U)\right) \leq \eta \tag{3}$$

with probability at least $1 - \delta$ over the training sample and the internal randomness of the learner. Here, $d \vee 1 = \max\{d, 1\}$.

Proof of Lemma 3. Let $\mu = \Pr_{U \sim \mathcal{D}}(h^*(U) \in \{0, 1\})$ be the probability of observing a non-$*$ label. Discard all examples whose labels equal $*$. Conditional on the number N of retained examples, the retained examples are generated i.i.d. from the distribution $\mathcal{D}$ conditioned on $h^*(U) \in \{0, 1\}$, and they are realizable by the partial class $\mathcal{H}$.

We run the one-inclusion graph learner followed by the boosting and sample-compression procedure of Alon et al. (2021) on these N retained examples. Conditional on $N = n \geq 1$, its conditional error is at most

$$C_0 \frac{(d \vee 1) \log^2((d \vee 1)n) + \log(2/\delta)}{n} \tag{4}$$

with probability at least $1 - \delta/2$, where $C_0 > 0$ is a universal constant. When $N = 0$, the learner returns an arbitrary binary predictor.

If $\mu \leq \eta$, then the left-hand side of Equation (3) is at most $\mu \leq \eta$ for every predictor. Suppose next that $\mu > \eta$. Since $N \sim \mathrm{Bin}(m, \mu)$, a multiplicative Chernoff bound gives that

$$\Pr\left(N < \frac{m\mu}{2}\right) \leq \exp\left(-\frac{m\mu}{8}\right).$$

By the sample-size assumption, choosing $C \geq 8$ yields

$$m\mu \geq m\eta \geq C \log\left(\frac{2}{\delta}\right) \geq 8 \log\left(\frac{2}{\delta}\right).$$

Consequently,

$$\Pr\left(N < \frac{m\mu}{2}\right) \leq \exp\left(-\log\left(\frac{2}{\delta}\right)\right) = \frac{\delta}{2}.$$

On the event $N \geq m\mu/2$, multiplying the conditional error in Equation (4) by μ shows that the error under the ambient distribution is at most

$$C_0\mu \frac{(d \vee 1) \log^2((d \vee 1)m) + \log(2/\delta)}{N} \leq \frac{2C_0}{m} \left((d \vee 1) \log^2((d \vee 1)m) + \log(2/\delta)\right).$$

The choice of m in Equation (2), with a sufficiently large universal constant, makes the last display at most η. Taking a union bound over the Chernoff event and the failure event of the partial-class learner proves the lemma. □

Theorem 4 (Theorem 2 restated). *Suppose that $d_{\mathrm{ALN}}(\mathcal{R}) = d < \infty$. For any $\gamma, \epsilon, \delta \in (0, 1)$, choose*

$$k \geq \max\left\{2, \left\lceil \frac{1}{\gamma} \log\left(\frac{2}{\epsilon}\right) \right\rceil\right\}. \tag{5}$$

There exists a universal constant $C > 0$ such that the pairwise tournament procedure satisfies

$$\Pr_{Z \sim \mathcal{D}_k}\left(r^*(X, Y^{(\hat{i}_m(Z))}) = 1\right) \geq 1 - \epsilon$$

with probability at least $1 - \delta$, provided that

$$m \geq C \frac{k^2}{\epsilon}\left((d \vee 1) \log^2\left(\frac{(d \vee 1)k}{\epsilon}\right) + \log\left(\frac{1}{\delta}\right)\right). \tag{6}$$

Proof of Theorem 4. Set $\eta = \epsilon/(k(k-1))$. By Definition 2, the partial comparison class $\mathcal{G}(\mathcal{R})$ has VC dimension d. Hence, Lemma 1 and Equation (6) imply that, with probability at least $1 - \delta$ over the training sample,

$$\Pr_{U \sim \mathcal{D}_{\mathrm{pair}}}\left(g^*(U) \in \{0, 1\} \text{ and } \hat{g}_m(U) \neq g^*(U)\right) \leq \eta. \tag{7}$$

Condition on a training sample for which Equation (7) holds. For a fresh test data, every ordered pair $(X, Y^{(a)}, Y^{(b)})$, with $a < b$, has marginal distribution $\mathcal{D}_{\mathrm{pair}}$. Let event $\mathcal{E}$ be that some pair contains exactly one good response and the algorithm selects incorrectly. A union bound over the $\binom{k}{2}$ pairs yields

$$\Pr_{Z \sim \mathcal{D}_k}(\mathcal{E}) \leq \binom{k}{2}\eta = \frac{\epsilon}{2}. \tag{8}$$

Moreover, the assumption of π_{ref} and the conditional independence of the test responses imply

$$\Pr_{Z\sim\mathcal{D}_k}\left(r^*(X,Y^{(j)})=0,\quad \forall j\in[k]\right)\le(1-\gamma)^k\le e^{-\gamma k}\le\frac{\epsilon}{2}, \tag{9}$$

where the last inequality follows from Equation (5). Whenever the test data contains a good response and no mixed pair is classified incorrectly, Lemma 2 guarantees that the returned response is good. Combining Equation (8) and Equation (9) gives

$$\Pr_{Z\sim\mathcal{D}_k}\left(r^*(X,Y^{(\hat{i}_m(Z))})=0\right)\le\epsilon.$$

This proves the result. □

C.2 Lower Bound

In this section, we provide the detailed proof of the lower bound. We first prove the following auxiliary lemma.

Lemma 4. *Suppose that $u_1,\dots,u_d$ are alignment shattered by $\mathcal{R}$, where $u_i=(x_i,y_i^{(0)},y_i^{(1)})$. If $i\neq j$ and $x_i=x_j$, then $\{y_i^{(0)},y_i^{(1)}\}\cap\{y_j^{(0)},y_j^{(1)}\}=\emptyset$.*

Proof of Lemma 4. Suppose for contradiction that $y_i^{(a)}=y_j^{(c)}$ for some $a,c\in\{0,1\}$. By alignment shattering, there exists a reward function corresponding to a boolean vector satisfying $b_i=a$ and $b_j=1-c$. The shattering conditions would require the shared response to have reward 1 through the i-th pair and reward 0 through the j-th pair, which is impossible. □

Then we can start to prove the lower bound, which is restated below.

Theorem 5 (Theorem 3 restated). *Suppose that $d_{\text{ALN}}(\mathcal{R})\ge d$. Fix any learning algorithm, any training sample size $m\in\mathbb{N}$, and any test data with k responses, where $k\in\mathbb{N}$. If $d\ge 8mk$, there exist a target reward function $r^*\in\mathcal{R}$, a distribution $\mathcal{D}_X$, and a reference policy π_{ref} satisfying the assumption of π_{ref} with $\gamma=1/2$, such that*

$$\Pr_{S_m}\left(\Pr_{Z\sim\mathcal{D}_k}\left(\hat{i}_m(Z)\notin A_{r^*}(Z)\right)\ge\frac{1}{8}\right)\ge\frac{2}{7}.$$

Proof of Theorem 5. For $i\in[d]$, let $u_i=(x_i,y_i^{(0)},y_i^{(1)})$ be alignment shattered by $\mathcal{R}$. For every prompt x appearing among $x_1,\dots,x_d$, define $I_x=\{i\in[d]:x_i=x\}$, and $n_x=|I_x|$. By Lemma 4, the response pairs indexed by I_x are disjoint. Define $\mathcal{D}_X(x)=n_x/d$ for every prompt appearing in the shattered family. Conditional on such a prompt x, define the reference policy by

$$\pi_{\text{ref}}\left(y_i^{(0)}\mid x\right)=\pi_{\text{ref}}\left(y_i^{(1)}\mid x\right)=\frac{1}{2n_x},\qquad i\in I_x. \tag{10}$$

Choose a boolean vector $\boldsymbol{b}$ uniformly from $\{0,1\}^d$, and let $r_{\boldsymbol{b}}\in\mathcal{R}$ be a reward function witnessing the corresponding preferences. For every prompt in the support of $\mathcal{D}_X$, exactly one response of each pair has reward 1. Consequently, $\Pr_{Y\sim\pi_{\text{ref}}(\cdot|x)}(r_{\boldsymbol{b}}(x,Y)=1)=1/2$, so the assumption of π_{ref} holds with $\gamma=1/2$.

For a training sample S_m, let $J(S_m)\subseteq[d]$ denote the indices of shattered tuples for which at least one response (among two) is observed in the training sample, i.e.,

$$J(S_m):=\left\{i\in[d]:\exists\, t\in[m]\ \text{ such that }\ X_t=x_i\ \text{ and }\ \{Y_t^{(0)},Y_t^{(1)}\}\cap\{y_i^{(0)},y_i^{(1)}\}\neq\emptyset\right\}.$$

Clearly, $|J(S_m)| \le 2m$. For every prompt x, let $s_x = |J(S_m) \cap I_x|$. Conditional on $X = x$, every test response first chooses an index uniformly from I_x, and then chooses one of the two responses uniformly.

For notational simplicity, for each $i \in [d]$, define

$$P_i := \{(x_i, y_i^{(0)}), (x_i, y_i^{(1)})\}.$$

Then, conditional on S_m, we have

$$\begin{aligned}
\Pr\left((X, Y^{(j)}) \notin \bigcup_{i \in J(S_m)} P_i, \ \ \forall j \in [k] \,\middle|\, S_m\right) &= \sum_{x: n_x > 0} \frac{n_x}{d} \left(1 - \frac{s_x}{n_x}\right)^k \\
&\ge \sum_{x: n_x > 0} \frac{n_x}{d} \left(1 - \frac{k s_x}{n_x}\right) \\
&= 1 - \frac{k}{d} \sum_x s_x \\
&= 1 - \frac{k |J(S_m)|}{d} \\
&\ge 1 - \frac{2mk}{d} \\
&\ge \frac{3}{4},
\end{aligned}$$

where the first inequality uses $(1-t)^k \ge 1 - kt$ for $t \in [0, 1]$.

On this event, whichever candidate the learner selects belongs to a pair whose preference bit b_i is not revealed by the training sample. Under the uniformly random choice of $\boldsymbol{b}$, that preference bit is independent of the training sample, the test data, and the learner's selection rule. Therefore, the selected response has reward 0 with probability $1/2$. It follows that

$$\mathbb{E}_{\boldsymbol{b}, S_m}\left[\Pr_{Z \sim \mathcal{D}_k}\left(r^*(X, Y^{(\hat{i}_m(Z))}) = 0\right)\right] \ge \frac{1}{2} \cdot \frac{3}{4} = \frac{3}{8}.$$

Hence, there exists a deterministic $\boldsymbol{b}$ such that, for $r^* = r_{\boldsymbol{b}}$,

$$\mathbb{E}_{S_m}\left[\Pr_{Z \sim \mathcal{D}_k}\left(r^*(X, Y^{(\hat{i}_m(Z))}) = 0\right)\right] \ge \frac{3}{8}.$$

Let $R(S_m) = \Pr_{Z \sim \mathcal{D}_k}(r^*(X, Y^{(\hat{i}_m(Z))}) = 0)$, we have

$$\frac{3}{8} \le \mathbb{E}[R(S_m)] \le \frac{1}{8} \Pr\left(R(S_m) < \frac{1}{8}\right) + \Pr\left(R(S_m) \ge \frac{1}{8}\right),$$

Therefore,

$$\Pr\left(R(S_m) \ge \frac{1}{8}\right) \ge \frac{\frac{3}{8} - \frac{1}{8}}{1 - \frac{1}{8}} = \frac{2}{7}.$$

This completes the proof. □

D Detailed Discussion

Besides the discussion we have made in Section 5, the following problems are more closely related to our formulation.

D.1 Learning to predict rather than to select

Another constraint of our formulation is that the learner is expected to select a good response from multiple candidates. However, practical alignment methods, especially RLHF methods, instead seek to predict a good response for a given test prompt.[6] More formally, under the setting of Section 2, the goal is then to design a learning algorithm $\mathcal{A}$ that takes S_m as input and outputs a predictor $\mathcal{A}(S_m) : \mathcal{X} \to \mathcal{Y}$ that, given a test instance $X \sim \mathcal{D}_X$, predicts a response $\hat{Y} = \mathcal{A}(S_m)(X) \in \mathcal{Y}$ such that $r^*(X, \hat{Y}) = 1$ with high probability.

While it might be interesting to have a theoretical understanding of this learning model, it is worth mentioning that this formulation mismatches practical alignment in a sense that, the learner's prediction is irrelevant to the reference policy π_{ref}. However, for practical alignment such as RLHF approaches, a regularization of KL-divergence $\mathbb{D}_{\text{KL}}(\cdot||\pi_{\text{ref}})$ is usually appended to a policy optimization objective to ensure that the learned policy (predictor) stays close to the reference policy and thus may inherit nice properties from it such as linguistic fluency and broad knowledge. It is also for this reason that we study selection-based prediction in this work.

D.2 Learning with preferences

A more significant and challenging extension is to study exactly alignment with preferences. Specifically, consider the setting of real-valued rewards, for example, $\mathcal{R} \subseteq [0, 1]^{\mathcal{X}\times\mathcal{Y}}$. Every training sample consists of a prompt $x \in \mathcal{X}$, two independent responses $y^{(0)}, y^{(1)}$ generated from the reference policy $\pi_{\text{ref}}(\cdot|x)$, and a binary preference label $\sigma = \mathbb{1}[y^{(0)} \succ y^{(1)}]$ indicating which response is more preferred according to the true reward function $r^* \in \mathcal{R}$. This setting could be considered as an offline variant of dueling bandits. Given i.i.d. samples $S_m = \{(x_i, y_i^{(0)}, y_i^{(1)}, \sigma_i)\}_{i=1}^m$, the learner $\mathbf{A}$ is expected to predict a response $\mathbf{A}(S_m)(x) \in \mathcal{Y}$ on a test prompt x such that, with probability at least $1 - \delta$,

$$\Pr_{x\sim\mathcal{D}_X, y'\sim\pi_{\text{ref}}(\cdot|x)} (\mathbf{A}(S_m)(x) \succ y') \geq 1 - \epsilon.$$

The challenge arises in how to measure the quality of the reference policy. Huang et al. (2025a) made an assumption on the reference policy's coverage. They defined the coverage coefficient as

$$\mathcal{C}^{\pi^*}(x) = \mathbb{E}_{y\sim\pi^*(\cdot|x)} \left[\frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} \right]$$

where π^* is a comparator policy that achieves high true rewards, and it is assumed that $\mathcal{C}^{\pi^*}(x) \leq C$ for some constant $C > 0$ holds uniformly for any x. Sriraman & Block (2026) proposed another measure of discrepancy between π_{ref} and π^* called the $\mathcal{E}_M$-divergence. Zhao et al. (2025) studied other coverage conditions including the Global-Policy Coverage and the Local KL-ball Coverage. Note that such assumptions are usually formulated as discrepancy measures between π_{ref} and a comparator policy. Instead, can we model the reference policy assumption to rely only on parameters γ, ϵ, and δ?

[6] Note that this does not necessarily require learning a policy from a hypothesis policy class.