# Strategic Classification under Unknown Personalized Manipulation

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

We study the fundamental mistake bound and sample complexity in the strategic classification, where agents can strategically manipulate their feature vector up to an extent in order to be predicted as positive. For example, given a classifier determining college admission, student candidates may try to take easier classes to improve their GPA, retake SAT and change schools in an effort to fool the classifier. *Ball manipulations* are a widely studied class of manipulations in the literature, where agents can modify their feature vector within a bounded radius ball. Unlike most prior work, our work consider manipulations to be *personalized*, meaning that agents can have different levels of manipulation abilities (e.g., varying radii for ball manipulations), and *unknown* to the learner.

We formalize the learning problem in an interaction model where the learner first deploys a classifier and the agent manipulates the feature vector within their manipulation set to game the deployed classifier. We investigate various scenarios in terms of the information available to the learner during the interaction, such as observing the original feature vector before or after deployment, observing the manipulated feature vector, or not seeing either the original or the manipulated feature vector. We begin by providing online mistake bounds and PAC sample complexity in these scenarios for ball manipulations. We also explore non-ball manipulations and show that, even in the simplest scenario where both the original and the manipulated feature vectors are revealed, the mistake bounds and sample complexity are lower bounded by $\Omega(|\mathcal{H}|)$ when the target function belongs to a known class $\mathcal{H}$.

## 1 Introduction

Strategic classification addresses the the problem of learning a classifier robust to manipulation and gaming by self-interested agents (Hardt et al., 2016). For example, given a classifier determining loan approval based on credit scores, applicants could open or close credit cards and bank accounts to increase their credit scores. In the case of a college admission classifier, students may try to take easier classes to improve their GPA, retake the SAT or change schools in an effort to be admitted. In both cases, such manipulations do not change their true qualifications. Recently, a collection of papers has studied strategic classification in both the online setting where examples are chosen by an adversary in a sequential manner (Dong et al., 2018; Chen et al., 2020; Ahmadi et al., 2021, 2023), and the distributional setting where the examples are drawn from an underlying data distribution (Hardt et al., 2016; Zhang and Conitzer, 2021; Sundaram et al., 2021; Lechner and Urner, 2022). Most existing works assume that manipulation ability is uniform across all agents or is known to the learner. However, in reality, this may not always be the case. For instance, low-income students may have a lower ability to manipulate the system compared to their wealthier peers due to factors such as the high costs of retaking the SAT or enrolling in additional classes, as well as facing more barriers to

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

accessing information about college (Milli et al., 2019) and it is impossible for the learner to know the highest achievable GPA or the maximum number of times a student may retake the SAT due to external factors such as socio-economic background and personal circumstances.

We characterize the manipulation of an agent by a set of alternative feature vectors that she can modify her original feature vector to, which we refer to as the *manipulation set*. *Ball manipulations* are a widely studied class of manipulations in the literature, where agents can modify their feature vector within a bounded radius ball. For example, Dong et al. (2018); Chen et al. (2020); Sundaram et al. (2021) studied ball manipulations with distance function being some norm and Zhang and Conitzer (2021); Lechner and Urner (2022); Ahmadi et al. (2023) studied a manipulation graph setting, which can be viewed as ball manipulation w.r.t. the graph distance on a predefined known graph.

In the online learning setting, the strategic agents come sequentially and try to game the current classifier. Following previous work, we model the learning process as a repeated Stackelberg game over $T$ time steps. In round $t$, the learner proposes a classifier $f_t$ and then the agent, with a manipulation set (unknown to the learner), manipulates her feature in an effort to receive positive prediction from $f_t$. There are several settings based on what and when the information is revealed about the original feature vector and the manipulated feature vector in the game. The simplest setting for the learner is observing the original feature vector before choosing $f_t$ and the manipulated vector after. In a slightly harder setting, the learner observes both the original and manipulated vectors after selecting $f_t$. An even harder setting involves observing only the manipulated feature vector after selecting $f_t$. The hardest and least informative scenario occurs when neither the original nor the manipulated feature vectors are observed.

In the distributional setting, the agents are sampled from an underlying data distribution. Previous work assumes that the learner has full knowledge of the original feature vector and the manipulation set, and then views learning as a one-shot game and solves it by computing the Stackelberg equilibria of it. However, when manipulations are personalized and unknown, we cannot compute an equilibrium and study learning as a one-shot game. In this work, we extend the iterative online interaction model from the online setting to the distributional setting, where the sequence of agents is sampled i.i.d. from the data distribution. After repeated learning for $T$ (which is equal to the sample size) rounds, the learner has to output a strategy-robust predictor for future use.

In both online and distributional settings, examples are viewed through the lens of the current predictor and the learner does not have the ability to inquire about the strategies the previous examples would have adopted under a different predictor.

**Related work** Our work is primarily related to strategic classification in online and distributional settings. Strategic classification was first studied in a distributional model by Hardt et al. (2016) and subsequently by Dong et al. (2018) in an online model. Hardt et al. (2016) assumed that agents manipulate by best response with respect to a uniform cost function known to the learner. Building on the framework of (Hardt et al., 2016), Lechner and Urner (2022); Sundaram et al. (2021); Zhang and Conitzer (2021); Hu et al. (2019); Milli et al. (2019) studied the distributional learning problem, and all of them assumed that the manipulations are predefined and known to the learner, either by a cost function or a predefined manipulation graph. For online learning, Dong et al. (2018) considered a similar manipulation setting as in this work, where manipulations are personalized and unknown. However, they studied linear classification with ball manipulations in the online setting and focused on finding appropriate conditions of the cost function to achieve sub-linear Stackelberg regret. Chen et al. (2020) also studied Stackelberg regret in linear classification with uniform ball manipulations. Ahmadi et al. (2021) studied the mistake bound under uniform (possbily unknown) ball manipulations, and Ahmadi et al. (2023) studied regret under a pre-defined and known manipulation. We postpone the discussion of studies on other objectives and models in the strategic setting to the Appendix A.

## 2 Model

**Strategic classification** Throughout this work, we consider the binary classification task. Let $\mathcal{X}$ denote the feature vector space, $\mathcal{Y} = \{+1, -1\}$ denote the label space, and $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ denote the hypothesis class. In the strategic setting, instead of an example being a pair $(x, y)$, an example, or *agent*, is a triple $(x, u, y)$ where $x \in \mathcal{X}$ is the original feature vector, $y \in \mathcal{Y}$ is the label, and $u \subseteq \mathcal{X}$ is the manipulation set, which is a set of feature vectors that the agent can modify their original feature vector $x$ to. In particular, given a hypothesis $h \in \mathcal{Y}^{\mathcal{X}}$, the agent will try to manipulate her feature vector $x$ to another feature vector $x'$ within $u$ in order to receive a positive prediction from $h$. The manipulation set $u$ is *unknown* to the learner. In this work, we will be considering several settings based on what the information is revealed to the learner, including both the original/manipulated feature vectors, the manipulated feature vector only, or neither, and when the information is revealed.

More formally, for agent $(x, u, y)$, given a predictor $h$, if $h(x) = -1$ and her manipulation set overlaps the positive region by $h$, i.e., $u \cap \mathcal{X}_{h,+} \neq \emptyset$ with $\mathcal{X}_{h,+} := \{x \in \mathcal{X} | h(x) = +1\}$, the agent will manipulate $x$ to $\Delta(x, h, u) \in u \cap \mathcal{X}_{h,+}$[1] to receive positive prediction by $h$. Otherwise, the agent will do nothing and maintain her feature vector at $x$, i.e., $\Delta(x, h, u) = x$. We call $\Delta(x, h, u)$ the manipulated feature vector of agent $(x, u, y)$ under predictor $h$.

A general and fundamental type of manipulations is *ball manipulations*, where agents can manipulate their feature within a ball of *personalized* radius. More specifically, given a metric $d$ over $\mathcal{X}$, the manipulation set is a ball $\mathcal{B}(x; r) = \{x' | d(x, x') \leq r\}$ centered at $x$ with radius $r$ for some $r \in \mathbb{R}_{\geq 0}$. Note that we allow different agents to have different manipulation power and the radius can vary over agents. Let $Q$ denote the set of allowed pairs $(x, u)$, which we refer to as the feature-manipulation set space. For ball manipulations, we have $\mathcal{Q} = \{(x, \mathcal{B}(x; r)) | x \in \mathcal{X}, r \in \mathbb{R}_{\geq 0}\}$ for some known metric $d$ over $\mathcal{X}$. In the context of ball manipulations, we use $(x, r, y)$ to represent $(x, \mathcal{B}(x; r), y)$ and $\Delta(x, h, r)$ to represent $\Delta(x, h, \mathcal{B}(x; r))$ for notation simplicity.

For any hypothesis $h$, let the strategic loss $\ell^{\text{str}}(h, (x, u, y))$ of $h$ be defined as the loss at the manipulated feature, i.e., $\ell^{\text{str}}(h, (x, u, y)) := \mathbb{1}(h(\Delta(x, h, u)) \neq y)$. According to our definition of $\Delta(\cdot)$, we can write down the strategic loss explicitly as

$$\ell^{\text{str}}(h, (x, u, y)) = \begin{cases} 1 & \text{if } y = -1, h(x) = +1 \\ 1 & \text{if } y = -1, h(x) = -1 \text{ and } u \cap \mathcal{X}_{h,+} \neq \emptyset\,, \\ 1 & \text{if } y = +1, h(x) = -1 \text{ and } u \cap \mathcal{X}_{h,+} = \emptyset\,, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

For any randomized predictor $p$ (a distribution over hypotheses), the strategic behavior depends on the realization of the predictor and the strategic loss of $p$ is $\ell^{\text{str}}(p, (x, u, y)) := \mathbb{E}_{h \sim p}\left[\ell^{\text{str}}(h, (x, u, y))\right]$.

**Online learning** We consider the task of sequential classification where the learner aims to classify a sequence of agents $(x_1, u_1, y_1), (x_2, u_2, y_2), \ldots, (x_T, u_T, y_T) \in \mathcal{Q} \times \mathcal{Y}$ that arrives in an online manner. At each round, the learner feeds a predictor to the environment and then observes his prediction $\widehat{y}_t$, the true feature $y_t$ and possibly along with some additional information about the original/manipulated feature vectors. We say the learner makes a mistake at round $t$ if $\widehat{y}_t \neq y_t$ and the learner's goal is to minimize the number of mistakes on the sequence. The interaction protocol (which repeats for $t = 1, \ldots, T$) is described in the following.

**Protocol 1** Learner-Agent Interaction at round $t$

1: The environment picks an agent $(x_t, u_t, y_t)$ and reveals some context $C(x_t)$. In the online setting, the agent is chosen adversarially, while in the distributional setting, the agent is sampled i.i.d.
2: The learner $\mathcal{A}$ observes $C(x_t)$ and picks a hypothesis $f_t \in \mathcal{Y}^{\mathcal{X}}$.
3: The learner $\mathcal{A}$ observes the true label $y_t$, the prediction $\widehat{y}_t = f_t(\Delta_t)$, and some feedback $F(x_t, \Delta_t)$, where $\Delta_t = \Delta(x_t, f_t, u_t)$ is the manipulated feature vector.

The context function $C(\cdot)$ and feedback function $F(\cdot)$ reveals information about the original feature vector $x_t$ and the manipulated feature vector $\Delta_t$. $C(\cdot)$ reveals the information before the learner picks $f_t$ while $F(\cdot)$ does after. We study several different settings based on what and when information is revealed.

- The simplest setting for the learner is observing the original feature vector $x_t$ before choosing $f_t$ and the manipulated vector $\Delta_t$ after. This setting corresponds to $C(x_t) = x_t$ and $F(x_t, \Delta_t) = \Delta_t$. We denote a setting by their values of $C, F$ and thus, we denote this setting by $(x, \Delta)$.
- In a slightly harder setting, the learner observes both the original and manipulated vectors after selecting $f_t$ and thus, $f_t$ cannot depend on the original feature vector in this case. Then $C(x_t) = \perp$ and $F(x_t, \Delta_t) = (x_t, \Delta_t)$, where $\perp$ is a token for "no information", and this setting is denoted by $(\perp, (x, \Delta))$.

[1]For ball manipulations, agents break ties by selecting the closest one. For non-ball manipulations, agents break ties randomly.

- An even harder setting involves observing only the manipulated feature vector after selecting $f_t$ (which can only be revealed after $f_t$ since $\Delta_t$ depends on $f_t$). Then $C(x_t) = \perp$ and $F(x_t, \Delta_t) = \Delta_t$ and this setting is denoted by $(\perp, \Delta)$.
- The hardest and least informative scenario occurs when neither the original nor the manipulated feature vectors are observed. Then $C(x_t) = \perp$ and $F(x_t, \Delta_t) = \perp$ and it is denoted by $(\perp, \perp)$.

Throughout this work, we focus on the *realizable* setting, where there exists a perfect classifier in $\mathcal{H}$ that never makes any mistake at the sequence of strategic agents. More specifically, there exists a hypothesis $h^* \in \mathcal{H}$ such that for any $t \in [T]$, we have $y_t = h^*(\Delta(x_t, h^*, u_t))$. Then we define the mistake bound as follows.

**Definition 1.** *For any choice of $(C, F)$, let $\mathcal{A}$ be an online learning algorithm under Protocol 1 in the setting of $(C, F)$. Given any realizable sequence $S = (x_1, u_1, h^*(\Delta(x_1, h^*, u_1))), \ldots, (x_T, u_T, h^*(\Delta(x_T, h^*, u_T))) \in (\mathcal{Q} \times \mathcal{Y})^T$, where $T$ is any integer and $h^* \in \mathcal{H}$, let $\mathcal{M}_{\mathcal{A}}(S)$ be the number of mistakes $\mathcal{A}$ makes on the sequence $S$. The mistake bound of $(\mathcal{H}, \mathcal{Q})$, denoted $\mathrm{MB}_{C,F}$, is the smallest number $B \in \mathbb{N}$ such that there exists an algorithm $\mathcal{A}$ such that $\mathcal{M}_{\mathcal{A}}(S) \leq B$ over all realizable sequences $S$ of the above form.*

According the rank of difficulty of the four settings with different choices of $(C, F)$, the mistake bounds are ranked in the order of $\mathrm{MB}_{x,\Delta} \leq \mathrm{MB}_{\perp,(x,\Delta)} \leq \mathrm{MB}_{\perp,\Delta} \leq \mathrm{MB}_{\perp,\perp}$.

**PAC learning** In the distributional setting, the agents are sampled from an underlying distribution $\mathcal{D}$ over $\mathcal{Q} \times \mathcal{Y}$. The learner's goal is to find a hypothesis $h$ with low population loss $\mathcal{L}^{\text{str}}_{\mathcal{D}}(h) := \mathbb{E}_{(x,u,y)\sim\mathcal{D}}[\ell^{\text{str}}(h, (x, u, y))]$. One may think of running empirical risk minimizer (ERM) over samples drawn from the underlying data distribution, i.e., returning $\arg\min_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \ell^{\text{str}}(h, (x_i, u_i, y_i))$, where $(x_1, u_1, y_1), \ldots, (x_m, u_m, y_m)$ are i.i.d. sampled from $\mathcal{D}$. However, ERM is unimplementable because the manipulation sets $u_i$'s are never revealed to the algorithm, and only the partial feedback in response to the implemented classifier is provided. In particular, in this work we consider using the same interaction protocol as in the online setting, i.e., Protocol 1, with agents $(x_t, u_t, y_t)$ i.i.d. sampled from the data distribution $\mathcal{D}$. After $T$ rounds of interaction (i.e., $T$ i.i.d. agents), the learner has to output a predictor $f_{\text{out}}$ for future use.

Again, we focus on the *realizable* setting, where the sequence of sampled agents (with manipulation) can be perfectly classified by a target function in $\mathcal{H}$. Alternatively, there exists a classifier with zero population loss, i.e., there exists a hypothesis $h^* \in \mathcal{H}$ such that $\mathcal{L}^{\text{str}}_{\mathcal{D}}(h^*) = 0$. Then we formalize the notion of PAC sample complexity under strategic behavior as follows.

**Definition 2.** *For any choice of $(C, F)$, let $\mathcal{A}$ be a learning algorithm that interacts with agents using Protocol 1 in the setting of $(C, F)$ and outputs a predictor $f_{\text{out}}$ in the end. For any $\varepsilon, \delta \in (0, 1)$, the sample complexity of realizable $(\varepsilon, \delta)$-PAC learning of $(\mathcal{H}, \mathcal{Q})$, denoted $\mathrm{SC}_{C,F}(\varepsilon, \delta)$, is defined as the smallest $m \in \mathbb{N}$ for which there exists a learning algorithm $\mathcal{A}$ in the above form such that for any distribution $\mathcal{D}$ over $\mathcal{Q} \times \mathcal{Y}$ where there exists a predictor $h^* \in \mathcal{H}$ with zero loss, $\mathcal{L}^{str}_{\mathcal{D}}(h) = 0$, with probability at least $1 - \delta$ over $(x_1, u_1, y_1), \ldots, (x_m, u_m, y_m) \overset{i.i.d.}{\sim} \mathcal{D}$, $\mathcal{L}^{str}_{\mathcal{D}}(f_{\text{out}}) \leq \varepsilon$.*

Similar to mistake bounds, the sample complexities are ranked in the same order $\mathrm{SC}_{x,\Delta} \leq \mathrm{SC}_{\perp,(x,\Delta)} \leq \mathrm{SC}_{\perp,\Delta} \leq \mathrm{SC}_{\perp,\perp}$ according the rank of difficulty of the four settings.

# 3 Overview of Results

In classic (non-strategic) online learning, the Halving algorithm achieves a mistake bound of $\log(|\mathcal{H}|)$ by employing the majority vote and eliminating inconsistent hypotheses at each round. In classic PAC learning, the sample complexity of $\mathcal{O}(\frac{\log(|\mathcal{H}|)}{\varepsilon})$ is achievable via ERM. Both mistake bound and sample complexity exhibit logarithmic dependency on $|\mathcal{H}|$. This logarithmic dependency on $|\mathcal{H}|$ (when there is no further structural assumptions) is tight in both settings, i.e., there exist examples of $\mathcal{H}$ with mistake bound of $\Omega(\log(|\mathcal{H}|))$ and with sample complexity of $\Omega(\frac{\log(|\mathcal{H}|)}{\varepsilon})$. In the setting where manipulation is known beforehand and only $\Delta_t$ is observed, Ahmadi et al. (2023) proved a lower bound of $\Omega(|\mathcal{H}|)$ for the mistake bound. Since in the strategic setting we can achieve a linear dependency on $|\mathcal{H}|$ by trying each hypothesis in $\mathcal{H}$ one by one and discarding it once it makes a mistake, the question arises:

***Can we achieve a logarithmic dependency on $|\mathcal{H}|$ in strategic classification?***

In this work, we show that the dependency on $|\mathcal{H}|$ varies across different settings and that in some settings mistake bound and PAC sample complexity can exhibit different dependencies on $|\mathcal{H}|$. We start by presenting our results for ball manipulations in the four settings.

- Setting of $(x, \Delta)$ (observing $x_t$ before choosing $f_t$ and observing $\Delta_t$ after) : For online learning, we propose an variant of the Halving algorithm, called Strategic Halving (Algorithm 1), which can eliminate half of the remaining hypotheses when making a mistake. The algorithm depends on observing $x_t$ before choosing the predictor $f_t$. Then by applying the standard technique of converting mistake bound to PAC bound, we are able to achieve sample complexity of $\mathcal{O}(\frac{\log(|\mathcal{H}|)\operatorname{loglog}(|\mathcal{H}|)}{\varepsilon})$.
- Setting of $(\perp, (x, \Delta))$ (observing both $x_t$ and $\Delta_t$ after selecting $f_t$) : We prove that, there exists an example of $(\mathcal{H}, \mathcal{Q})$ s.t. the mistake bound is lower bounded by $\Omega(|\mathcal{H}|)$. This implies that no algorithm can perform significantly better than sequentially trying each hypothesis, which would make at most $|\mathcal{H}|$ mistakes before finding the correct hypothesis. However, unlike the construction of mistake lower bounds in classic online learning, where all mistakes can be forced to occur in the initial rounds, we demonstrate that we require $\Theta(|\mathcal{H}|^2)$ rounds to ensure that all mistakes occur. In the PAC setting, we first show that, any learning algorithm with proper output $f_{\text{out}}$, i.e., $f_{\text{out}} \in \mathcal{H}$, needs a sample size of $\Omega(\frac{|\mathcal{H}|}{\varepsilon})$. We can achieve a sample complexity of $O(\frac{\log^2(|\mathcal{H}|)}{\varepsilon})$ by executing Algorithm 2, which is a randomized algorithm with improper output.
- Setting of $(\perp, \Delta)$ (observing only $\Delta_t$ after selecting $f_t$) : The mistake bound of $\Omega(|\mathcal{H}|)$ also holds in this setting, as it is known to be harder than the previous setting. For the PAC learning, we show that any conservative algorithm, which only depends on the information from the mistake rounds, requires $\Omega(\frac{|\mathcal{H}|}{\varepsilon})$ samples. The optimal sample complexity is left as an open problem.
- Setting of $(\perp, \perp)$ (observing neither $x_t$ nor $\Delta_t$) : Similarly, the mistake bound of $\Omega(|\mathcal{H}|)$ still holds. For the PAC learning, we show that the sample complexity is $\Omega(\frac{|\mathcal{H}|}{\varepsilon})$ by reducing the problem to a stochastic linear bandit problem.

Then we move on to non-ball manipulations. However, we show that even in the simplest setting of observing $x_t$ before choosing $f_t$ and observing $\Delta_t$ after, there is an example of $(\mathcal{H}, \mathcal{Q})$ such that the sample complexity is $\widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$. This implies that in all four settings of different revealed information, we will have sample complexity of $\widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$ and mistake bound of $\widetilde{\Omega}(|\mathcal{H}|)$. We summarize our results in Table 1.

| | setting | mistake bound | sample complexity |
|---|---|---|---|
| ball | $(x, \Delta)$ | $\Theta(\log(\lvert\mathcal{H}\rvert))$ (Thm 1) | $\widetilde{\mathcal{O}}(\frac{\log(\lvert\mathcal{H}\rvert)}{\varepsilon})^a$ (Thm 2), $\Omega(\frac{\log(\lvert\mathcal{H}\rvert)}{\varepsilon})$ |
| | $(\perp, (x, \Delta))$ | $\mathcal{O}(\min(\sqrt{\log(\lvert\mathcal{H}\rvert)T}, \lvert\mathcal{H}\rvert))$ (Thm 4)<br>$\Omega(\min(\frac{T}{\lvert\mathcal{H}\rvert\log(\lvert\mathcal{H}\rvert)}, \lvert\mathcal{H}\rvert))$(Thm 3) | $\mathcal{O}(\frac{\log^2(\lvert\mathcal{H}\rvert)}{\varepsilon})$ (Thm 6), $\Omega(\frac{\log(\lvert\mathcal{H}\rvert)}{\varepsilon})$<br>$\mathrm{SC}^{\text{prop}} = \Omega(\frac{\lvert\mathcal{H}\rvert}{\varepsilon})$ (Thm 5) |
| | $(\perp, \Delta)$ | $\Theta(\lvert\mathcal{H}\rvert)$ (implied by Thm 3) | $\mathrm{SC}^{\text{csv}} = \widetilde{\Omega}(\frac{\lvert\mathcal{H}\rvert}{\varepsilon})$ (Thm 7) |
| | $(\perp, \perp)$ | $\Theta(\lvert\mathcal{H}\rvert)$ (implied by Thm 3) | $\widetilde{\mathcal{O}}(\frac{\lvert\mathcal{H}\rvert}{\varepsilon})$ , $\widetilde{\Omega}(\frac{\lvert\mathcal{H}\rvert}{\varepsilon})$ (Thm 8) |
| nonball | all | $\widetilde{\Omega}(\lvert\mathcal{H}\rvert)$(Cor 1) , $\mathcal{O}(\lvert\mathcal{H}\rvert)$ | $\widetilde{\mathcal{O}}(\frac{\lvert\mathcal{H}\rvert}{\varepsilon})$ , $\widetilde{\Omega}(\frac{\lvert\mathcal{H}\rvert}{\varepsilon})$ (Cor 1) |

[a] A factor of $\operatorname{loglog}(|\mathcal{H}|)$ is neglected.

Table 1: The summary of results. $\widetilde{\mathcal{O}}$ and $\widetilde{\Omega}$ ignore logarithmic factors on $|\mathcal{H}|$ and $\frac{1}{\varepsilon}$. The superscripts prop stands for proper learning algorithms and csv stands for conservative learning algorithms. All lower bounds in the non-strategic setting also apply to the strategic setting, implying that $\mathrm{MB}_{C,F} \geq \Omega(\log(|\mathcal{H}|))$ and $\mathrm{SC}_{C,F} \geq \Omega(\frac{\log(|\mathcal{H}|)}{\varepsilon})$ for all settings of $(C, F)$. In all four settings, a mistake bound of $\mathcal{O}(|\mathcal{H}|)$ can be achieved by simply trying each hypothesis in $\mathcal{H}$ while the sample complexity can be achieved as $\widetilde{\mathcal{O}}(\frac{|\mathcal{H}|}{\varepsilon})$ by converting the mistake bound of $\mathcal{O}(|\mathcal{H}|)$ to a PAC bound using standard techniques.

# 4 Ball manipulations

In ball manipulations, when $\mathcal{B}(x; r) \cap \mathcal{X}_{h,+}$ has multiple elements, the agent will always break ties by selecting the one closest to $x$, i.e., $\Delta(x, h, r) = \arg\min_{x' \in \mathcal{B}(x;r) \cap \mathcal{X}_{h,+}} d(x, x')$. In round $t$, the learner deploys predictor $f_t$, and once he knows $x_t$ and $\widehat{y}_t$, he can calculate $\Delta_t$ himself without needing knowledge of $r_t$ by

$$\Delta_t = \begin{cases} \arg\min_{x' \in \mathcal{X}_{f_t,+}} d(x_t, x') & \text{if } \widehat{y}_t = +1\,, \\ x_t & \text{if } \widehat{y}_t = -1\,. \end{cases}$$

Thus, for ball manipulations, knowing $x_t$ is equivalent to knowing both $x_t$ and $\Delta_t$.

### 4.1 Setting $(x, \Delta)$: Observing $x_t$ Before Choosing $f_t$

**Online learning** We propose a new algorithm with mistake bound of $\log(|\mathcal{H}|)$ in setting $(x, \Delta)$. To achieve a logarithmic mistake bound, we must construct a predictor $f_t$ such that if it makes a mistake, we can reduce a constant fraction of the remaining hypotheses. The primary challenge is that we do not have access to the full information, and predictions of other hypotheses are hidden. To extract the information of predictions of other hypotheses, we take advantage of ball manipulations, which induces an ordering over all hypotheses. Specifically, for any hypothesis $h$ and feature vector $x$, we define the distance between $x$ and $h$ by the distance between $x$ and the positive region by $h$, $\mathcal{X}_h^+$, i.e.,

$$d(x, h) := \min\{d(x, x') | x' \in \mathcal{X}_h^+\}\,. \tag{2}$$

At each round $t$, given $x_t$, the learner calculates the distance $d(x_t, h)$ for all $h$ in the version space (meaning hypotheses consistent with history) and selects a hypothesis $f_t$ such that $d(x_t, f_t)$ is the median among all distances $d(x_t, h)$ for $h$ in the version space. We can show that by selecting $f_t$ in this way, the learner can eliminate half of the version space if $f_t$ makes a mistake. We refer to this algorithm as Strategic Halving, and provide a detailed description of it in Algorithm 1.

**Theorem 1.** *For any feature-ball manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$, Strategic Halving achieves mistake bound* $\mathrm{MB}_{x,\Delta} \leq \log(|\mathcal{H}|)$.

**Algorithm 1** Strategic Halving

```
Initialize the version space VS = H.
for t = 1, ..., T do
    pick an f_t ∈ VS such that d(x_t, f_t) is the median of {d(x_t, h) | h ∈ VS}.
    if ŷ_t ≠ y_t and y_t = + then VS ← VS \ {h ∈ VS | d(x_t, h) ≥ d(x_t, f_t)};
    else if ŷ_t ≠ y_t and y_t = − then VS ← VS \ {h ∈ VS | d(x_t, h) ≤ d(x_t, f_t)}.
end for
```

To prove Theorem 1, we only need to show that each mistake reduces the version space by half. Supposing that $f_t$ misclassifies a true positive example $(x_t, r_t, +1)$ by negative, then we know that $d(x_t, f_t) > r_t$ while the target hypothesis $h^*$ must satisfy that $d(x_t, h^*) \leq r_t$. Hence any $h$ with $d(x_t, h) \geq d(x_t, f_t)$ cannot be $h^*$ and should be eliminated. Since $d(x_t, f_t)$ is the median of $\{d(x_t, h) | h \in \mathrm{VS}\}$, we can elimate half of the version space. It is similar when $f_t$ misclassifies a true negative. The detailed proof is deferred to Appendix C.

**PAC learning** We can convert Strategic Halving to a PAC learner by the standard technique of converting a mistake bound to a PAC bound (GALLANT, 1986). Specifically, the learner runs Strategic Halving until it produces a hypothesis $f_t$ that survives for $\frac{1}{\varepsilon}\log(\frac{\log(|\mathcal{H}|)}{\delta})$ rounds and outputs this $f_t$. Then we have Theorem 2, and the proof is included in Appendix D.

**Theorem 2.** *For any feature-ball manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$, we can achieve* $\mathrm{SC}_{x,\Delta}(\varepsilon, \delta) = \mathcal{O}(\frac{\log(|\mathcal{H}|)}{\varepsilon}\log(\frac{\log(|\mathcal{H}|)}{\delta}))$ *by combining Strategic Halving and the standard technique of converting a mistake bound to a PAC bound.*

### 4.2 Setting $(\perp, (x, \Delta))$: Observing $x_t$ After Choosing $f_t$

When $x_t$ is not revealed before the learner choosing $f_t$, the algorithm of Strategic Halving does not work anymore. We demonstrate that it is impossible to reduce constant fraction of version space when making a mistake, and prove that the mistake bound is lower bounded by $\Omega(|\mathcal{H}|)$ by constructing a negative example of $(\mathcal{H}, \mathcal{Q})$. However, we can still achieve sample complexity with poly-logarithmic dependency on $|\mathcal{H}|$ in the distributional setting.

#### 4.2.1 Results in the Online Learning Model

To offer readers an intuitive understanding of the distinctions between the strategic setting and standard online learning, we commence by presenting an example in which no deterministic learners, including the Halving algorithm, can make fewer than $|\mathcal{H}| - 1$ mistakes.

**Example 1.** *Consider a star shape metric space $(\mathcal{X}, d)$, where $\mathcal{X} = \{0, 1, \ldots, n\}$, $d(i, j) = 2$ and $d(0, i) = 1$ for all $i, j \in [n]$ with $i \neq j$. The hypothesis class is composed of singletons over $[n]$, i.e., $\mathcal{H} = \{2\mathbb{1}_{\{i\}} - 1 | i \in [n]\}$. When the learner is deterministic, the environment can pick an agent $(x_t, r_t, y_t)$ dependent on $f_t$. If $f_t$ is all-negative, then the environment picks $(x_t, r_t, y_t) = (0, 1, +1)$, and then the learner makes a mistake but no hypothesis can be eliminated. If $f_t$ predicts $0$ by positive, the environment will pick $(x_t, r_t, y_t) = (0, 0, -1)$, and then the learner makes a mistake but no hypothesis can be eliminated. If $f_t$ predicts some $i \in [n]$ by positive, the environment will pick $(x_t, r_t, y_t) = (i, 0, -1)$, and then the learner makes a mistake with only one hypothesis $2\mathbb{1}_{\{i\}} - 1$ eliminated. Therefore, the learner will make $n - 1$ mistakes.*

In this work, we allow the learner to be randomized. When an $(x_t, r_t, y_t)$ is generated by the environment, the learner can randomly pick an $f_t$, and the environment does not know the realization of $f_t$ but knows the distribution where $f_t$ comes from. It turns out that randomization does not help much. We prove that there exists an example in which any (possibly randomized) learner will incur $\Omega(|\mathcal{H}|)$ mistakes.

**Theorem 3.** *There exists a feature-ball manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$ s.t. the mistake bound $\mathrm{MB}_{\perp,(x,\Delta)} \geq |\mathcal{H}| - 1$. For any (randomized) algorithm $\mathcal{A}$ and any $T \in \mathbb{N}$, there exists a realizable sequence of $(x_t, r_t, y_t)_{1:T}$ such that with probability at least $1 - \delta$ (over randomness of $\mathcal{A}$), $\mathcal{A}$ makes at least $\min(\frac{T}{5|\mathcal{H}|\log(|\mathcal{H}|/\delta)}, |\mathcal{H}| - 1)$ mistakes.*

Essentially, we design an adversarial environment such that the learner has a probability of $\frac{1}{|\mathcal{H}|}$ of making a mistake at each round before identifying the target function $h^*$. The learner only gains information about the target function when a mistake is made. The detailed proof is deferred to Appendix E. Theorem 3 establishes a lower bound on the mistake bound, which is $|\mathcal{H}| - 1$. However, achieving this bound requires a sufficiently large number of rounds, specifically $T = \widetilde{\Omega}(|\mathcal{H}|^2)$. This raises the question of whether there exists a learning algorithm that can make $o(T)$ mistakes for any $T \leq |\mathcal{H}|^2$. In Example 1, we observed that the adversary can force any deterministic learner to make $|\mathcal{H}| - 1$ mistakes in $|\mathcal{H}| - 1$ rounds. Consequently, no deterministic algorithm can achieve $o(T)$ mistakes.

To address this, we propose a randomized algorithm that closely resembles Algorithm 1, with a modification in the selection of $f_t$. Instead of using line 3, we choose $f_t$ randomly from VS since we lack prior knowledge of $x_t$. This algorithm can be viewed as a variation of the well-known multiplicative weights method, applied exclusively during mistake rounds. For improved clarity, we present this algorithm as Algorithm 3 in Appendix F due to space limitations.

**Theorem 4.** *For any $T \in \mathbb{N}$, Algorithm 3 will make at most $\min(\sqrt{4\log(|\mathcal{H}|)T}, |\mathcal{H}| - 1)$ mistakes in expectation in $T$ rounds.*

Note that the $T$-dependent upper bound in Theorem 4 matches the lower bound in Theorem 3 up to a logarithmic factor when $T = |\mathcal{H}|^2$. This implies that approximately $|\mathcal{H}|^2$ rounds are needed to achieve $|\mathcal{H}| - 1$ mistakes, which is a tight bound up to a logarithmic factor. Proof of Theorem 4 is included in Appendix F.

#### 4.2.2 Results in the PAC Learning Model

In the PAC setting, the goal of the learner is to output a predictor $f_{\mathrm{out}}$ after the repeated interactions. A common class of learning algorithms, which outputs a hypothesis $f_{\mathrm{out}} \in \mathcal{H}$, is called proper. Proper learning algorithms are a common starting point when designing algorithms for new learning problems due to their natural appeal and ability to achieve good performance, such as ERM in classic PAC learning. However, in the current setting, we show that proper learning algorithms do not work well and require a sample size linear in $|\mathcal{H}|$. The formal theorem is stated as follows and the proof is deferred to Appendix G.

**Theorem 5.** *There exists a feature-ball manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$ s.t. $\mathrm{SC}^{prop}_{\perp,\Delta}(\varepsilon, \frac{7}{8}) = \Omega(\frac{|\mathcal{H}|}{\varepsilon})$, where $\mathrm{SC}^{prop}_{\perp,\Delta}(\varepsilon, \delta)$ is the $(\varepsilon, \delta)$-PAC sample complexity achievable by proper algorithms.*

Theorem 5 implies that any algorithm capable of achieving sample complexity sub-linear in $|\mathcal{H}|$ must be improper. As a result, we are inspired to devise an improper learning algorithm. Before presenting the algorithm, we introduce some notations. For two hypotheses $h_1, h_2$, let $h_1 \vee h_2$ denote the union of them, i.e., $(h_1 \vee h_2)(x) = +1$ iff. $h_1(x) = +1$ or $h_2(x) = +1$. Similarly, we can define the union of more than two hypotheses. Then for any union of $k$ hypotheses, $f = \vee_{i=1}^{k} h_i$, the positive region of $f$ is the union of positive regions of the $k$ hypotheses and thus, we have $d(x, f) = \min_{i \in [k]} d(x, h_i)$. Therefore, we can decrease the distance between $f$ and any feature vector $x$ by increasing $k$. Based on this, we devise a new randomized algorithm with improper output, described in Algorithm 2.

**Theorem 6.** *For any feature-ball manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$, we can achieve $\mathrm{SC}_{\perp,(x,\Delta)}(\varepsilon, \delta) = \mathcal{O}(\frac{\log^2(|\mathcal{H}|)+\log(1/\delta)}{\varepsilon} \log(\frac{1}{\delta}))$ by combining Algorithm 2 with a standard confidence boosting technique. Note that the algorithm is improper.*

---

**Algorithm 2**

1: Initialize the version space $\mathrm{VS}_0 = \mathcal{H}$.
2: **for** $t = 1, \ldots, T$ **do**
3: randomly pick $k_t \sim \mathrm{Unif}(\{1, 2, 2^2, \ldots, 2^{\lfloor \log_2(n_t) - 1 \rfloor}\})$ where $n_t = |\mathrm{VS}_{t-1}|$;
4: sample $k_t$ hypotheses $h_1, \ldots, h_{k_t}$ independently and uniformly at random from $\mathrm{VS}_{t-1}$;
5: let $f_t = \vee_{i=1}^{k_t} h_i$.
6: **if** $\widehat{y}_t \neq y_t$ and $y_t = +$ **then** $\mathrm{VS}_t = \mathrm{VS}_{t-1} \setminus \{h \in \mathrm{VS}_{t-1} | d(x_t, h) \geq d(x_t, f_t)\}$;
7: **else if** $\widehat{y}_t \neq y_t$ and $y_t = -$ **then** $\mathrm{VS}_t = \mathrm{VS}_{t-1} \setminus \{h \in \mathrm{VS}_{t-1} | d(x_t, h) \leq d(x_t, f_t)\}$;
8: **else** $\mathrm{VS}_t = \mathrm{VS}_{t-1}$.
9: **end for**
10: randomly pick $\tau$ from $[T]$ and randomly sample $h_1, h_2$ from $\mathrm{VS}_{\tau-1}$ with replacement.
11: **output** $h_1 \vee h_2$

---

Now we outline the high-level ideas behind Algorithm 2. In correct rounds where $f_t$ makes no mistake, the predictions of all hypotheses are either correct or unknown, and thus, it is hard to determine how to make updates. In mistake rounds, we can always update the version space similar to what was done in Strategic Halving. To achieve a poly-logarithmic dependency on $|\mathcal{H}|$, we aim to reduce a significant number of misclassifying hypotheses in mistake rounds. The maximum number we can hope to reduce is a constant fraction of the misclassifying hypotheses. We achieve this by randomly sampling a $f_t$ (lines 3-5) s.t. $f_t$ makes a mistake, and $d(x_t, f_t)$ is greater (smaller) than the median of $d(x_t, h)$ for all misclassifying hypotheses $h$ for true negative (positive) examples. However, due to the asymmetric nature of manipulation, which aims to be predicted as positive, the rate of decreasing misclassifications over true positives is slower than over true negatives. To compensate for this asymmetry, we output a $f_{\mathrm{out}} = h_1 \vee h_2$ with two selected hypotheses $h_1, h_2$ (lines 10-11) instead of a single one to increase the chance of positive prediction.

We prove that Algorithm 2 can achieve small strategic loss in expectation as described in Lemma 1. Then we can achieve the sample complexity in Theorem 6 by boosting Algorithm 2 to a strong learner. This is accomplished by running Algorithm 2 multiple times until we obtain a good predictor. The proofs of Lemma 1 and Theorem 6 are deferred to Appendix H.

**Lemma 1.** *Let $S = (x_t, r_t, y_t)_{t=1}^{T} \sim \mathcal{D}^T$ denote the i.i.d. sampled agents in $T$ rounds and let $\mathcal{A}(S)$ denote the output of Algorithm 2 interacting with $S$. For any feature-ball manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$, when $T \geq \frac{320 \log^2(|\mathcal{H}|)}{\varepsilon}$, we have $\mathbb{E}_{\mathcal{A},S}[\mathcal{L}^{str}(\mathcal{A}(S))] \leq \varepsilon$.*

### 4.3 Settings $(\perp, \Delta)$ and $(\perp, \perp)$

**Online learning** As mentioned in Section 2, both the settings of $(\perp, \Delta)$ and $(\perp, \perp)$ are harder than the setting of $(\perp, (x, \Delta))$, all lower bounds in the setting of $(\perp, (x, \Delta))$ also hold in the former two settings. Therefore, by Theorem 3, we have $\mathrm{MB}_{\perp,\perp} \geq \mathrm{MB}_{\perp,\Delta} \geq \mathrm{MB}_{\perp,(x,\Delta)} = |\mathcal{H}| - 1$.

**PAC learning** In the setting of $(\perp, \Delta)$, Algorithm 2 is not applicable anymore since the learner lacks observation of $x_t$, making it impossible to replicate the version space update steps in lines 6-7. It is worth noting that both PAC learning algorithms we have discussed so far fall under a general category called conservative algorithms, depend only on information from the mistake rounds. Specifically, an algorithm is said to be conservative if for any $t$, the predictor $f_t$ only depends on the history of mistake rounds up to $t$, i.e., $\tau < t$ with $\widehat{y}_\tau \neq y_\tau$, and the output $f_{\mathrm{out}}$ only depends on the history of mistake rounds, i.e., $(f_t, \widehat{y}_t, y_t, \Delta_t)_{t:\widehat{y}_t \neq y_t}$. Any algorithm that goes beyond this category would need to utilize the information in correct rounds. As mentioned earlier, in correct rounds, the predictions of all hypotheses are either correct or unknown, which makes it challenging to determine how to make updates. For conservative algorithms, we present a lower bound on the sample complexity in the following theorem, which is $\widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$, and its proof is included in Appendix I. The optimal sample complexity in the setting $(\perp, \Delta)$ is left as an open problem.

**Theorem 7.** *There exists a feature-ball manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$ s.t. $\mathrm{SC}^{csv}_{\perp,\Delta}(\varepsilon, \frac{7}{8}) = \widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$, where $\mathrm{SC}^{csv}_{\perp,\Delta}(\varepsilon, \delta)$ is $(\varepsilon, \delta)$-PAC the sample complexity achievable by conservative algorithms.*

In the setting of $(\perp, \perp)$, our problem reduces to a best arm identification problem in stochastic bandits. We prove a lower bound on the sample complexity of $\widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$ in Theorem 8 by reduction to stochastic linear bandits and applying the tools from information theory. The proof is deferred to Appendix J.

**Theorem 8.** *There exists a feature-ball manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$ s.t. $\mathrm{SC}_{\perp,\perp}(\varepsilon, \frac{7}{8}) = \widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$.*

# 5 Non-ball Manipulations

In this section, we move on to non-ball manipulations. In ball manipulations, for any feature vector $x$, we have an ordering of hypotheses according to their distances to $x$, which helps to infer the predictions of some hypotheses without implementing them. However, in non-ball manipulations, we don't have such structure anymore. Therefore, even in the simplest setting of observing $x_t$ before $f_t$ and $\Delta_t$, we have the PAC sample complexity lower bounded by $\widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$.

**Theorem 9.** *There exists a feature-manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$ s.t. $\mathrm{SC}_{x,\Delta}(\varepsilon, \frac{7}{8}) = \widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$.*

The proof is deferred to Appendix K. It is worth noting that in the construction of the proof, we let all agents to have their original feature vector $x_t = \mathbf{0}$ such that $x_t$ does not provide any information. Since $(x, \Delta)$ is the simplest setting and any mistake bound can be converted to a PAC bound via standard techniques (see Section B.2 for more details), we have the following corollary.

**Corollary 1.** *There exists a feature-manipulation set space $\mathcal{Q}$ and hypothesis class $\mathcal{H}$ s.t. for all choices of $(C, F)$, $\mathrm{SC}_{C,F}(\varepsilon, \frac{7}{8}) = \widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$ and $\mathrm{MB}_{C,F} = \widetilde{\Omega}(|\mathcal{H}|)$.*

# 6 Discussion and Open Problems

In this work, we investigate the mistake bound and sample complexity of strategic classification across multiple settings. Unlike prior work, we assume that the manipulation is personalized and unknown to the learner, which makes the strategic classification problem more challenging. In the case of ball manipulations, when the original feature vector $x_t$ is revealed prior to choosing $f_t$, the problem exhibits a similar level of difficulty as the non-strategic setting (see Table 1 for details). However, when the original feature vector $x_t$ is not revealed beforehand, the problem becomes significantly more challenging. Specifically, any learner will experience a mistake bound that scales linearly with $|\mathcal{H}|$, and any proper learner will face sample complexity that also scales linearly with $|\mathcal{H}|$. In the case of non-ball manipulations, the situation worsens. Even in the simplest setting, where the original feature is observed before choosing $f_t$ and the manipulated feature is observed afterward, any learner will encounter a linear mistake bound and sample complexity.

Besides the question of optimal sample complexity in the setting of $(\perp, \Delta)$ as mentioned in Sec 4.3, there are some other fundamental open questions.

**Combinatorial measure** Throughout this work, our main focus is on analyzing the dependency on the size of the hypothesis class $|\mathcal{H}|$ without assuming any specific structure of $\mathcal{H}$. Just as VC dimension provides tight characterization for PAC learnability and Littlestone dimension characterizes online learnability, we are curious if there exists a combinatorial measure that captures the essence of strategic classification in this context. In the proofs of the most lower bounds in this work, we consider hypothesis class to be singletons, in which both the VC dimension and Littlestone dimension are 1. Therefore, they cannot be candidates to characterize learnability in the strategic setting.

**Agnostic setting** We primarily concentrate on the realizable setting in this work. However, investigating the sample complexity and regret bounds in the agnostic setting would be an interesting avenue for future research.

# References


Ahmadi, S., Beyhaghi, H., Blum, A., and Naggita, K. (2021). The strategic perceptron. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 6–25.

Ahmadi, S., Beyhaghi, H., Blum, A., and Naggita, K. (2022). On classification of strategic agents who can both game and improve. *arXiv preprint arXiv:2203.00124*.

Ahmadi, S., Blum, A., and Yang, K. (2023). Fundamental bounds on online strategic classification. *arXiv preprint arXiv:2302.12355*.

Brückner, M. and Scheffer, T. (2011). Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–555.

Chen, Y., Liu, Y., and Podimata, C. (2020). Learning strategy-aware linear classifiers. *Advances in Neural Information Processing Systems*, 33:15265–15276.

Dalvi, N., Domingos, P., Sanghai, S., and Verma, D. (2004). Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108.

Dong, J., Roth, A., Schutzman, Z., Waggoner, B., and Wu, Z. S. (2018). Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70.

GALLANT, S. I. (1986). Optimal linear discriminants. *Eighth International Conference on Pattern Recognition*, pages 849–852.

Haghtalab, N., Immorlica, N., Lucier, B., and Wang, J. Z. (2020). Maximizing welfare with incentive-aware evaluation mechanisms. *arXiv preprint arXiv:2011.01956*.

Haghtalab, N., Lykouris, T., Nietert, S., and Wei, A. (2022). Learning in stackelberg games with non-myopic agents. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 917–918.

Hardt, M., Megiddo, N., Papadimitriou, C., and Wootters, M. (2016). Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122.

Hu, L., Immorlica, N., and Vaughan, J. W. (2019). The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 259–268.

Jagadeesan, M., Mendler-Dünner, C., and Hardt, M. (2021). Alternative microfoundations for strategic classification. In *International Conference on Machine Learning*, pages 4687–4697. PMLR.

Kleinberg, J. and Raghavan, M. (2020). How do classifiers induce agents to invest effort strategically? *ACM Transactions on Economics and Computation (TEAC)*, 8(4):1–23.

Lechner, T. and Urner, R. (2022). Learning losses for strategic classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7337–7344.

Milli, S., Miller, J., Dragan, A. D., and Hardt, M. (2019). The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239.

Montasser, O., Hanneke, S., and Srebro, N. (2019). Vc classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory*, pages 2512–2530. PMLR.

Rajaraman, N., Han, Y., Jiao, J., and Ramchandran, K. (2023). Beyond ucb: Statistical complexity and optimal algorithms for non-linear ridge bandits. *arXiv preprint arXiv:2302.06025*.

Sundaram, R., Vullikanti, A., Xu, H., and Yao, F. (2021). Pac-learning for strategic classification. In *International Conference on Machine Learning*, pages 9978–9988. PMLR.

Zhang, H. and Conitzer, V. (2021). Incentive-aware pac learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5797–5804.

Zrnic, T., Mazumdar, E., Sastry, S., and Jordan, M. (2021). Who leads and who follows in strategic classification? *Advances in Neural Information Processing Systems*, 34:15257–15269.

# A Additional Related Work

There has been a lot of research on various other issues and models in strategic classification. Beyond sample complexity, Hu et al. (2019); Milli et al. (2019) focused on other social objectives, such as social burden and fairness. Recent works also explored different models of agent behavior, including proactive agents Zrnic et al. (2021), non-myopic agents (Haghtalab et al., 2022) and noisy agents (Jagadeesan et al., 2021). Ahmadi et al. (2023) considers two agent models of randomized learners: a randomized algorithm model where the agents respond to the realization, and a fractional classifier model where agents respond to the expectation, and our model corresponds to the randomized algorithm model. Additionally, there is also a line of research on agents interested in improving their qualifications instead of gaming (Kleinberg and Raghavan, 2020; Haghtalab et al., 2020; Ahmadi et al., 2022).

Beyond strategic classification, there is a more general research area of learning using data from strategic sources, such as a single data generation player who manipulates the data distribution (Brückner and Scheffer, 2011; Dalvi et al., 2004). Adversarial perturbations can be viewed as another type of strategic source (Montasser et al., 2019).

# B Technical Lemmas

## B.1 Boosting expected guarantee to high probability guarantee

Consider any (possibly randomized) PAC learning algorithm $\mathcal{A}$ in strategic setting, which can output a predictor $\mathcal{A}(S)$ after $T$ steps of interaction with i.i.d. agents $S \sim \mathcal{D}^T$ s.t. $\mathbb{E}\left[\mathcal{L}^{\text{str}}(\mathcal{A}(S))\right] \leq \varepsilon$, where the expectation is taken over both the randomness of $S$ and the randomness of algorithm. One standard way in classic PAC learning of boosting the expected loss guarantee to high probability loss guarantee is: running $\mathcal{A}$ on new data $S$ and verifying the loss of $\mathcal{A}(S)$ on a validation data set; if the validation loss is low, outputting the current $\mathcal{A}(S)$, and repeating this process otherwise.

We will adopt this method to boost the confidence as well. The only difference in our strategic setting is that we can not re-use validation data set as we are only allowed to interact with the data through the interaction protocol. Our boosting scheme is described in the following.

- For round $r = 1, \ldots, R$,
  - Run $\mathcal{A}$ for $T$ steps of interactions to obtain a predictor $h_r$.
  - Apply $h_r$ for the following $m_0$ rounds to obtain the empirical strategic loss on $m_0$, denoted as $\widehat{l}_r = \frac{1}{m_0}\sum_{t=t_r+1}^{t_r+m_0} \ell^{\text{str}}(h_r, (x_t, r_t, y_t))$, where $t_r + 1$ is the starting time of these $m_0$ rounds.
  - Break and output $h_r$ if $\widehat{l}_r \leq 4\varepsilon$.
- If for all $r \in [R]$, $\widehat{l}_r > 4\varepsilon$, output an arbitrary hypothesis.

**Lemma 2.** *Given an algorithm $\mathcal{A}$, which can output a predictor $\mathcal{A}(S)$ after $T$ steps of interaction with i.i.d. agents $S \sim \mathcal{D}^T$ s.t. the expected loss satisfies $\mathbb{E}\left[\mathcal{L}^{str}(\mathcal{A}(S))\right] \leq \varepsilon$. Let $h_{\mathcal{A}}$ denote the output of the above boosting scheme given algorithm $\mathcal{A}$ as input. By setting $R = \log\frac{2}{\delta}$ and $m_0 = \frac{3\log(4R/\delta)}{2\varepsilon}$, we have $\mathcal{L}^{str}(h_{\mathcal{A}}) \leq 8\varepsilon$ with probability $1-\delta$. The total sample size is $R(T+m_0) = \mathcal{O}(\log(\frac{1}{\delta})(T + \frac{\log(1/\delta)}{\varepsilon}))$.*

*Proof.* For all $r = 1, \ldots, R$, we have $\mathbb{E}\left[\mathcal{L}^{\text{str}}(h_r)\right] \leq \varepsilon$. By Markov's inequality, we have

$$\Pr(\mathcal{L}^{\text{str}}(h_r) > 2\varepsilon) \leq \frac{1}{2}\,.$$

For any fixed $h_r$, if $\mathcal{L}^{\text{str}}(h_r) \geq 8\varepsilon$, we will have $\widehat{l}_r \leq 4\varepsilon$ with probability $\leq e^{-m_0\varepsilon}$; if $\mathcal{L}^{\text{str}}(h_r) \leq 2\varepsilon$, we will have $\widehat{l}_r \leq 4\varepsilon$ with probability $\geq 1 - e^{-2m_0\varepsilon/3}$ by Chernoff bound.

Let $E$ denote the event of $\{\exists r \in [R], \mathcal{L}^{\text{str}}(h_r) \leq 2\varepsilon\}$ and $F$ denote the event of $\{\widehat{l_r} > 4\varepsilon$ for all $r \in [R]\}$. When $F$ does not hold, our boosting will output $h_r$ for some $r \in [R]$.

$$\begin{aligned}
&\Pr(\mathcal{L}^{\text{str}}(h_{\mathcal{A}}) > 8\varepsilon) \\
\leq &\Pr(E, \neg F) \Pr(\mathcal{L}^{\text{str}}(h_{\mathcal{A}}) > 8\varepsilon | E, \neg F) + \Pr(E, F) + \Pr(\neg E) \\
\leq &\sum_{r=1}^{R} \Pr(h_{\mathcal{A}} = h_r, \mathcal{L}^{\text{str}}(h_r) > 8\varepsilon | E, \neg F) + \Pr(E, F) + \Pr(\neg E) \\
\leq &R e^{-m_0 \varepsilon} + e^{-2m_0\varepsilon/3} + \frac{1}{2^R} \\
\leq &\delta \,,
\end{aligned}$$

by setting $R = \log \frac{2}{\delta}$ and $m_0 = \frac{3\log(4R/\delta)}{2\varepsilon}$. □

### B.2 Converting mistake bound to PAC bound

In any setting of $(C, F)$, if there is an algorithm $\mathcal{A}$ that can achieve the mistake bound of $B$, then we can convert $\mathcal{A}$ to a conservative algorithm by not updating at correct rounds. The new algorithm can still achieve mistake bound of $B$ as $\mathcal{A}$ still sees a legal sequence of examples. Given any conservative online algorithm, we can convert it to a PAC learning algorithm using the standard longest survivor technique (GALLANT, 1986).

**Lemma 3.** *In any setting of $(C, F)$, given any conservative algorithm $\mathcal{A}$ with mistake bound $B$, let algorithm $\mathcal{A}'$ run $\mathcal{A}$ and output the first $f_t$ which survives over $\frac{1}{\varepsilon}\log(\frac{B}{\delta})$ examples. $\mathcal{A}'$ can achieve sample complexity of $\mathcal{O}(\frac{B}{\varepsilon}\log(\frac{B}{\delta}))$.*

*Proof of Lemma 3.* When the sample size $m \geq \frac{B}{\varepsilon}\log(\frac{B}{\delta})$, the algorithm $\mathcal{A}$ will produce at most $B$ different hypotheses and there must exist one surviving for $\frac{1}{\varepsilon}\log(\frac{B}{\delta})$ rounds since $\mathcal{A}$ is a conservative algorithm with at most $B$ mistakes. Let $h_1, \ldots, h_B$ denote these hypotheses and let $t_1, \ldots, t_B$ denote the time step they are produced. Then we have

$$\begin{aligned}
&\Pr(f_{\text{out}} = h_i \text{ and } \mathcal{L}^{\text{str}}(h_i) > \varepsilon) = \mathbb{E}\left[\Pr(f_{\text{out}} = h_i \text{ and } \mathcal{L}^{\text{str}}(h_i) > \varepsilon | t_i, z_{1:t_i-1})\right] \\
<&\mathbb{E}\left[(1-\varepsilon)^{\frac{1}{\varepsilon}\log(\frac{B}{\delta})}\right] = \frac{\delta}{B} \,.
\end{aligned}$$

By union bound, we have

$$\Pr(\mathcal{L}^{\text{str}}(f_{\text{out}}) > \varepsilon) \leq \sum_{i=1}^{B} \Pr_{z_{1:T}}(f_{\text{out}} = h_i \text{ and } \mathcal{L}^{\text{str}}(h_i) > \varepsilon) < \delta.$$

We are done. □

### B.3 Smooth the distribution

**Lemma 4.** *For any two data distribution $\mathcal{D}_1$ and $\mathcal{D}_2$, let $\mathcal{D}_3 = (1-p)\mathcal{D}_1 + p\mathcal{D}_2$ be the mixture of them. For any setting of $(C, F)$ and any algorithm, let $\mathbf{P}_{\mathcal{D}}$ be the dynamics of $(C(x_1), f_1, y_1, \widehat{y}_1, F(x_1, \Delta_1), \ldots, C(x_T), f_T, y_T, \widehat{y}_T, F(x_T, \Delta_T))$ under the data distribution $\mathcal{D}$. Then for any event $A$, we have $|\mathbf{P}_{\mathcal{D}_3}(A) - \mathbf{P}_{\mathcal{D}_1}(A)| \leq 2pT$.*

*Proof.* Let $B$ denote the event of all $(x_t, u_t, y_t)_{t=1}^T$ being sampled from $\mathcal{D}_1$. Then $\mathbf{P}_{\mathcal{D}_3}(\neg B) \leq pT$. Then

$$\begin{aligned}
\mathbf{P}_{\mathcal{D}_3}(A) &= \mathbf{P}_{\mathcal{D}_3}(A|B)\mathbf{P}_{\mathcal{D}_3}(B) + \mathbf{P}_{\mathcal{D}_3}(A|\neg B)\mathbf{P}_{\mathcal{D}_3}(\neg B) \\
&= \mathbf{P}_{\mathcal{D}_1}(A)\mathbf{P}_{\mathcal{D}_3}(B) + \mathbf{P}_{\mathcal{D}_3}(A|\neg B)\mathbf{P}_{\mathcal{D}_3}(\neg B) \\
&= \mathbf{P}_{\mathcal{D}_1}(A)(1 - \mathbf{P}_{\mathcal{D}_3}(\neg B)) + \mathbf{P}_{\mathcal{D}_3}(A|\neg B)\mathbf{P}_{\mathcal{D}_3}(\neg B) \,.
\end{aligned}$$

By re-arranging terms, we have

$$|\mathbf{P}_{\mathcal{D}_1}(A) - \mathbf{P}_{\mathcal{D}_3}(A)| = |\mathbf{P}_{\mathcal{D}_1}(A)\mathbf{P}_{\mathcal{D}_3}(\neg B) - \mathbf{P}_{\mathcal{D}_3}(A|\neg B)\mathbf{P}_{\mathcal{D}_3}(\neg B)| \leq 2pT \,.$$

□

## C Proof of Theorem 1

*Proof.* When a mistake occurs, there are two cases.

- If $f_t$ misclassifies a true positive example $(x_t, r_t, +1)$ by negative, we know that $d(x_t, f_t) > r_t$ while the target hypothesis $h^*$ must satisfy that $d(x_t, h^*) \leq r_t$. Then any $h \in \text{VS}$ with $d(x_t, h) \geq d(x_t, f_t)$ cannot be $h^*$ and are eliminated. Since $d(x_t, f_t)$ is the median of $\{d(x_t, h) | h \in \text{VS}\}$, we can eliminate half of the version space.
- If $f_t$ misclassifies a true negative example $(x_t, r_t, -1)$ by positive, we know that $d(x_t, f_t) \leq r_t$ while the target hypothesis $h^*$ must satisfy that $d(x_t, h^*) > r_t$. Then any $h \in \text{VS}$ with $d(x_t, h) \leq d(x_t, f_t)$ cannot be $h^*$ and are eliminated. Since $d(x_t, f_t)$ is the median of $\{d(x_t, h) | h \in \text{VS}\}$, we can eliminate half of the version space.

Each mistake reduces the version space by half and thus, the algorithm of Strategic Halving suffers at most $\log_2(|\mathcal{H}|)$ mistakes. □

## D Proof of Theorem 2

*Proof.* In online learning setting, an algorithm is conservative if it updates it's current predictor only when making a mistake. It is straightforward to check that Strategic Halving is conservative. Combined with the technique of converting mistake bound to PAC bound in Lemma 3, we prove Theorem 2. □

## E Proof of Theorem 3

*Proof.* Consider the feature space $\mathcal{X} = \{\mathbf{0}, \mathbf{e}_1, \ldots, \mathbf{e}_n, 0.9\mathbf{e}_1, \ldots, 0.9\mathbf{e}_n\}$, where $\mathbf{e}_i$'s are standard basis vectors in $\mathbb{R}^n$ and metric $d(x, x') = \|x - x'\|_2$ for all $x, x' \in \mathcal{X}$. Let the hypothesis class be a set of singletons over $\{\mathbf{e}_i | i \in [n]\}$, i.e., $\mathcal{H} = \{2\mathbb{1}_{\{\mathbf{e}_i\}} - 1 | i \in [n]\}$. We divide all possible hypotheses (not necessarily in $\mathcal{H}$) into three categories:

- The hypothesis $2\mathbb{1}_\emptyset - 1$, which predicts all negative.
- For each $x \in \{\mathbf{0}, 0.9\mathbf{e}_1, \ldots, 0.9\mathbf{e}_n\}$, let $F_{x,+}$ denote the class of hypotheses $h$ predicting $x$ as positive.
- For each $i \in [n]$, let $F_i$ denote the class of hypotheses $h$ satisfying $h(x) = -1$ for all $x \in \{\mathbf{0}, 0.9\mathbf{e}_1, \ldots, 0.9\mathbf{e}_n\}$ and $h(\mathbf{e}_i) = +1$. And let $F_* = \cup_{i \in [n]} F_i$ denote the union of them.

Note that all hypotheses over $\mathcal{X}$ fall into one of the three categories.

Now we consider a set of adversaries $E_1, \ldots, E_n$, such that the target function in the adversarial environment $E_i$ is $2\mathbb{1}_{\{\mathbf{e}_i\}} - 1$. We allow the learners to be randomized and thus, at round $t$, the learner draws an $f_t$ from a distribution $D(f_t)$ over hypotheses. The adversary, who only knows the distribution $D(f_t)$ but not the realization $f_t$, picks an agent $(x_t, r_t, y_t)$ in the following way.

- Case 1: If there exists $x \in \{\mathbf{0}, 0.9\mathbf{e}_1, \ldots, 0.9\mathbf{e}_n\}$ such that $\Pr_{f_t \sim D(f_t)}(f_t \in F_{x,+}) \geq c$ for some $c > 0$, then for all $j \in [n]$, the adversary $E_j$ picks $(x_t, r_t, y_t) = (x, 0, -1)$. Let $B_{1,x}^t$ denote the event of $f_t \in F_{x,+}$.
  - In this case, the learner will make a mistake with probability $c$. Since for all $h \in \mathcal{H}$, $h(\Delta(x, h, 0)) = h(x) = -1$, they are all consistent with $(x, 0, -1)$.
- Case 2: If $\Pr_{f_t \sim D(f_t)}(f_t = 2\mathbb{1}_\emptyset - 1) \geq c$, then for all $j \in [n]$, the adversary $E_j$ picks $(x_t, r_t, y_t) = (\mathbf{0}, 1, +1)$. Let $B_2^t$ denote the event of $f_t = 2\mathbb{1}_\emptyset - 1$.
  - In this case, with probability $c$, the learner will sample a $f_t = 2\mathbb{1}_\emptyset - 1$ and misclassify $(\mathbf{0}, 1, +1)$. Since for all $h \in \mathcal{H}$, $h(\Delta(\mathbf{0}, h, 1)) = +1$, they are all consistent with $(\mathbf{0}, 1, +1)$.

- Case 3: If the above two cases do not hold, let $i_t = \arg\max_{i\in[n]} \Pr(f_t(\mathbf{e}_i) = 1|f_t \in F_*)$, $x_t = 0.9\mathbf{e}_{i_t}$. For radius and label, different adversaries set them differently. Adversary $E_{i_t}$ will set $(r_t, y_t) = (0, -1)$ while other $E_j$ for $j \neq i_t$ will set $(r_t, y_t) = (0.1, -1)$. Since Cases 1 and 2 do not hold, we have $\Pr_{f_t\sim D(f_t)}(f_t \in F_*) \geq 1 - (n+2)c$. Let $B_3^t$ denote the event of $f_t \in F_*$ and $B_{3,i}^t$ denote the event of $f_t \in F_i$.

  (a) With probability $\Pr(B_{3,i_t}^t) \geq \frac{1}{n}\Pr(B_3^t) \geq \frac{1-(n+2)c}{n}$, the learner samples a $f_t \in F_{i_t}$, and thus misclassifies $(0.9\mathbf{e}_{i_t}, 0.1, -1)$ in $E_j$ for $j \neq i_t$ but correctly classifies $(0.9\mathbf{e}_{i_t}, 0, -1)$. In this case, the learner observes the same feedback in all $E_j$ for $j \neq i_t$ and identifies the target function $2\mathbb{1}_{\{\mathbf{e}_{i_t}\}} - 1$ in $E_{i_t}$.

  (b) If the learner samples a $f_t$ with $f_t(\mathbf{e}_{i_t}) = f_t(0.9\mathbf{e}_{i_t}) = -1$, then the learner observes $x_t = 0.9\mathbf{e}_{i_t}$, $y_t = -1$ and $\widehat{y}_t = -1$ in all $E_j$ for $j \in [n]$. Therefore the learner cannot distinguish between adversaries in this case.

  (c) If the learner samples a $f_t$ with $f_t(0.9\mathbf{e}_{i_t}) = +1$, then the learner observes $x_t = 0.9\mathbf{e}_{i_t}$, $y_t = -1$ and $\widehat{y}_t = +1$ in all $E_j$ for $j \in [n]$. Again, since the feedback are identical in all $E_j$ and the learner cannot distinguish between adversaries in this case.

For any learning algorithm $\mathcal{A}$, his predictions are identical in all of adversarial environments $\{E_j|j \in [n]\}$ before he makes a mistake in Case 3(a) in one environment $E_{i_t}$. His predictions in the following rounds are identical in all of adversarial environments $\{E_j|j \in [n]\} \setminus \{E_{i_t}\}$ before he makes another mistake in Case 3(a). Suppose that we run $\mathcal{A}$ in all adversarial environment of $\{E_j|j \in [n]\}$ simultaneously. Note that once we make a mistake, the mistake must occur simultaneously in at least $n-1$ environments. Specifically, if we make a mistake in Case 1, 2 or 3(c), such a mistake simultaneously occur in all $n$ environments. If we make a mistake in Case 3(a), such a mistake simultaneously occur in all $n$ environments except $E_{i_t}$. Since we will make a mistake with probability at least $\min(c, \frac{1-(n+2)c}{n})$ at each round, there exists one environment in $\{E_j|j \in [n]\}$ in which $\mathcal{A}$ will make $n-1$ mistakes.

Now we lower bound the number of mistakes dependent on $T$. Let $t_1, t_2, \ldots$ denote the time steps in which we makes a mistake. Let $t_0 = 0$ for convenience. Now we prove that

$$
\begin{aligned}
&\Pr(t_i > t_{i-1} + k|t_{i-1}) = \prod_{\tau=t_{i-1}+1}^{t_{i-1}+k} \Pr(\text{we don't make a mistake in round } \tau)\\
\leq &\prod_{\tau=t_{i-1}+1}^{t_{i-1}+k} (\mathbb{1}(\text{Case 3 at round } \tau)(1 - \frac{1-(n+2)c}{n}) + \mathbb{1}(\text{Case 1 or 2 at round } \tau)(1-c))\\
\leq &(1 - \min(\frac{1-(n+2)c}{n}, c))^k \leq (1 - \frac{1}{2(n+2)})^k\,,
\end{aligned}
$$

by setting $c = \frac{1}{2(n+2)}$. Then by letting $k = 2(n+2)\ln(n/\delta)$, we have

$$
\Pr(t_i > t_{i-1} + k|t_{i-1}) \leq \delta/n\,.
$$

For any $T$,

$$
\begin{aligned}
&\Pr(\text{\# of mistakes} < \min(\frac{T}{k+1}, n-1))\\
=\leq &\Pr(\exists i \in [n-1], t_i - t_{i-1} > k)\\
\leq &\sum_{i=1}^{n-1} \Pr(t_i - t_{i-1} > k) \leq \delta\,.
\end{aligned}
$$

Therefore, we have proved that for any $T$, with probability at least $1-\delta$, we will make at least $\min(\frac{T}{2(n+2)\ln(n/\delta)+1}, n-1)$ mistakes. □

## F Proof of Theorem 4

**Algorithm 3** MWMR (Multiplicative Weights on Mistake Rounds)

Initialize the version space $\mathrm{VS} = \mathcal{H}$.
**for** t=1,...,T **do**
  Pick one hypotheses $f_t$ from VS uniformly at random.
  **if** $\widehat{y}_t \neq y_t$ and $y_t = +$ **then**
    $\mathrm{VS} \leftarrow \mathrm{VS} \setminus \{h \in \mathrm{VS} | d(x_t, h) \geq d(x_t, f_t)\}$.
  **else if** $\widehat{y}_t \neq y_t$ and $y_t = -$ **then**
    $\mathrm{VS} \leftarrow \mathrm{VS} \setminus \{h \in \mathrm{VS} | d(x_t, h) \leq d(x_t, f_t)\}$.
  **end if**
**end for**

*Proof.* First, when the algorithm makes a mistake at round $t$, he can at least eliminate $f_t$. Therefore, the total number of mistakes will be upper bounded by $|\mathcal{H}| - 1$.

Let $p_t$ denote the fraction of hypotheses misclassifying $x_t$. We say a hypothesis $h$ is inconsistent with $(x_t, f_t, y_t, \widehat{y}_t)$ iff $(d(x_t, h) \geq d(x_t, f_t) \wedge \widehat{y}_t = - \wedge y_t = +)$ or $(d(x_t, h) \leq d(x_t, f_t) \wedge \widehat{y}_t = + \wedge y_t = -)$. Then we define the following events.

- $E_t$ denotes the event that MWMR makes a mistake at round $t$. We have $\Pr(E_t) = p_t$.
- $B_t$ denotes the event that at least $\frac{p_t}{2}$ fraction of hypotheses are inconsistent with $(x_t, f_t, y_t, \widehat{y}_t)$. We have $\Pr(B_t | E_t) \geq \frac{1}{2}$.

Let $n = |\mathcal{H}|$ denote the cardinality of hypothesis class and $n_t$ denote the number of hypotheses in VS after round $t$. Then we have

$$1 \leq n_T = n \cdot \prod_{t=1}^{T} (1 - \mathbb{1}(E_t) \mathbb{1}(B_t) \frac{p_t}{2}) \,.$$

By taking logarithm of both sides, we have

$$0 \leq \ln(n_T) = \ln(n) + \sum_{t=1}^{T} \ln(1 - \mathbb{1}(E_t) \mathbb{1}(B_t) \frac{p_t}{2}) \leq \ln(n) - \sum_{t=1}^{T} \mathbb{1}(E_t) \mathbb{1}(B_t) \frac{p_t}{2} \,,$$

where the last inequality adopts $\ln(1-x) \leq -x$ for $x \in [0, 1)$. Then by taking expectation of both sides, we have

$$0 \leq \ln(n) - \sum_{t=1}^{T} \Pr(E_t \wedge B_t) \frac{p_t}{2} \,.$$

Since $\Pr(E_t) = p_t$ and $\Pr(B_t | E_t) \geq \frac{1}{2}$, then we have

$$\frac{1}{4} \sum_{t=1}^{T} p_t^2 \leq \ln(n) \,.$$

Then we have the expected number of mistakes $\mathbb{E}\left[\mathcal{M}_{\mathrm{MWMR}}(T)\right]$ as

$$\mathbb{E}\left[\mathcal{M}_{\mathrm{MWMR}}(T)\right] = \sum_{t=1}^{T} p_t \leq \sqrt{\sum_{t=1}^{T} p_t^2} \cdot \sqrt{T} \leq \sqrt{4 \ln(n) T} \,,$$

where the first inequality applies Cauchy-Schwarz inequality. □

## G Proof of Theorem 5

*Proof.* **Construction of $\mathcal{Q}, \mathcal{H}$ and a set of realizable distributions**

- Let feature space $\mathcal{X} = \{\mathbf{0}, \mathbf{e}_1, \dots, \mathbf{e}_n\} \cup X_0$, where $X_0 = \{\frac{\sigma(0,1,\dots,n-1)}{z} | \sigma \in \mathcal{S}_n\}$ with $z = \frac{\sqrt{1^2+\dots+(n-1)^2}}{\alpha}$ for some small $\alpha = 0.1$. Here $\mathcal{S}_n$ is the set of all permutations over $n$ elements. So $X_0$ is the set of points whose coordinates are a permutation of $\{0, 1/z, \dots, (n-1)/z\}$ and all points in $X_0$ have the $\ell_2$ norm equal to $\alpha$. Define a metric $d$ by letting $d(x_1, x_2) = \|x_1 - x_2\|_2$ for all $x_1, x_2 \in \mathcal{X}$. Then for any $x \in X_0$ and $i \in [n]$, $d(x, \mathbf{e}_i) = \|x - \mathbf{e}_i\|_2 = \sqrt{(x_i - 1)^2 + \sum_{j \neq i} x_j^2} = \sqrt{1 + \sum_{j=1}^n x_j^2 - 2x_i} = \sqrt{1 + \alpha^2 - 2x_i}$. Note that we consider space $(\mathcal{X}, d)$ rather than $(\mathbb{R}^n, \|\cdot\|_2)$.
- Let the hypothesis class be a set of singletons over $\{\mathbf{e}_i | i \in [n]\}$, i.e., $\mathcal{H} = \{2\mathbb{1}_{\{\mathbf{e}_i\}} - 1 | i \in [n]\}$.
- We now define a collection of distributions $\{\mathcal{D}_i | i \in [n]\}$ in which $\mathcal{D}_i$ is realized by $2\mathbb{1}_{\{\mathbf{e}_i\}} - 1$. For any $i \in [n]$, $\mathcal{D}_i$ puts probability mass $1 - 3n\varepsilon$ on $(\mathbf{0}, 0, -1)$. For the remaining $3n\varepsilon$ probability mass, $\mathcal{D}_i$ picks $x$ uniformly at random from $X_0$ and label it as positive. If $x_i = 0$, set radius $r(x) = r_u := \sqrt{1 + \alpha^2}$; otherwise, set radius $r(x) = r_l := \sqrt{1 + \alpha^2 - 2 \cdot \frac{1}{z}}$). Hence, $X_0$ are all labeled as positive. For $j \neq i$, $h_j = 2\mathbb{1}_{\{\mathbf{e}_j\}} - 1$ labels $\{x \in X_0 | x_j = 0\}$ negative since $r(x) = r_l$ and $d(x, h_j) = r_u > r(x)$. Therefore, $\mathcal{L}^{\text{str}}(h_j) = \frac{1}{n} \cdot 3n\varepsilon = 3\varepsilon$. To output $f_{\text{out}} \in \mathcal{H}$, we must identify the true target function.

**Information gain from different choices of $f_t$** Let $h^* = 2\mathbb{1}_{\{\mathbf{e}_{i^*}\}} - 1$ denote the target function. Since $(\mathbf{0}, 0, -1)$ is realized by all hypotheses, we can only gain information about the target function when $x_t \in X_0$. For any $x_t \in X_0$, if $d(x_t, f_t) \leq r_l$ or $d(x_t, f_t) > r_u$, we cannot learn anything about the target function. In particular, if $d(x_t, f_t) \leq r_l$, the learner will observe $x_t \sim \text{Unif}(X_0)$, $y_t = +1$, $\widehat{y}_t = +1$ in all $\{\mathcal{D}_i | i \in [n]\}$. If $d(x_t, f_t) > r_u$, the learner will observe $x_t \sim \text{Unif}(X_0)$, $y_t = +1$, $\widehat{y}_t = -1$ in all $\{\mathcal{D}_i | i \in [n]\}$. Therefore, we cannot obtain any information about the target function.

Now for any $x_t \in X_0$, with the $i_t$-th coordinate being 0, we enumerate the distance between $x$ and $x'$ for all $x' \in \mathcal{X}$.

- For all $x' \in X_0$, $d(x, x') \leq \|x\| + \|x'\| \leq 2\alpha < r_l$;
- For all $j \neq i_t$, $d(x, \mathbf{e}_j) = \sqrt{1 + \alpha^2 - 2x_j} \leq r_l$;
- $d(x, \mathbf{e}_{i_t}) = r_u$;
- $d(x, \mathbf{0}) = \alpha < r_l$.

Only $f_t = 2\mathbb{1}_{\{\mathbf{e}_{i_t}\}} - 1$ satisfies that $r_l < d(x_t, f_t) \leq r_u$ and thus, we can only obtain information when $f_t = 2\mathbb{1}_{\{\mathbf{e}_{i_t}\}} - 1$. And the only information we learn is whether $i_t = i^*$ because if $i_t \neq i^*$, no matter which $i^*$ is, our observation is identical. If $i_t \neq i^*$, we can eliminate $2\mathbb{1}_{\{\mathbf{e}_{i_t}\}} - 1$.

**Sample size analysis** For any algorithm $\mathcal{A}$, his predictions are identical in all environments $\{\mathcal{D}_i | i \in [n]\}$ before a round $t$ in which $f_t = 2\mathbb{1}_{\{\mathbf{e}_{i_t}\}} - 1$. Then either he learns $i_t$ in $\mathcal{D}_{i_t}$ or he eliminates $2\mathbb{1}_{\{\mathbf{e}_{i_t}\}} - 1$ and continues to perform the same in the other environments $\{\mathcal{D}_i | i \neq i_t\}$. Suppose that we run $\mathcal{A}$ in all stochastic environments $\{\mathcal{D}_i | i \in [n]\}$ simultaneously. When we identify $i_t$ in environment $\mathcal{D}_{i_t}$, we terminate $\mathcal{A}$ in $\mathcal{D}_{i_t}$. Consider a good algorithm $\mathcal{A}$ which can identify $i$ in $\mathcal{D}_i$ with probability $\frac{7}{8}$ after $T$ rounds of interaction for each $i \in [n]$, that is,

$$\Pr_{\mathcal{D}_i, \mathcal{A}}(i_{\text{out}} \neq i) \leq \frac{1}{8}, \forall i \in [n] \,. \tag{3}$$

Therefore, we have

$$\sum_{i \in [n]} \Pr_{\mathcal{D}_i, \mathcal{A}}(i_{\text{out}} \neq i) \leq \frac{n}{8} \,. \tag{4}$$

Let $n_T$ denote the number of environments that have been terminated by the end of round $T$. Let $B_t$ denote the event of $x_t$ being in $X_0$ and $C_t$ denote the event of $f_t = 2\mathbb{1}_{\{\mathbf{e}_{i_t}\}} - 1$. Then we have $\Pr(B_t) = 3n\varepsilon$ and $\Pr(C_t|B_t) = \frac{1}{n}$, and thus $\Pr(B_t \wedge C_t) = 3n\varepsilon \cdot \frac{1}{n}$. Since at each round, we can eliminate one environment only when $B_t \wedge C_t$ is true, then we have

$$\mathbb{E}\left[n_T\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}(B_t \wedge C_t)\right] = T \cdot 3n\varepsilon \cdot \frac{1}{n} = 3\varepsilon T\,.$$

Therefore, by setting $T = \frac{\lfloor \frac{n}{2} \rfloor - 1}{6\varepsilon}$ and Markov's inequality, we have

$$\Pr(n_T \geq \left\lfloor \frac{n}{2} \right\rfloor - 1) \leq \frac{3\varepsilon T}{\lfloor \frac{n}{2} \rfloor - 1} = \frac{1}{2}\,.$$

When there are $\lceil \frac{n}{2} \rceil + 1$ environments remaining, the algorithm has to pick one $i_{\text{out}}$, which fails in at least $\lceil \frac{n}{2} \rceil$ of the environments. Then we have

$$\sum_{i \in [n]} \Pr_{\mathcal{D}_i, \mathcal{A}}(i_{\text{out}} \neq i) \geq \left\lceil \frac{n}{2} \right\rceil \Pr(n_T \leq \left\lfloor \frac{n}{2} \right\rfloor - 1) \geq \frac{n}{4}\,,$$

which conflicts with Eq (4). Therefore, for any algorithm $\mathcal{A}$, to achieve Eq (3), it requires $T \geq \frac{\lfloor \frac{n}{2} \rfloor - 1}{6\varepsilon}$. □

# H Proof of Theorem 6

Given Lemma 1, we can upper bound the expected strategic loss, then we can boost the confidence of the algorithm through the scheme in Section B.1. Theorem 6 follows by combining Lemma 1 and Lemma 2. Now we only need to prove Lemma 1.

*Proof of Lemma 1.* For any set of hypotheses $H$, for every $z = (x, r, y)$, we define

$$\kappa_p(H, z) := \begin{cases} |\{h \in H | h(\Delta(x, h, r)) = -\}| & \text{if } y = +\,, \\ 0 & \text{otherwise.} \end{cases}$$

So $\kappa_p(H, z)$ is the number of hypotheses mislabeling $z$ for positive $z$'s and $0$ for negative $z$'s. Similarly, we define $\kappa_n$ as follows,

$$\kappa_n(H, z) := \begin{cases} |\{h \in H | h(\Delta(x, h, r)) = +\}| & \text{if } y = -\,, \\ 0 & \text{otherwise.} \end{cases}$$

So $\kappa_n(H, z)$ is the number of hypotheses mislabeling $z$ for negative $z$'s and $0$ for positive $z$'s.

In the following, we divide the proof into two parts. First, recall that in Algorithm 2, the output is constructed by randomly sampling two hypotheses with replacement and taking the union of them. We represent the loss of such a random predictor using $\kappa_p(H, z)$ and $\kappa_n(H, z)$ defined above. Then we show that whenever the algorithm makes a mistake, with some probability, we can reduce $\frac{\kappa_p(\text{VS}_{t-1}, z_t)}{2}$ or $\frac{\kappa_n(\text{VS}_{t-1}, z_t)}{2}$ hypotheses and utilize this to provide a guarantee on the loss of the final output.

**Upper bounds on the strategic loss** For any hypothesis $h$, let $\text{fpr}(h)$ and $\text{fnr}(h)$ denote the false positive rate and false negative rate of $h$ respectively. Let $p_+$ denote the probability of drawing a positive sample from $\mathcal{D}$, i.e., $\Pr_{(x,r,y)\sim\mathcal{D}}(y = +)$ and $p_-$ denote the probability of drawing a negative sample from $\mathcal{D}$. Let $\mathcal{D}_+$ and $\mathcal{D}_-$ denote the data distribution conditional on that the label is positive and that the label is negative respectively. Given any set of hypotheses $H$, we define a random predictor $R2(H) = h_1 \vee h_2$ with $h_1, h_2$ randomly picked from $H$ with replacement. For a true positive $z$, $R2(H)$ will misclassify it with probability $\frac{\kappa_p(H,z)^2}{|H|^2}$. Then we can find that the false negative rate of $R2(H)$ is

$$\text{fnr}(R2(H)) = \mathbb{E}_{z=(x,r,+)\sim\mathcal{D}_+}\left[\Pr(R2(H)(x) = -)\right] = \mathbb{E}_{z=(x,r,+)\sim\mathcal{D}_+}\left[\frac{\kappa_p(H, z)^2}{|H|^2}\right]\,.$$

Similarly, for a true negative $z$, $R2(H)$ will misclassify it with probability $1-(1-\frac{\kappa_n(H,z)}{|H|})^2 \leq \frac{2\kappa_n(H,z)}{|H|}$. Then the false positive rate of $R2(H)$ is

$$\text{fpr}(R2(H)) = \mathbb{E}_{z=(x,r,-)\sim\mathcal{D}_-}\left[\Pr(R2(H)(x)=+)\right] \leq \mathbb{E}_{z=(x,r,-)\sim\mathcal{D}_+}\left[\frac{2\kappa_n(H,z)}{|H|}\right].$$

Hence the loss of $R2(H)$ is

$$\begin{aligned}\mathcal{L}^{\text{str}}(R2(H)) &\leq p_+\mathbb{E}_{z\sim\mathcal{D}_+}\left[\frac{\kappa_p(H,z)^2}{|H|^2}\right] + p_-\mathbb{E}_{z\sim\mathcal{D}_+}\left[\frac{2\kappa_n(H,z)}{|H|}\right] \\ &= \mathbb{E}_{z\sim\mathcal{D}}\left[\frac{\kappa_p(H,z)^2}{|H|^2} + 2\frac{\kappa_n(H,z)}{|H|}\right],\end{aligned} \tag{5}$$

where the last equality holds since $\kappa_p(H,z) = 0$ for true negatives and $\kappa_n(H,z) = 0$ for true positives.

**Loss analysis** In each round, the data $z_t = (x_t, r_t, y_t)$ is sampled from $\mathcal{D}$. When the label $y_t$ is positive, if the drawn $f_t$ satisfying that 1) $f_t(\Delta(x_t, f_t, r_t)) = -$ and 2) $d(x_t, f_t) \leq \text{median}(\{d(x_t,h)|h \in \text{VS}_{t-1}, h(\Delta(x_t,h,r_t)) = -\})$, then we are able to remove $\frac{\kappa_p(\text{VS}_{t-1},z_t)}{2}$ hypotheses from the version space. Let $E_{p,t}$ denote the event of $f_t$ satisfying the conditions 1) and 2). With probability $\frac{1}{\lceil \log_2(n_t)\rceil}$, we sample $k_t = 1$. Then we sample an $f_t \sim \text{Unif}(\text{VS}_{t-1})$. With probability $\frac{\kappa_p(\text{VS}_{t-1},z_t)}{2n_t}$, the sampled $f_t$ satisfies the two conditions. So we have

$$\Pr(E_{p,t}|z_t, \text{VS}_{t-1}) \geq \frac{1}{\log_2(n_t)}\frac{\kappa_p(\text{VS}_{t-1},z_t)}{2n_t}. \tag{6}$$

The case of $y_t$ being negative is similar to the positive case. Let $E_{n,t}$ denote the event of $f_t$ satisfying that 1) $f_t(\Delta(x_t, f_t, r_t)) = +$ and 2) $d(x_t, f_t) \geq \text{median}(\{d(x_t,h)|h \in \text{VS}_{t-1}, h(\Delta(x_t,h,r_t)) = +\})$. If $\kappa_n(\text{VS}_{t-1},z_t) \geq \frac{n_t}{2}$, then with probability $\frac{1}{\lceil \log_2(n_t)\rceil}$, we sample $k_t = 1$. Then with probability greater than $\frac{1}{4}$ we will sample an $f_t$ satisfying that 1) $f_t(\Delta(x_t, f_t, r_t)) = +$ and 2) $d(x_t, f_t) \geq \text{median}(\{d(x_t,h)|h \in \text{VS}_{t-1}, h(\Delta(x_t,h,r_t)) = +\})$. If $\kappa_n(\text{VS}_{t-1},z_t) < \frac{n_t}{2}$, then with probability $\frac{1}{\lceil \log_2(n_t)\rceil}$, we sampled a $k_t$ satisfying

$$\frac{n_t}{4\kappa_n(\text{VS}_{t-1},z_t)} < k_t \leq \frac{n_t}{2\kappa_n(\text{VS}_{t-1},z_t)}.$$

Then we randomly sample $k_t$ hypotheses and the expected number of sampled hypotheses which mislabel $z_t$ is $k_t \cdot \frac{\kappa_n(\text{VS}_{t-1},z_t)}{n_t} \in (\frac{1}{4}, \frac{1}{2}]$. Let $g_t$ (given the above fixed $k_t$) denote the number of sampled hypotheses which mislabel $x_t$ and we have $\mathbb{E}[g_t] \in (\frac{1}{4}, \frac{1}{2}]$. When $g_t > 0$, $f_t$ will misclassify $z_t$ by positive. We have

$$\Pr(g_t = 0) = (1 - \frac{\kappa_n(\text{VS}_{t-1},z_t)}{n_t})^{k_t} < (1 - \frac{\kappa_n(\text{VS}_{t-1},z_t)}{n_t})^{\frac{n_t}{4\kappa_n(\text{VS}_{t-1},z_t)}} \leq e^{-1/4} \leq 0.78$$

and by Markov's inequality, we have

$$\Pr(g_t \geq 3) \leq \frac{\mathbb{E}[g_t]}{3} \leq \frac{1}{6} \leq 0.17.$$

Thus $\Pr(g_t \in \{1,2\}) \geq 0.05$. Conditional on $g_t$ is either 1 or 2, with probability $\geq \frac{1}{4}$, all of these $g_t$ hypotheses $h'$ satisfies $d(x_t, h') \geq \text{median}(\{d(x_t,h)|h \in \text{VS}_{t-1}, h(\Delta(x_t,h,r_t)) = +\})$, which implies that $d(x_t, f_t) \geq \text{median}(\{d(x_t,h)|h \in \text{VS}_{t-1}, h(\Delta(x_t,h,r_t)) = +\})$. Therefore, we have

$$\Pr(E_{n,t}|z_t,, \text{VS}_{t-1}) \geq \frac{1}{80\log_2(n_t)}. \tag{7}$$

Let $v_t$ denote the fraction of hypotheses we eliminated at round $t$, i.e., $v_t = 1 - \frac{n_{t+1}}{n_t}$ . Then we have

$$v_t \geq \mathbb{1}(E_{p,t})\frac{\kappa_p(\text{VS}_{t-1},z_t)}{2n_t} + \mathbb{1}(E_{n,t})\frac{\kappa_n(\text{VS}_{t-1},z_t)}{2n_t}. \tag{8}$$

Since $n_{t+1} = n_t(1 - v_t)$, we have

$$1 \leq n_{T+1} = n \prod_{t=1}^{T} (1 - v_t)\,.$$

By taking logarithm of both sides, we have

$$0 \leq \ln n_{T+1} = \ln n + \sum_{t=1}^{T} \ln(1 - v_t) \leq \ln n - \sum_{t=1}^{T} v_t\,,$$

where we use $\ln(1 - x) \leq -x$ for $x \in [0, 1)$ in the last inequality. By re-arranging terms, we have

$$\sum_{t=1}^{T} v_t \leq \ln n\,.$$

Combined with Eq (8), we have

$$\sum_{t=1}^{T} \mathbb{1}(E_{p,t}) \frac{\kappa_p(\mathrm{VS}_{t-1}, z_t)}{2n_t} + \mathbb{1}(E_{n,t}) \frac{\kappa_n(\mathrm{VS}_{t-1}, z_t)}{2n_t} \leq \ln n\,.$$

By taking expectation w.r.t. the randomness of $f_{1:T}$ and dataset $S = z_{1:T}$ on both sides, we have

$$\sum_{t=1}^{T} \mathbb{E}_{f_{1:T}, z_{1:T}} \left[ \mathbb{1}(E_{p,t}) \frac{\kappa_p(\mathrm{VS}_{t-1}, z_t)}{2n_t} + \mathbb{1}(E_{n,t}) \frac{\kappa_n(\mathrm{VS}_{t-1}, z_t)}{2n_t} \right] \leq \ln n\,.$$

Since the $t$-th term does not depend on $f_{t+1:T}$, $z_{t+1:T}$ and $\mathrm{VS}_{t-1}$ is determined by $z_{1:t-1}$ and $f_{1:t-1}$, the $t$-th term becomes

$$\begin{aligned}
&\mathbb{E}_{f_{1:t}, z_{1:t}} \left[ \mathbb{1}(E_{p,t}) \frac{\kappa_p(\mathrm{VS}_{t-1}, z_t)}{2n_t} + \mathbb{1}(E_{n,t}) \frac{\kappa_n(\mathrm{VS}_{t-1}, z_t)}{2n_t} \right] \\
=&\mathbb{E}_{f_{1:t-1}, z_{1:t}} \left[ \mathbb{E}_{f_t} \left[ \mathbb{1}(E_{p,t}) \frac{\kappa_p(\mathrm{VS}_{t-1}, z_t)}{2n_t} + \mathbb{1}(E_{n,t}) \frac{\kappa_n(\mathrm{VS}_{t-1}, z_t)}{2n_t} | f_{1:t-1}, z_{1:t} \right] \right] \\
=&\mathbb{E}_{f_{1:t-1}, z_{1:t}} \left[ \mathbb{E}_{f_t} \left[ \mathbb{1}(E_{p,t}) | f_{1:t-1}, z_{1:t} \right] \frac{\kappa_p(\mathrm{VS}_{t-1}, z_t)}{2n_t} + \mathbb{E}_{f_t} \left[ \mathbb{1}(E_{n,t}) | f_{1:t-1}, z_{1:t} \right] \frac{\kappa_n(\mathrm{VS}_{t-1}, z_t)}{2n_t} \right] \qquad (9) \\
\geq&\mathbb{E}_{f_{1:t-1}, z_{1:t}} \left[ \frac{1}{\log_2(n_t)} \frac{\kappa_p^2(\mathrm{VS}_{t-1}, z_t)}{4n_t^2} + \frac{1}{80 \log_2(n_t)} \frac{\kappa_n(\mathrm{VS}_{t-1}, z_t)}{2n_t} \right]\,, \qquad (10)
\end{aligned}$$

where Eq (9) holds due to that $\mathrm{VS}_{t-1}$ is determined by $f_{1:t-1}, z_{1:t-1}$ and does not depend on $f_t$ and Eq (10) holds since $\Pr_{f_t}(E_{p,t} | f_{1:t-1}, z_{1:t}) = \Pr_{f_t}(E_{p,t} | \mathrm{VS}_{t-1}, z_t) \geq \frac{1}{\log_2(n_t)} \frac{\kappa_p(\mathrm{VS}_{t-1}, z_t)}{2n_t}$ by Eq (6) and $\Pr_{f_t}(E_{n,t} | f_{1:t-1}, z_{1:t}) = \Pr_{f_t}(E_{n,t} | \mathrm{VS}_{t-1}, z_t) \geq \frac{1}{80 \log_2(n_t)}$ by Eq (7). Thus, we have

$$\sum_{t=1}^{T} \mathbb{E}_{f_{1:t-1}, z_{1:t}} \left[ \frac{1}{\log_2(n_t)} \frac{\kappa_p^2(\mathrm{VS}_{t-1}, z_t)}{4n_t^2} + \frac{1}{80 \log_2(n_t)} \frac{\kappa_n(\mathrm{VS}_{t-1}, z_t)}{2n_t} \right] \leq \ln n\,.$$

Since $z_t \sim \mathcal{D}$ and $z_t$ is independent of $z_{1:t-1}$ and $f_{1:t-1}$, thus, we have the $t$-th term on the LHS being

$$\begin{aligned}
&\mathbb{E}_{f_{1:t-1}, z_{1:t}} \left[ \frac{1}{\log_2(n_t)} \frac{\kappa_p^2(\mathrm{VS}_{t-1}, z_t)}{4n_t^2} + \frac{1}{80 \log_2(n_t)} \frac{\kappa_n(\mathrm{VS}_{t-1}, z_t)}{2n_t} \right] \\
=&\mathbb{E}_{f_{1:t-1}, z_{1:t-1}} \left[ \mathbb{E}_{z_t \sim \mathcal{D}} \left[ \frac{1}{\log_2(n_t)} \frac{\kappa_p^2(\mathrm{VS}_{t-1}, z_t)}{4n_t^2} + \frac{1}{80 \log_2(n_t)} \frac{\kappa_n(\mathrm{VS}_{t-1}, z_t)}{2n_t} \right] \right] \\
\geq&\frac{1}{320 \log_2(n)} \mathbb{E}_{f_{1:t-1}, z_{1:t-1}} \left[ \mathbb{E}_{z \sim \mathcal{D}} \left[ \frac{\kappa_p^2(\mathrm{VS}_{t-1}, z)}{n_t^2} + \frac{2\kappa_n(\mathrm{VS}_{t-1}, z)}{n_t} \right] \right] \\
\geq&\frac{1}{320 \log_2(n)} \mathbb{E}_{f_{1:t-1}, z_{1:t-1}} \left[ \mathcal{L}^{\mathrm{str}}(R2(\mathrm{VS}_{t-1})) \right]\,,
\end{aligned}$$

where the last inequality adopts Eq (5). By summing them up and re-arranging terms, we have

$$\mathbb{E}_{f_{1:T},z_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}^{\text{str}}(R2(\text{VS}_{t-1}))\right]=\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{f_{1:t-1},z_{1:t-1}}\left[\mathcal{L}^{\text{str}}(R2(\text{VS}_{t-1}))\right]\le\frac{320\log_2(n)\ln(n)}{T}\,.$$

For the output of Algorithm 2, which randomly picks $\tau$ from $[T]$, randomly samples $h_1, h_2$ from $\text{VS}_{\tau-1}$ with replacement and outputs $h_1 \vee h_2$, the expected loss is

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}^{\text{str}}(\mathcal{A}(S))\right]=&\mathbb{E}_{S,f_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{h_1,h_2\sim\text{Unif}(\text{VS}_{t-1})}\left[\mathcal{L}^{\text{str}}(h_1\vee h_2)\right]\right]\\=&\mathbb{E}_{S,f_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}^{\text{str}}(R2(\text{VS}_{t-1}))\right]\\\le&\frac{320\log_2(n)\ln(n)}{T}\le\varepsilon\,,\end{aligned}$$

when $T \ge \frac{320\log_2(n)\ln(n)}{\varepsilon}$. □

**Post proof discussion of Lemma 1**

- Upon first inspection, readers might perceive a resemblance between the proof of the loss analysis section and the standard proof of converting regret bound to error bound.This standard proof converts a regret guarantee on $f_{1:T}$ to an error guarantee of $\frac{1}{T}\sum_{t=1}^{T} f_t$. However, in this proof, the predictor employed in each round is $f_t$, while the output is an average over $R2(\text{VS}_{t-1})$ for all $t \in [T]$. Our algorithm does not provide a regret guarantee on $f_{1:T}$.
- Please note that our analysis exhibits asymmetry regarding losses on true positives and true negatives. Specifically, the probability of identifying and reducing half of the misclassifying hypotheses on true positives, denoted as $\Pr(E_{p,t}|z_t, \text{VS}_{t-1})$ (Eq (6)), is lower than the corresponding probability for true negatives, $\Pr(E_{n,t}|z_t, \text{VS}_{t-1})$ (Eq (7)). This discrepancy arises due to the different levels of difficulty in detecting misclassifying hypotheses. For example, if there is exactly one hypothesis $h$ misclassifying a true positive $z_t = (x_t, r_t, y_t)$, it is very hard to detect this $h$. We must select an $f_t$ satisfying that $d(x_t, f_t) > d(x_t, h')$ for all $h' \in \mathcal{H} \setminus \{h\}$ (hence $f_t$ will make a mistake), and that $d(x_t, f_t) \le d(x_t, h)$ (so that we will know $h$ misclassifies $z_t$). Algorithm 2 controls the distance $d(x_t, f_t)$ through $k_t$, which is the number of hypotheses in the union. In this case, we can only detect $h$ when $k_t = 1$ and $f_t = h$, which occurs with probability $\frac{1}{n_t \log(n_t)}$.

  However, if there is exactly one hypothesis $h$ misclassifying a true negative $z_t = (x_t, r_t, y_t)$, we have that $d(x_t, h) = \min_{h'\in\mathcal{H}} d(x_t, h')$. Then by setting $f_t = \vee_{h\in\mathcal{H}} h$, which will makes a mistake and tells us $h$ is a misclassifying hypothesis. Our algorithm will pick such an $f_t$ with probability $\frac{1}{\log(n_t)}$.

# I Proof of Theorem 7

*Proof.* We will prove Theorem 7 by constructing an instance of $\mathcal{Q}$ and $\mathcal{H}$ and showing that for any conservative learning algorithm, there exists a realizable data distribution s.t. achieving $\varepsilon$ loss requires at least $\widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$ samples.

**Construction of $\mathcal{Q}$, $\mathcal{H}$ and a set of realizable distributions**

- Let the input metric space $(\mathcal{X}, d)$ be constructed in the following way. Consider the feature space $\mathcal{X} = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\} \cup X_0$, where $X_0 = \{\frac{\sigma(0,1,\ldots,n-1)}{z}|\sigma \in \mathcal{S}_n\}$ with $z = \frac{\sqrt{1^2+\ldots+(n-1)^2}}{\alpha}$ for some small $\alpha = 0.1$. Here $\mathcal{S}_n$ is the set of all permutations over $n$ elements. So $X_0$ is the set of points whose coordinates are a permutation of $\{0, 1/z, \ldots, (n-1)/z\}$ and all points in $X_0$ have the $\ell_2$ norm equal to $\alpha$. We define the metric $d$ by restricting

$\ell_2$ distance to $\mathcal{X}$, i.e., $d(x_1, x_2) = \|x_1 - x_2\|_2$ for all $x_1, x_2 \in \mathcal{X}$. Then we have that for any $x \in X_0$ and $i \in [n]$, the distance between $x$ and $\mathbf{e}_i$ is

$$d(x, \mathbf{e}_i) = \|x - \mathbf{e}_i\|_2 = \sqrt{(x_i - 1)^2 + \sum_{j \neq i} x_j^2} = \sqrt{1 + \sum_{j=1}^{n} x_j^2 - 2x_i} = \sqrt{1 + \alpha^2 - 2x_i}\,,$$

which is greater than $\sqrt{1 + \alpha^2 - 2\alpha} > 0.8 > 2\alpha$. For any two points $x, x' \in X_0$, $d(x, x') \leq 2\alpha$ by triangle inequality.

- Let the hypothesis class be a set of singletons over $\{\mathbf{e}_i | i \in [n]\}$, i.e., $\mathcal{H} = \{2\mathbb{1}_{\{\mathbf{e}_i\}} - 1 | i \in [n]\}$.
- We now define a collection of distributions $\{\mathcal{D}_i | i \in [n]\}$ in which $\mathcal{D}_i$ is realized by $2\mathbb{1}_{\{\mathbf{e}_i\}} - 1$. For any $i \in [n]$, we define $\mathcal{D}_i$ in the following way. Let the marginal distribution $\mathcal{D}_\mathcal{X}$ over $\mathcal{X}$ be uniform over $X_0$. For any $x$, the label $y$ is $+$ with probability $1 - 6\varepsilon$ and $-$ with probability $6\varepsilon$, i.e., $\mathcal{D}(y|x) = \mathrm{Rad}(1 - 6\varepsilon)$. Note that the marginal distribution $\mathcal{D}_{\mathcal{X}\times\mathcal{Y}} = \mathrm{Unif}(X_0) \times \mathrm{Rad}(1 - 6\varepsilon)$ is identical for any distribution in $\{\mathcal{D}_i | i \in [n]\}$ and does not depend on $i$.

  If the label is positive $y = +$, then let the radius $r = 2$. If the label is negative $y = -$, then let $r = \sqrt{1 + \alpha^2 - 2(x_i + \frac{1}{z})}$, which guarantees that $x$ can be manipulated to $\mathbf{e}_j$ iff $d(x, \mathbf{e}_j) < d(x, \mathbf{e}_i)$ for all $j \in [n]$. Since $x_i \leq \alpha$ and $\frac{1}{z} \leq \alpha$, we have $\sqrt{1 + \alpha^2 - 2(x_i + \frac{1}{z})} \geq \sqrt{1 - 4\alpha} > 2\alpha$. Therefore, for both positive and negative examples, we have radius $r$ strictly greater than $2\alpha$ in both cases.

**Randomization and improperness of the output $f_{\mathrm{out}}$ do not help** Note that algorithms are allowed to output a randomized $f_{\mathrm{out}}$ and to output $f_{\mathrm{out}} \notin \mathcal{H}$. We will show that randomization and improperness of $f_{\mathrm{out}}$ don't make the problem easier. That is, supposing that the data distribution is $\mathcal{D}_{i^*}$ for some $i^* \in [n]$, finding a (possibly randomized and improper) $f_{\mathrm{out}}$ is not easier than identifying $i^*$. Since our feature space $\mathcal{X}$ is finite, we can enumerate all hypotheses not equal to $2\mathbb{1}_{\{\mathbf{e}_{i^*}\}} - 1$ and calculate their strategic population loss as follows.

- $2\mathbb{1}_\emptyset - 1$ predicts all negative and thus $\mathcal{L}^{\mathrm{str}}(2\mathbb{1}_\emptyset - 1) = 1 - 6\varepsilon$;
- For any $a \subset \mathcal{X}$ s.t. $a \cap X_0 \neq \emptyset$, $2\mathbb{1}_a - 1$ will predict any point drawn from $\mathcal{D}_{i^*}$ as positive (since all points have radius greater than $2\alpha$ and the distance between any two points in $X_0$ is smaller than $2\alpha$) and thus $\mathcal{L}^{\mathrm{str}}(2\mathbb{1}_a - 1) = 6\varepsilon$;
- For any $a \subset \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ satisfying that $\exists i \neq i^*$, $\mathbf{e}_i \in a$, we have $\mathcal{L}^{\mathrm{str}}(2\mathbb{1}_a - 1) \geq 3\varepsilon$. This is due to that when $y = -$, $x$ is chosen from $\mathrm{Unif}(X_0)$ and the probability of $d(x, \mathbf{e}_i) < d(x, \mathbf{e}_{i^*})$ is $\frac{1}{2}$. When $d(x, \mathbf{e}_i) < d(x, \mathbf{e}_{i^*})$, $2\mathbb{1}_a - 1$ will predict $x$ as positive.

Under distribution $\mathcal{D}_{i^*}$, if we are able to find a (possibly randomized) $f_{\mathrm{out}}$ with strategic loss of $\mathcal{L}^{\mathrm{str}}(f_{\mathrm{out}}) \leq \varepsilon$, then we have $\mathcal{L}^{\mathrm{str}}(f_{\mathrm{out}}) = \mathbb{E}_{h\sim f_{\mathrm{out}}}[\mathcal{L}^{\mathrm{str}}(h)] \geq \mathrm{Pr}_{h\sim f_{\mathrm{out}}}(h \neq 2\mathbb{1}_{\{\mathbf{e}_{i^*}\}} - 1) \cdot 3\varepsilon$. Thus, $\mathrm{Pr}_{h\sim f_{\mathrm{out}}}(h = 2\mathbb{1}_{\{\mathbf{e}_{i^*}\}} - 1) \geq \frac{2}{3}$. Hence, if we are able to find a (possibly randomized) $f_{\mathrm{out}}$ with $\varepsilon$ error, then we are able to identify $i^*$ by checking which realization of $f_{\mathrm{out}}$ has probability greater than $\frac{2}{3}$. In the following, we will focus on the sample complexity to identify $i^*$. Let $i_{\mathrm{out}}$ denote the algorithm's answer to question "what is $i^*$?".

**Conservative algorithms** When running a conservative algorithm, the rule of choosing $f_t$ at round $t$ and choosing the final output $f_{\mathrm{out}}$ does not depend on the correct rounds, i.e. $\{\tau \in [T] | \widehat{y}_\tau = y_\tau\}$. Let's define

$$\Delta'_t = \begin{cases} \Delta_t & \text{if } \widehat{y}_t \neq y_t \\ \perp & \text{if } \widehat{y}_t = y_t\,, \end{cases} \tag{11}$$

where $\perp$ is just a symbol representing "no information". Then for any conservative algorithm, the selected predictor $f_t$ is determined by $(f_\tau, \widehat{y}_\tau, y_\tau, \Delta'_\tau)$ for $\tau < t$ and the final output $f_{\mathrm{out}}$ is determined by $(f_t, \widehat{y}_t, y_t, \Delta'_t)_{t=1}^T$. From now on, we consider $\Delta'_t$ as the feedback in the learning process of a conservative algorithm since it make no difference from running the same algorithm with feedback $\Delta_t$.

**Smooth the data distribution** For technical reasons (appearing later in the analysis), we don't want to analyze distribution $\{\mathcal{D}_i | i \in [n]\}$ directly as the probability of $\Delta_t = \mathbf{e}_i$ is 0 when $f_t(\mathbf{e}_i) = +1$ under distribution $\mathcal{D}_i$. Instead, we consider the mixture of $\mathcal{D}_i$ and another distribution $\mathcal{D}''_i$, which is identical to $\mathcal{D}_i$ except that $r(x) = d(x, \mathbf{e}_i)$ when $y = -$. More specifically, let $\mathcal{D}'_i = (1-p)\mathcal{D}_i + p\mathcal{D}''_i$ with some extremely small $p$, where $\mathcal{D}''_i$'s marginal distribution over $\mathcal{X} \times \mathcal{Y}$ is still $\mathrm{Unif}(X_0) \times \mathrm{Rad}(1-6\varepsilon)$; the radius is $r = 2$ when $y = +$, ; and the radius is $r = d(x, \mathbf{e}_i)$ when $y = -$. For any data distribution $\mathcal{D}$, let $\mathbf{P}_\mathcal{D}$ be the dynamics of $(f_1, y_1, \widehat{y}_1, \Delta'_1, \ldots, f_T, y_T, \widehat{y}_T, \Delta'_T)$ under $\mathcal{D}$. According to Lemma 4, by setting $p = \frac{\varepsilon}{16n^2}$, when $T \leq \frac{n}{\varepsilon}$, with high probability we never sample from $\mathcal{D}''_i$ and have that for any $i, j \in [n]$

$$\left|\mathbf{P}_{\mathcal{D}_i}(i_{\mathrm{out}} = j) - \mathbf{P}_{\mathcal{D}'_i}(i_{\mathrm{out}} = j)\right| \leq \frac{1}{8}\,. \tag{12}$$

From now on, we only consider distribution $\mathcal{D}'_i$ instead of $\mathcal{D}_i$. The readers might have the question that why not using $\mathcal{D}'_i$ for construction directly. This is because $\mathcal{D}'_i$ does not satisfy realizability and no hypothesis has zero loss under $\mathcal{D}'_i$.

**Information gain from different choices of $f_t$** In each round of interaction, the learner picks a predictor $f_t$, which can be out of $\mathcal{H}$. Here we enumerate all choices of $f_t$.

- $f_t(\cdot) = 2\mathbb{1}_\emptyset - 1$ predicts all points in $\mathcal{X}$ by negative. No matter what $i^*$ is, we will observe $(\Delta_t = x_t, y_t) \sim \mathrm{Unif}(X_0) \times \mathrm{Rad}(1-6\varepsilon)$ and $\widehat{y}_t = -$. They are identically distributed for all $i^* \in [n]$, and thus, $\Delta'_t$ is also identically distributed. We cannot tell any information of $i^*$ from this round.
- $f_t = 2\mathbb{1}_{a_t} - 1$ for some $a_t \subset \mathcal{X}$ s.t. $a \cap X_0 \neq \emptyset$. Then $\Delta_t = \Delta(x_t, f_t, r_t) = \Delta(x_t, f_t, 2\alpha)$ since $r_t > 2\alpha$ and $d(x_t, f_t) \leq 2\alpha$, $\widehat{y}_t = +$, $y_t \sim \mathrm{Rad}(1-6\varepsilon)$. None of these depends on $i^*$ and again, the distribution of $(\widehat{y}_t, y_t, \Delta'_t)$ is identical for all $i^*$ and we cannot tell any information of $i^*$ from this round.
- $f_t = 2\mathbb{1}_{a_t} - 1$ for some non-empty $a_t \subset \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$. For rounds with $y_t = +$, we have $\widehat{y}_t = +$ and $\Delta_t = \Delta(x_t, f_t, 2)$, which still not depend on $i^*$. Thus we cannot learn any information about $i^*$. But we can learn when $y_t = -$. For rounds with $y_t = -$, if $\Delta_t \in a_t$, then we could observe $\widehat{y}_t = +$ and $\Delta'_t = \Delta_t$, which at least tells that $2\mathbb{1}_{\{\Delta_t\}} - 1$ is not the target function (with high probability); if $\Delta_t \notin a_t$, then $\widehat{y}_t = -$ and we observe $\Delta'_t = \perp$.

Therefore, we only need to focus on the rounds with $f_t = 2\mathbb{1}_{a_t} - 1$ for some non-empty $a_t \subset \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ and $y_t = -$. It is worth noting that drawing an example $x$ from $X_0$ uniformly, it is equivalent to uniformly drawing a permutation of $\mathcal{H}$ such that the distances between $x$ and $h$ over all $h \in \mathcal{H}$ are permuted according to it. Then $\Delta_t = \mathbf{e}_j$ iff $\mathbf{e}_j \in a_t$, $d(x, \mathbf{e}_j) \leq d(x, \mathbf{e}_{i^*})$ and $d(x, \mathbf{e}_j) \leq d(x, \mathbf{e}_l)$ for all $\mathbf{e}_l \in a_t$. Let $k_t = |a_t|$ denote the cardinality of $a_t$. In such rounds, under distribution $\mathcal{D}'_{i^*}$, the distribution of $\Delta'_t$ are described as follows.

1. The case of $\mathbf{e}_{i^*} \in a_t$: For all $j \in a_t \setminus \{i^*\}$, with probability $\frac{1}{k_t}$, $d(x_t, \mathbf{e}_j) = \min_{\mathbf{e}_l \in a_t} d(x_t, \mathbf{e}_l)$ and thus, $\Delta'_t = \Delta_t = \mathbf{e}_j$ and $\widehat{y}_t = +$ (mistake round). With probability $\frac{1}{k_t}$, we have $d(x_t, \mathbf{e}_{i^*}) = \min_{\mathbf{e}_l \in a_t} d(x_t, \mathbf{e}_l)$. If the example is drawn from $\mathcal{D}_{i^*}$, we have $\Delta_t = x_t$ and $y_t = -$ (correct round), thus $\Delta'_t = \perp$. If the example is drawn from $\mathcal{D}''_{i^*}$, we have we have $\Delta'_t = \Delta_t = \mathbf{e}_{i^*}$ and $y_t = +$ (mistake round). Therefore, according to the definition of $\Delta'_t$ (Eq (11)), we have

$$\Delta'_t = \begin{cases} \mathbf{e}_j & \text{w.p. } \frac{1}{k_t} \text{ for } \mathbf{e}_j \in a_t, j \neq i^* \\ \mathbf{e}_{i^*} & \text{w.p. } \frac{1}{k_t} p \\ \perp & \text{w.p. } \frac{1}{k_t}(1-p)\,. \end{cases}$$

   We denote this distribution by $P_\in(a_t, i^*)$.

2. The case of $\mathbf{e}_{i^*} \notin a_t$: For all $j \in a_t$, with probability $\frac{1}{k_t+1}$, then $d(x_t, \mathbf{e}_j) = \min_{\mathbf{e}_l \in a_t \cup \{\mathbf{e}_{i^*}\}} d(x_t, \mathbf{e}_l)$ and thus, $\Delta_t = \mathbf{e}_j$ and $\widehat{y}_t = +$ (mistake round). With probability $\frac{1}{k_t+1}$, we have $d(x, \mathbf{e}_{i^*}) < \min_{\mathbf{e}_l \in a_t} d(x_t, \mathbf{e}_l)$ and thus, $\Delta_t = x_t$, $\widehat{y}_t = -$ (correct round), and $\Delta'_t = \perp$. Therefore, the distribution of $\Delta'_t$ is

$$\Delta'_t = \begin{cases} \mathbf{e}_j & \text{w.p. } \frac{1}{k_t+1} \text{ for } \mathbf{e}_j \in a_t \\ \perp & \text{w.p. } \frac{1}{k_t+1} \,. \end{cases}$$

We denote this distribution by $P_{\notin}(a_t)$.

To measure the information obtained from $\Delta'_t$, we will utilize the KL divergence of the distribution of $\Delta'_t$ under the data distribution $\mathcal{D}_{i^*}$ from that under a benchmark distribution. Let $\overline{\mathcal{D}} = \frac{1}{n}\sum_{i\in n} \mathcal{D}'_i$ denote the average distribution. The process of sampling from $\overline{\mathcal{D}}$ is equivalent to sampling $i^*$ uniformly at random from $[n]$ first and drawing a sample from $\mathcal{D}_{i^*}$. Then under $\overline{\mathcal{D}}$, for any $\mathbf{e}_j \in a_t$, we have

$$\begin{aligned}\Pr(\Delta'_t = \mathbf{e}_j) =& \Pr(i^* = j)\Pr(\Delta'_t = \mathbf{e}_j | i^* = j) + \Pr(i^* \in a_t \setminus \{j\})\Pr(\Delta'_t = \mathbf{e}_j | i^* \in a_t \setminus \{j\}) \\ &+ \Pr(i^* \notin a_t)\Pr(\Delta'_t = \mathbf{e}_j | i^* \notin a_t) \\ =& \frac{1}{n}\cdot\frac{p}{k_t} + \frac{k_t-1}{n}\cdot\frac{1}{k_t} + \frac{n-k_t}{n}\cdot\frac{1}{k_t+1} = \frac{nk_t - 1 + p(k_t+1)}{nk_t(k_t+1)}\,,\end{aligned}$$

and

$$\begin{aligned}\Pr(\Delta'_t = \perp) &= \Pr(i^* \in a_t)\Pr(\Delta'_t = \perp | i^* \in a_t) + \Pr(i^* \notin a_t)\Pr(\Delta'_t = \perp | i^* \notin a_t) \\ &= \frac{k_t}{n}\cdot\frac{1-p}{k_t} + \frac{n-k_t}{n}\cdot\frac{1}{k_t+1} = \frac{n+1-p(k_t+1)}{n(k_t+1)}\,.\end{aligned}$$

Thus, the distribution of $\Delta'_t$ under $\overline{\mathcal{D}}$ is

$$\Delta'_t = \begin{cases} \mathbf{e}_j & \text{w.p. } \frac{nk_t-1+p(k_t+1)}{nk_t(k_t+1)} \text{ for } \mathbf{e}_j \in a_t \\ \perp & \text{w.p. } \frac{n+1-p(k_t+1)}{n(k_t+1)} \,. \end{cases}$$

We denote this distribution by $\overline{P}(a_t)$. Next we will compute the KL divergences of $P_{\in}(a_t, i^*)$ and $P_{\notin}(a_t)$ from $\overline{P}(a_t)$. We will use the inequality $\log(1+x) \le x$ for $x \ge 0$ in the following calculation. For any $i^*$ s.t. $\mathbf{e}_{i^*} \in a_t$, we have

$$\begin{aligned}&\mathrm{D}_{\mathrm{KL}}(\overline{P}(a_t)\|P_{\in}(a_t, i^*)) \\ =&(k_t-1)\frac{nk_t-1+p(k_t+1)}{nk_t(k_t+1)}\log(\frac{nk_t-1+p(k_t+1)}{nk_t(k_t+1)}k_t) \\ &+ \frac{nk_t-1+p(k_t+1)}{nk_t(k_t+1)}\log(\frac{nk_t-1+p(k_t+1)}{nk_t(k_t+1)}\cdot\frac{k_t}{p}) \\ &+ \frac{n+1-p(k_t+1)}{n(k_t+1)}\log(\frac{n+1-p(k_t+1)}{n(k_t+1)}\cdot\frac{k_t}{1-p}) \\ \le& 0 + \frac{1}{k_t+1}\log(\frac{1}{p}) + \frac{2p}{k_t+1} = \frac{1}{k_t+1}\log(\frac{1}{p}) + \frac{2p}{k_t+1}\,,\end{aligned} \tag{13}$$

and

$$\begin{aligned}&\mathrm{D}_{\mathrm{KL}}(\overline{P}(a_t)\|P_{\notin}(a_t)) \\ =&k_t\frac{nk_t-1+p(k_t+1)}{nk_t(k_t+1)}\log(\frac{nk_t-1+p(k_t+1)}{nk_t(k_t+1)}(k_t+1)) \\ &+ \frac{n+1-p(k_t+1)}{n(k_t+1)}\log(\frac{n+1-p(k_t+1)}{n(k_t+1)}(k_t+1)) \\ \le& 0 + \frac{n+1}{n^2(k_t+1)} = \frac{n+1}{n^2(k_t+1)}\,.\end{aligned} \tag{14}$$

**Lower bound of the information** We utilize the information theoretical framework of proving lower bounds for linear bandits (Theorem 11 by Rajaraman et al. (2023)) here. For notation simplicity, for all $i \in [n]$, let $\mathbf{P}_i$ denote the dynamics of $(f_1, \Delta'_1, y_1, \widehat{y}_1, \dots, f_T, \Delta'_T, y_T, \widehat{y}_T)$ under $\mathcal{D}'_i$ and $\overline{\mathbf{P}}$ denote the dynamics under $\overline{\mathcal{D}}$. Let $B_t$ denote the event of $\{f_t = 2\mathbb{1}_{a_t} - 1$ for some non-empty $a_t \subset \{\mathbf{e}_1, \dots, \mathbf{e}_n\}\}$. As discussed before, for any $a_t$, conditional on $\neg B_t$ or $y_t = +1$, $(\Delta'_t, y_t, \widehat{y}_t)$ are identical in all $\{\mathcal{D}'_i | i \in [n]\}$, and therefore, also identical in $\overline{\mathcal{D}}$. We can only obtain information at rounds when $B_t \wedge (y_t = -1)$ occurs. In such rounds, we know that $f_t$ is fully determined by history (possibly with external randomness , which does not depend on data distribution), $y_t = -1$ and $\widehat{y}_t$ is fully determined by $\Delta'_t$ ($\widehat{y}_t = +1$ iff. $\Delta'_t \in a_t$).

Therefore, conditional the history $H_{t-1} = (f_1, \Delta'_1, y_1, \widehat{y}_1, \dots, f_{t-1}, \Delta'_{t-1}, y_{t-1}, \widehat{y}_{t-1})$ before time $t$, we have

$$\begin{aligned}
&\mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(f_t, \Delta'_t, y_t, \widehat{y}_t | H_{t-1}) \| \mathbf{P}_i(f_t, \Delta'_t, y_t, \widehat{y}_t | H_{t-1})) \\
=&\overline{\mathbf{P}}(B_t \wedge (y_t = -1)) \mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(\Delta'_t | H_{t-1}, B_t \wedge (y_t = -1)) \| \mathbf{P}_i(\Delta'_t | H_{t-1}, B_t \wedge (y_t = -1))) \\
=&6\varepsilon \overline{\mathbf{P}}(B_t) \mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(\Delta'_t | H_{t-1}, B_t \wedge (y_t = -1)) \| \mathbf{P}_i(\Delta'_t | H_{t-1}, B_t \wedge (y_t = -1)))\,, \qquad (15)
\end{aligned}$$

where the last equality holds due to that $y_t \sim \mathrm{Rad}(1 - 6\varepsilon)$ and does not depend on $B_t$.

For any algorithm that can successfully identify $i$ under the data distribution $\mathcal{D}_i$ with probability $\frac{3}{4}$ for all $i \in [n]$, then $\mathbf{P}_{\mathcal{D}_i}(i_{\mathrm{out}} = i) \geq \frac{3}{4}$ and $\mathbf{P}_{\mathcal{D}_j}(i_{\mathrm{out}} = i) \leq \frac{1}{4}$ for all $j \neq i$. Recall that $\mathcal{D}_i$ and $\mathcal{D}'_i$ are very close when the mixture parameter $p$ is small. Combining with Eq (12), we have

$$\begin{aligned}
&|\mathbf{P}_i(i_{\mathrm{out}} = i) - \mathbf{P}_j(i_{\mathrm{out}} = i)| \\
\geq &\left|\mathbf{P}_{\mathcal{D}_i}(i_{\mathrm{out}} = i) - \mathbf{P}_{\mathcal{D}_j}(i_{\mathrm{out}} = i)\right| - |\mathbf{P}_{\mathcal{D}_i}(i_{\mathrm{out}} = i) - \mathbf{P}_i(i_{\mathrm{out}} = i)| - \left|\mathbf{P}_{\mathcal{D}_j}(i_{\mathrm{out}} = i) - \mathbf{P}_j(i_{\mathrm{out}} = i)\right| \\
\geq &\frac{1}{2} - \frac{1}{4} = \frac{1}{4}\,.
\end{aligned}$$

Then we have the total variation distance between $\mathbf{P}_i$ and $\mathbf{P}_j$

$$\mathrm{TV}(\mathbf{P}_i, \mathbf{P}_j) \geq |\mathbf{P}_i(i_{\mathrm{out}} = i) - \mathbf{P}_j(i_{\mathrm{out}} = i)| \geq \frac{1}{4}\,. \qquad (16)$$

Then we have

$$\begin{aligned}
&\mathbb{E}_{i \sim \mathrm{Unif}([n])}\left[\mathrm{TV}^2(\mathbf{P}_i, \mathbf{P}_{(i+1) \bmod n})\right] \leq 4\mathbb{E}_{i \sim \mathrm{Unif}([n])}\left[\mathrm{TV}^2(\mathbf{P}_i, \overline{\mathbf{P}})\right] \\
\leq &2\mathbb{E}_i\left[\mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}} \| \mathbf{P}_i)\right] && \text{(Pinsker's ineq)} \\
= &2\mathbb{E}_i\left[\sum_{t=1}^{T} \mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(f_t, \Delta'_t, y_t, \widehat{y}_t | H_{t-1}) \| \mathbf{P}_i(f_t, \Delta'_t, y_t, \widehat{y}_t | H_{t-1}))\right] && \text{(Chain rule)} \\
= &12\varepsilon \mathbb{E}_i\left[\sum_{t=1}^{T} \overline{\mathbf{P}}(B_t) \mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(\Delta'_t | H_{t-1}, B_t \wedge (y_t = -1)) \| \mathbf{P}_i(\Delta'_t | H_{t-1}, B_t \wedge (y_t = -1)))\right] && \text{(Apply Eq (15))} \\
= &\frac{12\varepsilon}{n} \sum_{t=1}^{T} \overline{\mathbf{P}}(B_t) \sum_{i=1}^{n} \mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(\Delta'_t | H_{t-1}, B_t \wedge (y_t = -1)) \| \mathbf{P}_i(\Delta'_t | H_{t-1}, B_t \wedge (y_t = -1))) \\
= &\frac{12\varepsilon}{n} \mathbb{E}_{f_{1:T} \sim \overline{\mathbf{P}}}\left[\sum_{t=1}^{T} \mathbb{1}(B_t)\left(\sum_{i: i \in a_t} \mathrm{D}_{\mathrm{KL}}(\overline{P}(a_t) \| P_{\in}(a_t, i)) + \sum_{i: i \notin a_t} \mathrm{D}_{\mathrm{KL}}(\overline{P}(a_t) \| P_{\notin}(a_t))\right)\right] \\
\leq &\frac{12\varepsilon}{n} \sum_{t=1}^{T} \mathbb{E}_{f_{1:T} \sim \overline{\mathbf{P}}}\left[\sum_{i: i \in a_t}\left(\frac{1}{k_t + 1} \log(\frac{1}{p}) + \frac{2p}{k_t + 1}\right) + \sum_{i: i \notin a_t} \frac{n+1}{n^2(k_t + 1)}\right] && \text{(Apply Eq (13),(14))} \\
\leq &\frac{12\varepsilon}{n} \sum_{t=1}^{T} (\log(\frac{1}{p}) + 2p + 1) \\
\leq &\frac{12T\varepsilon(\log(16n^2/\varepsilon) + 2)}{n}\,.
\end{aligned}$$

Combining with Eq (16), we have that there exists a universal constant $c$ such that $T \geq \frac{cn}{\varepsilon(\log(n/\varepsilon)+1)}$.

□

## J Proof of Theorem 8

*Proof.* We will prove Theorem 8 by constructing an instance of $\mathcal{Q}$ and $\mathcal{H}$ and then reduce it to a linear stochastic bandit problem.

**Construction of $\mathcal{Q}, \mathcal{H}$ and a set of realizable distributions**

- Consider the input metric space in the shape of a star, where $\mathcal{X} = \{0, 1, \ldots, n\}$ and the distance function of $d(0, i) = 1$ and $d(i, j) = 2$ for all $i \neq j \in [n]$.
- Let the hypothesis class be a set of singletons over $[n]$, i.e., $\mathcal{H} = \{2\mathbb{1}_{\{i\}} - 1 | i \in [n]\}$.
- We define a collection of distributions $\{\mathcal{D}_i | i \in [n]\}$ in which $\mathcal{D}_i$ is realized by $2\mathbb{1}_{\{i\}} - 1$. The data distribution $\mathcal{D}_i$ put $1 - 3(n-1)\varepsilon$ on $(0, 1, +)$ and $3\varepsilon$ on $(i, 1, -)$ for all $i \neq i^*$. Hence, note that all distributions in $\{\mathcal{D}_i | i \in [n]\}$ share the same distribution support $\{(0, 1, +)\} \cup \{(i, 1, -) | i \in [n]\}$, but have different weights.

**Randomization and improperness of the output $f_{\text{out}}$ do not help.** Note that algorithms are allowed to output a randomized $f_{\text{out}}$ and to output $f_{\text{out}} \notin \mathcal{H}$. We will show that randomization and improperness of $f_{\text{out}}$ don't make the problem easier. Supposing that the data distribution is $\mathcal{D}_{i^*}$ for some $i^* \in [n]$, finding a (possibly randomized and improper) $f_{\text{out}}$ is not easier than identifying $i^*$. Since our feature space $\mathcal{X}$ is finite, we can enumerate all hypotheses not equal to $2\mathbb{1}_{\{i^*\}} - 1$ and calculate their strategic population loss as follows. The hypothesis $2\mathbb{1}_{\emptyset} - 1$ will predict all by negative and thus $\mathcal{L}^{\text{str}}(2\mathbb{1}_{\emptyset} - 1) = 1 - 3(n-1)\varepsilon$. For any hypothesis predicting 0 by positive, it will predict all points in the distribution support by positive and thus incurs strategic loss $3(n-1)\varepsilon$. For any hypothesis predicting 0 by negative and some $i \neq i^*$ by positive, then it will misclassify $(i, 1, -)$ and incur strategic loss $3\varepsilon$. Therefore, for any hypothesis $h \neq 2\mathbb{1}_{\{i^*\}} - 1$, we have $\mathcal{L}^{\text{str}}_{\mathcal{D}_{i^*}}(h) \geq 3\varepsilon$.

Similar to the proof of Theorem 7, under distribution $\mathcal{D}_{i^*}$, if we are able to find a (possibly randomized) $f_{\text{out}}$ with strategic loss $\mathcal{L}^{\text{str}}(f_{\text{out}}) \leq \varepsilon$. Then $\Pr_{h\sim f_{\text{out}}}(h = 2\mathbb{1}_{\{i^*\}} - 1) \geq \frac{2}{3}$. We can identify $i^*$ by checking which realization of $f_{\text{out}}$ has probability greater than $\frac{2}{3}$. In the following, we will focus on the sample complexity to identify the target function $2\mathbb{1}_{\{i^*\}} - 1$ or simply $i^*$. Let $i_{\text{out}}$ denote the algorithm's answer to question of "what is $i^*$?".

**Smooth the data distribution** For technical reasons (appearing later in the analysis), we don't want to analyze distribution $\{\mathcal{D}_i | i \in [n]\}$ directly as the probability of $(i, 1, -)$ is 0 under distribution $\mathcal{D}_i$. Instead, for each $i \in [n]$, let $\mathcal{D}'_i = (1-p)\mathcal{D}_i + p\mathcal{D}''_i$ be the mixture of $\mathcal{D}_i$ and $\mathcal{D}''_i$ for some small $p$, where $\mathcal{D}''_i = (1 - 3(n-1)\varepsilon)\mathbb{1}_{\{(0,1,+)\}} + 3(n-1)\varepsilon\mathbb{1}_{\{(i,1,-)\}}$. Specifically,

$$\mathcal{D}'_i(z) = \begin{cases} 1 - 3(n-1)\varepsilon & \text{for } z = (0, 1, +) \\ 3(1-p)\varepsilon & \text{for } z = (j, 1, -), \forall j \neq i \\ 3(n-1)p\varepsilon & \text{for } z = (i, 1, -) \end{cases}$$

For any data distribution $\mathcal{D}$, let $\mathbf{P}_{\mathcal{D}}$ be the dynamics of $(f_1, y_1, \widehat{y}_1, \ldots, f_T, y_T, \widehat{y}_T)$ under $\mathcal{D}$. According to Lemma 4, by setting $p = \frac{\varepsilon}{16n^2}$, when $T \leq \frac{n}{\varepsilon}$, we have that for any $i, j \in [n]$

$$\left|\mathbf{P}_{\mathcal{D}_i}(i_{\text{out}} = j) - \mathbf{P}_{\mathcal{D}'_i}(i_{\text{out}} = j)\right| \leq \frac{1}{8}. \tag{17}$$

From now on, we only consider distribution $\mathcal{D}'_i$ instead of $\mathcal{D}_i$. The readers might have the question that why not using $\mathcal{D}'_i$ for construction directly. This is because no hypothesis has zero loss under $\mathcal{D}'_i$, and thus $\mathcal{D}'_i$ does not satisfy realizability requirement.

**Information gain from different choices of $f_t$** Note that in each round, the learner picks a $f_t$ and then only observes $\widehat{y}_t$ and $y_t$. Here we enumerate choices of $f_t$ as follows.

1. $f_t = 2\mathbb{1}_\emptyset - 1$ predicts all points in $\mathcal{X}$ by negative. No matter what $i^*$ is, we observe $\widehat{y}_t = -$ and $y_t = 2\mathbb{1}(x_t = 0) - 1$. Hence $(\widehat{y}_t, y_t)$ are identically distributed for all $i^* \in [n]$, and thus, we cannot learn anything about $i^*$ from this round.

2. $f_t$ predicts 0 by positive. Then no matter what $i^*$ is, we have $\widehat{y}_t = +$ and $y_t = \mathbb{1}(x_t = 0)$. Thus again, we cannot learn anything about $i^*$.

3. $f_t = 2\mathbb{1}_{a_t} - 1$ for some non-empty $a_t \subset [n]$. For rounds with $x_t = 0$, we have $\widehat{y}_t = y_t = +$ no matter what $i^*$ is and thus, we cannot learn anything about $i^*$. For rounds with $y_t = -$, i.e., $x_t \neq 0$, we will observe $\widehat{y}_t = f_t(\Delta(x_t, f_t, 1)) = \mathbb{1}(x_t \in a_t)$.

Hence, we can only extract information with the third type of $f_t$ at rounds with $x_t \neq 0$.

**Reduction to stochastic linear bandits** In rounds with $f_t = 2\mathbb{1}_{a_t} - 1$ for some non-empty $a_t \subset [n]$ and $x_t \neq 0$, our problem is identical to a stochastic linear bandit problem. Let us state our problem as Problem 1 and a linear bandit problem as Problem 2. Let $A = \{0,1\}^n \setminus \{\mathbf{0}\}$.

**Problem 1.** *The environment picks an $i^* \in [n]$. At each round $t$, the environment picks $x_t \in \{\mathbf{e}_i | i \in [n]\}$ with $P(i) = \frac{1-p}{n-1}$ for $i \neq i^*$ and $P(i^*) = p$ and the learner picks an $a_t \in A$ (where we use a $n$-bit string to represent $a_t$ and $a_{t,i} = 1$ means that $a_t$ predicts $i$ by positive). Then the learner observes $\widehat{y}_t = \mathbb{1}(a_t^\intercal x_t > 0)$ (where we use 0 to represent nagative label).*

**Problem 2.** *The environment picks a linear parameter $w^* \in \{w^i | i \in [n]\}$ with $w^i = \frac{1-p}{n-1}\mathbf{1} - (\frac{1-p}{n-1} - p)\mathbf{e}_i$. The arm set is $A$. For each arm $a \in A$, the reward is i.i.d. from the following distribution:*

$$r_w(a) = \begin{cases} -1, & \text{w.p. } w^\intercal a\,, \\ 0\,. \end{cases} \tag{18}$$

*If the linear parameter $w^* = w^{i^*}$, the optimal arm is $\mathbf{e}_{i^*}$.*

**Claim 1.** *For any $\delta > 0$, for any algorithm $\mathcal{A}$ that identify $i^*$ correctly with probability $1 - \delta$ within $T$ rounds for any $i^* \in [n]$ in Problem 1, we can construct another algorithm $\mathcal{A}'$ can also identify the optimal arm in any environment with probability $1 - \delta$ within $T$ rounds in Problem 2.*

This claim follows directly from the problem descriptions. Given any algorithm $\mathcal{A}$ for Problem 1, we can construct another algorithm $\mathcal{A}'$ which simulates $\mathcal{A}$. At round $t$, if $\mathcal{A}$ selects predictor $a_t$, then $\mathcal{A}'$ picks arm the same as $a_t$. Then $\mathcal{A}'$ observes a reward $r_{w^{i^*}}(a_t)$, which is $-1$ w.p. $w^{i^*\intercal} a_t$ and feed $-r_{w^{i^*}}(a_t)$ to $\mathcal{A}$. Since $\widehat{y}_t$ in Problem 1 is 1 w.p. $\sum_{i=1}^n a_{t,i} P(i) = w^{i^*\intercal} a_t$, it is distributed identically as $-r_{w^{i^*}}(a_t)$. Since $\mathcal{A}$ will be able to identify $i^*$ w.p. $1 - \delta$ in $T$ rounds, $\mathcal{A}'$ just need to output $\mathbf{e}_{i^*}$ as the optimal arm.

Then any lower bound on $T$ for Problem 2 also lower bounds Problem 1. Hence, we adopt the information theoretical framework of proving lower bounds for linear bandits (Theorem 11 by Rajaraman et al. (2023)) to prove a lower bound for our problem. In fact, we also apply this framework to prove the lower bounds in other settings of this work, including Theorem 7 and Theorem 9.

**Lower bound of the information** For notation simplicity, for all $i \in [n]$, let $\mathbf{P}_i$ denote the dynamics of $(f_1, y_1, \widehat{y}_1, \ldots, f_T, y_T, \widehat{y}_T)$ under $\mathcal{D}'_i$ and and $\overline{\mathbf{P}}$ denote the dynamics under $\overline{\mathcal{D}} = \frac{1}{n}\mathcal{D}'_i$. Let $B_t$ denote the event of $\{f_t = 2\mathbb{1}_{a_t} - 1$ for some non-empty $a_t \subset [n]\}$. As discussed before, for any $a_t$, conditional on $\neg B_t$ or $y_t = +1$, $(x_t, y_t, \widehat{y}_t)$ are identical in all $\{\mathcal{D}'_i | i \in [n]\}$, and therefore, also identical in $\overline{\mathcal{D}}$. We can only obtain information at rounds when $B_t \wedge y_t = -1$ occurs. In such rounds, $f_t$ is fully determined by history (possibly with external randomness , which does not depend on data distribution), $y_t = -1$ and $\widehat{y}_t = -r_w(a_t)$ with $r_w(a_t)$ sampled from the distribution defined in Eq (18).

For any algorithm that can successfully identify $i$ under the data distribution $\mathcal{D}_i$ with probability $\frac{3}{4}$ for all $i \in [n]$, then $\mathbf{P}_{\mathcal{D}_i}(i_{\text{out}} = i) \geq \frac{3}{4}$ and $\mathbf{P}_{\mathcal{D}_j}(i_{\text{out}} = i) \leq \frac{1}{4}$ for all $j \neq i$. Recall that $\mathcal{D}_i$ and $\mathcal{D}'_i$

are very close when the mixture parameter $p$ is small. Combining with Eq (17), we have

$$
\begin{aligned}
&|\mathbf{P}_i(i_{\text{out}} = i) - \mathbf{P}_j(i_{\text{out}} = i)| \\
\geq& \left|\mathbf{P}_{\mathcal{D}_i}(i_{\text{out}} = i) - \mathbf{P}_{\mathcal{D}_j}(i_{\text{out}} = i)\right| - |\mathbf{P}_{\mathcal{D}_i}(i_{\text{out}} = i) - \mathbf{P}_i(i_{\text{out}} = i)| - \left|\mathbf{P}_{\mathcal{D}_j}(i_{\text{out}} = i) - \mathbf{P}_j(i_{\text{out}} = i)\right| \\
\geq& \frac{1}{2} - \frac{1}{4} = \frac{1}{4} \,. 
\end{aligned} \tag{19}
$$

Let $\overline{w} = \frac{1}{n}\mathbf{1}$. Let $\mathrm{kl}(q, q')$ denote the KL divergence from $\mathrm{Ber}(q)$ to $\mathrm{Ber}(q')$. Let $H_{t-1} = (f_1, y_1, \widehat{y}_1, \ldots, f_{t-1}, y_{t-1}, \widehat{y}_{t-1})$ denote the history up to time $t-1$. Then we have

$$
\begin{aligned}
&\mathbb{E}_{i\sim \mathrm{Unif}([n])}\left[\mathrm{TV}^2(\mathbf{P}_i, \mathbf{P}_{i+1 \bmod n})\right] \leq 4\mathbb{E}_{i\sim \mathrm{Unif}([n])}\left[\mathrm{TV}^2(\mathbf{P}_i, \overline{\mathbf{P}})\right] \\
\leq& 2\mathbb{E}_i\left[\mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}\|\mathbf{P}_i)\right] && \text{(Pinsker's ineq)} \\
=& 2\mathbb{E}_i\left[\sum_{t=1}^T \mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(f_t, y_t, \widehat{y}_t|H_{t-1})\|\mathbf{P}_i(f_t, y_t, \widehat{y}_t|H_{t-1}))\right] && \text{(Chain rule)} \\
=& 2\mathbb{E}_i\left[\sum_{t=1}^T \overline{\mathbf{P}}(B_t \wedge y_t = -1)\mathbb{E}_{a_{1:T}\sim\overline{\mathbf{P}}}\left[\mathrm{D}_{\mathrm{KL}}(\mathrm{Ber}(\langle \overline{w}, a_t\rangle)\|\mathrm{Ber}(\langle w^i, a_t\rangle))\right]\right] \\
=& 6(n-1)\varepsilon\mathbb{E}_i\left[\sum_{t=1}^T \overline{\mathbf{P}}(B_t)\mathbb{E}_{a_{1:T}\sim\overline{\mathbf{P}}}\left[\mathrm{D}_{\mathrm{KL}}(\mathrm{Ber}(\langle \overline{w}, a_t\rangle)\|\mathrm{Ber}(\langle w^i, a_t\rangle))\right]\right] \\
=& \frac{6(n-1)\varepsilon}{n}\sum_{t=1}^T \mathbb{E}_{a_{1:T}\sim\overline{\mathbf{P}}}\left[\sum_{i=1}^n \mathrm{D}_{\mathrm{KL}}(\mathrm{Ber}(\langle \overline{w}, a_t\rangle)\|\mathrm{Ber}(\langle w^i, a_t\rangle))\right] \\
=& \frac{6(n-1)\varepsilon}{n}\sum_{t=1}^T \mathbb{E}_{a_{1:T}\sim\overline{\mathbf{P}}}\left[\sum_{i:i\in a_t}\mathrm{kl}(\frac{k_t}{n}, \frac{(k_t-1)(1-p)}{n-1}+p) + \sum_{i:i\notin a_t}\mathrm{kl}(\frac{k_t}{n}, \frac{k_t(1-p)}{n-1})\right] \\
=& \frac{6(n-1)\varepsilon}{n}\sum_{t=1}^T \mathbb{E}_{a_{1:T}\sim\overline{\mathbf{P}}}\left[k_t\mathrm{kl}(\frac{k_t}{n}, \frac{(k_t-1)(1-p)}{n-1}+p) + (n-k_t)\mathrm{kl}(\frac{k_t}{n}, \frac{k_t(1-p)}{n-1})\right] && (20)
\end{aligned}
$$

If $k_t = 1$, then

$$
k_t \cdot \mathrm{kl}(\frac{k_t}{n}, \frac{(k_t-1)(1-p)}{n-1}+p) = \mathrm{kl}(\frac{1}{n}, p) \leq \frac{1}{n}\log(\frac{1}{p})\,,
$$

and

$$
(n-k_t)\cdot\mathrm{kl}(\frac{k_t}{n}, \frac{k_t(1-p)}{n-1}) = (n-1)\cdot\mathrm{kl}(\frac{1}{n}, \frac{1-p}{n-1}) \leq \frac{1}{(1-p)n(n-2)}\,,
$$

where the ineq holds due to $\mathrm{kl}(q, q') \leq \frac{(q-q')^2}{q'(1-q')}$. If $k_t = n-1$, it is symmetric to the case of $k_t = 1$. We have

$$
\begin{aligned}
&k_t \cdot \mathrm{kl}(\frac{k_t}{n}, \frac{(k_t-1)(1-p)}{n-1}+p) = (n-1)\mathrm{kl}(\frac{n-1}{n}, \frac{n-2}{n-1} + \frac{1}{n-1}p) = (n-1)\mathrm{kl}(\frac{1}{n}, \frac{1-p}{n-1}) \\
\leq& \frac{1}{(1-p)n(n-2)}\,,
\end{aligned}
$$

and

$$
(n-k_t)\cdot\mathrm{kl}(\frac{k_t}{n}, \frac{k_t(1-p)}{n-1}) = \mathrm{kl}(\frac{n-1}{n}, 1-p) = \mathrm{kl}(\frac{1}{n}, p) \leq \frac{1}{n}\log(\frac{1}{p})\,.
$$

If $1 < k_t < n-1$, then

$$
\begin{aligned}
k_t \cdot \mathrm{kl}(\frac{k_t}{n}, \frac{(k_t-1)(1-p)}{n-1}+p) =& k_t \cdot \mathrm{kl}(\frac{k_t}{n}, \frac{k_t-1}{n-1} + \frac{n-k_t}{n-1}p) \overset{(a)}{\leq} k_t \cdot \mathrm{kl}(\frac{k_t}{n}, \frac{k_t-1}{n-1}) \\
\overset{(b)}{\leq}& k_t \cdot \frac{(\frac{k_t}{n} - \frac{k_t-1}{n-1})^2}{\frac{k_t-1}{n-1}(1-\frac{k_t-1}{n-1})} = k_t \cdot \frac{n-k_t}{n^2(k_t-1)} \leq \frac{k_t\cdot}{n(k_t-1)} \leq \frac{2}{n}\,,
\end{aligned}
$$

where inequality (a) holds due to that $\frac{k_t-1}{n-1} + \frac{n-k_t}{n-1}p \leq \frac{k_t}{n}$ and $\mathrm{kl}(q, q')$ is monotonically decreasing in $q'$ when $q' \leq q$ and inequality (b) adopts $\mathrm{kl}(q, q') \leq \frac{(q-q')^2}{q'(1-q')}$, and

$$(n-k_t) \cdot \mathrm{kl}(\frac{k_t}{n}, \frac{k_t(1-p)}{n-1}) \leq (n-k_t) \cdot \mathrm{kl}(\frac{k_t}{n}, \frac{k_t}{n-1}) \leq \frac{k_t(n-k_t)}{n^2(n-1-k_t)} \leq \frac{2k_t}{n^2} ,$$

where the first inequality hold due to that $\frac{k_t(1-p)}{n-1} \geq \frac{k_t}{n}$, and $\mathrm{kl}(q, q')$ is monotonically increasing in $q'$ when $q' \geq q$ and the second inequality adopts $\mathrm{kl}(q, q') \leq \frac{(q-q')^2}{q'(1-q')}$. Therefore, we have

$$\text{Eq (20)} \leq \frac{6(n-1)\varepsilon}{n} \sum_{t=1}^{T} \mathbb{E}_{a_{1:T} \sim \overline{\mathbf{P}}} \left[\frac{2}{n} \log(\frac{1}{p})\right] \leq \frac{12\varepsilon T \log(1/p)}{n} .$$

Combining with Eq (19), we have that there exists a universal constant $c$ such that $T \geq \frac{cn}{\varepsilon(\log(n/\varepsilon)+1)}$.

□

# K Proof of Theorem 9

*Proof.* We will prove Theorem 9 by constructing an instance of $\mathcal{Q}$ and $\mathcal{H}$ and showing that for any learning algorithm, there exists a realizable data distribution s.t. achieving $\varepsilon$ loss requires at least $\widetilde{\Omega}(\frac{|\mathcal{H}|}{\varepsilon})$ samples.

**Construction of $\mathcal{Q}$, $\mathcal{H}$ and a set of realizable distributions**

- Let feature vector space $\mathcal{X} = \{0, 1, \ldots, n\}$ and let the space of feature-manipulation set pairs $\mathcal{Q} = \{(0, \{0\} \cup s) | s \subset [n]\}$. That is to say, every agent has the same original feature vector $x = 0$ but has different manipulation ability according to $s$.
- Let the hypothesis class be a set of singletons over $[n]$, i.e., $\mathcal{H} = \{2\mathbb{1}_{\{i\}} - 1 | i \in [n]\}$.
- We now define a collection of distributions $\{\mathcal{D}_i | i \in [n]\}$ in which $\mathcal{D}_i$ is realized by $2\mathbb{1}_{\{i\}} - 1$. For any $i \in [n]$, let $\mathcal{D}_i$ put probability mass $1 - 6\varepsilon$ on $(0, \mathcal{X}, +1)$ and $6\varepsilon$ uniformly over $\{(0, \{0\} \cup s_{\sigma,i}, -1) | \sigma \in \mathcal{S}_n\}$, where $\mathcal{S}_n$ is the set of all permutations over $n$ elements and $s_{\sigma,i} := \{j | \sigma^{-1}(j) < \sigma^{-1}(i)\}$ is the set of elements appearing before $i$ in the permutation $(\sigma(1), \ldots, \sigma(n))$. In other words, with probability $1 - 6\varepsilon$, we will sample $(0, \mathcal{X}, +1)$ and with $\varepsilon$, we will randomly draw a permutation $\sigma \sim \mathrm{Unif}(\mathcal{S}_n)$ and return $(0, \{0\} \cup s_{\sigma,i}, -1)$. The data distribution $\mathcal{D}_i$ is realized by $2\mathbb{1}_{\{i\}} - 1$ since for negative examples $(0, \{0\} \cup s_{\sigma,i}, -1)$, we have $i \notin s$ and for positive examples $(0, \mathcal{X}, +1)$, we have $i \in \mathcal{X}$.

**Randomization and improperness of the output $f_{\text{out}}$ do not help** Note that algorithms are allowed to output a randomized $f_{\text{out}}$ and to output $f_{\text{out}} \notin \mathcal{H}$. We will show that randomization and improperness of $f_{\text{out}}$ don't make the problem easier. That is, supposing that the data distribution is $\mathcal{D}_{i^*}$ for some $i^* \in [n]$, finding a (possibly randomized and improper) $f_{\text{out}}$ is not easier than identifying $i^*$. Since our feature space $\mathcal{X}$ is finite, we can enumerate all hypotheses not equal to $2\mathbb{1}_{\{i^*\}} - 1$ and calculate their strategic population loss as follows.

- $2\mathbb{1}_{\emptyset} - 1$ predicts all points in $\mathcal{X}$ by negative and thus $\mathcal{L}^{\text{str}}(2\mathbb{1}_{\emptyset} - 1) = 1 - 6\varepsilon$;
- For any $a \subset \mathcal{X}$ s.t. $0 \in a$, $2\mathbb{1}_a - 1$ will predict 0 as positive and thus will predict any point drawn from $\mathcal{D}_{i^*}$ as positive. Hence $\mathcal{L}^{\text{str}}(2\mathbb{1}_a - 1) = 6\varepsilon$;
- For any $a \subset [n]$ s.t. $\exists i \neq i^*$, $i \in a$, we have $\mathcal{L}^{\text{str}}(2\mathbb{1}_a - 1) \geq 3\varepsilon$. This is due to that when $y = -1$, the probability of drawing a permutation $\sigma$ with $\sigma^{-1}(i) < \sigma^{-1}(i^*)$ is $\frac{1}{2}$. In this case, we have $i \in s_{\sigma,i^*}$ and the prediction of $2\mathbb{1}_a - 1$ is $+1$.

Under distribution $\mathcal{D}_{i^*}$, if we are able to find a (possibly randomized) $f_{\text{out}}$ with strategic loss $\mathcal{L}^{\text{str}}(f_{\text{out}}) \leq \varepsilon$, then we have $\mathcal{L}^{\text{str}}(f_{\text{out}}) = \mathbb{E}_{h \sim f_{\text{out}}}[\mathcal{L}^{\text{str}}(h)] \geq \Pr_{h \sim f_{\text{out}}}(h \neq 2\mathbb{1}_{\{i^*\}} - 1) \cdot 3\varepsilon$. Thus, $\Pr_{h \sim f_{\text{out}}}(h = 2\mathbb{1}_{\{i^*\}} - 1) \geq \frac{2}{3}$ and then, we can identify $i^*$ by checking which realization of $f_{\text{out}}$

has probability greater than $\frac{2}{3}$. In the following, we will focus on the sample complexity to identify the target function $2\mathbb{1}_{\{i^*\}} - 1$ or simply $i^*$. Let $i_{\text{out}}$ denote the algorithm's answer to question of "what is $i^*$?".

**Smoothing the data distribution** For technical reasons (appearing later in the analysis), we don't want to analyze distribution $\{\mathcal{D}_i | i \in [n]\}$ directly as the probability of $\Delta_t = i^*$ is 0 when $f_t(i^*) = +1$. Instead, we consider the mixture of $\mathcal{D}_i$ and another distribution $\mathcal{D}''_i$ to make the probability of $\Delta_t = i^*$ be a small positive number. More specifically, let $\mathcal{D}'_i = (1-p)\mathcal{D}_i + p\mathcal{D}''_i$, where $\mathcal{D}''_i$ is defined by drawing $(0, \mathcal{X}, +1)$ with probability $1-6\varepsilon$ and $(0, \{0, i\}, -1)$ with probability $6\varepsilon$. When $p$ is extremely small, we will never sample from $\mathcal{D}''_i$ when time horizon $T$ is not too large and therefore, the algorithm behaves the same under $\mathcal{D}'_i$ and $\mathcal{D}_i$. For any data distribution $\mathcal{D}$, let $\mathbf{P}_{\mathcal{D}}$ be the dynamics of $(x_1, f_1, \Delta_1, y_1, \widehat{y}_1, \ldots, x_T, f_T, \Delta_T, y_T, \widehat{y}_T)$ under $\mathcal{D}$. According to Lemma 4, by setting $p = \frac{\varepsilon}{16n^2}$, when $T \leq \frac{n}{\varepsilon}$, we have that for any $i, j \in [n]$

$$\left|\mathbf{P}_{\mathcal{D}_i}(i_{\text{out}} = j) - \mathbf{P}_{\mathcal{D}'_i}(i_{\text{out}} = j)\right| \leq \frac{1}{8}\,. \tag{21}$$

From now on, we only consider distribution $\mathcal{D}'_i$ instead of $\mathcal{D}_i$. The readers might have the question that why not using $\mathcal{D}'_i$ for construction directly. This is because no hypothesis has zero loss under $\mathcal{D}'_i$, and thus $\mathcal{D}'_i$ does not satisfy realizability requirement.

**Information gain from different choices of $f_t$** In each round of interaction, the learner picks a predictor $f_t$, which can be out of $\mathcal{H}$. Suppose that the target function is $2\mathbb{1}_{\{i^*\}} - 1$ . Here we enumerate all choices of $f_t$ and discuss how much we can learn from each choice.

- $f_t = 2\mathbb{1}_{\emptyset} - 1$ predicts all points in $\mathcal{X}$ by negative. No matter what $i^*$ is, we will observe $\Delta_t = x_t = 0$, $y_t \sim \text{Rad}(1-6\varepsilon)$, $\widehat{y}_t = -1$. They are identically distributed for any $i^* \in [n]$ and thus we cannot tell any information of $i^*$ from this round.
- $f_t = 2\mathbb{1}_{a_t} - 1$ for some $a_t \subset \mathcal{X}$ s.t. $0 \in a_t$. Then no matter what $i^*$ is, we will observe $\Delta_t = x_t = 0$, $y_t \sim \text{Rad}(1-6\varepsilon)$, $\widehat{y}_t = +1$. Again, we cannot tell any information of $i^*$ from this round.
- $f_t = 2\mathbb{1}_{a_t} - 1$ for some some non-empty $a_t \subset [n]$. For rounds with $y_t = +1$, we have $x_t = 0, \widehat{y}_t = +1$ and $\Delta_t = \Delta(0, f_t, \mathcal{X}) \sim \text{Unif}(a_t)$, which still do not depend on $i^*$. For rounds with $y_t = -1$, if the drawn example $(0, \{0\} \cup s, -1)$ satisfies that $s \cap a_t \neq \emptyset$, the we would observe $\Delta_t \in a_t$ and $\widehat{y}_t = +1$. At least we could tell that $\mathbb{1}_{\{\Delta_t\}}$ is not the target function. Otherwise, we would observe $\Delta_t = x_t = 0$ and $\widehat{y}_t = -1$.

Therefore, we can only gain some information about $i^*$ at rounds in which $f_t = 2\mathbb{1}_{a_t} - 1$ for some non-empty $a_t \subset [n]$ and $y_t = -1$. In such rounds, under distribution $\mathcal{D}'_{i^*}$, the distribution of $\Delta_t$ is described as follows. Let $k_t = |a_t|$ denote the cardinality of $a_t$. Recall that agent $(0, \{0\} \cup s, -1)$ breaks ties randomly when choosing $\Delta_t$ if there are multiple elements in $a_t \cap s$. Here are two cases: $i^* \in a_t$ and $i^* \notin a_t$.

1. The case of $i^* \in a_t$: With probability $p$, we are sampling from $\mathcal{D}''_{i^*}$ and then $\Delta_t = i^*$. With probability $1-p$, we are sampling from $\mathcal{D}_{i^*}$. Conditional on this, with probability $\frac{1}{k_t}$, we sample an agent $(0, \{0\} \cup s_{\sigma, i^*}, -1)$ with the permutation $\sigma$ satisfying that $\sigma^{-1}(i^*) < \sigma^{-1}(j)$ for all $j \in a_t \setminus \{i^*\}$ and thus, $\Delta_t = 0$. With probability $1 - \frac{1}{k_t}$, there exists $j \in a_t \setminus \{i^*\}$ s.t. $\sigma^{-1}(j) < \sigma^{-1}(i^*)$ and $\Delta_t \neq 0$. Since all $j \in a_t \setminus \{i^*\}$ are symmetric, we have $\Pr(\Delta_t = j) = (1-p)(1 - \frac{1}{k_t}) \cdot \frac{1}{k_t - 1} = \frac{1-p}{k_t}$. Hence, the distribution of $\Delta_t$ is
$$\Delta_t = \begin{cases} j & \text{w.p. } \frac{1-p}{k_t} \text{ for } j \in a_t, j \neq i^* \\ i^* & \text{w.p. } p \\ 0 & \text{w.p. } \frac{1-p}{k_t}\,. \end{cases}$$
We denote this distribution by $P_{\in}(a_t, i^*)$.

2. The case of $i^* \notin a_t$: With probability $p$, we are sampling from $\mathcal{D}''_{i^*}$, we have $\Delta_t = x_t = 0$. With probability $1-p$, we are sampling from $\mathcal{D}_{i^*}$. Conditional on this, with probability of $\frac{1}{k_t+1}$, $\sigma^{-1}(i^*) < \sigma^{-1}(j)$ for all $j \in a_t$ and thus, $\Delta_t = x_t = 0$. With probability $1 - \frac{1}{k_t+1}$ there exists $j \in a_t$ s.t. $\sigma^{-1}(j) < \sigma^{-1}(i^*)$ and $\Delta_t \in a_t$. Since all $j \in a_t$ are symmetric, we have $\Pr(\Delta_t = j) = (1-p)(1-\frac{1}{k_t+1}) \cdot \frac{1}{k_t} = \frac{1-p}{k_t+1}$. Hence the distribution of $\Delta_t$ is

$$\Delta_t = \begin{cases} j & \text{w.p. } \frac{1-p}{k_t+1} \text{ for } j \in a_t \\ 0 & \text{w.p. } p + \frac{1-p}{k_t+1}\,. \end{cases}$$

We denote this distribution by $P_{\notin}(a_t)$.

To measure the information obtained from $\Delta_t$, we will use the KL divergence of the distribution of $\Delta_t$ under the data distribution $\mathcal{D}'_{i^*}$ from that under a benchmark data distribution. We use the average distribution over $\{\mathcal{D}'_i | i \in [n]\}$, which is denoted by $\overline{\mathcal{D}} = \frac{1}{n}\sum_{i\in n} \mathcal{D}'_i$. The sampling process is equivalent to drawing $i^* \sim \mathrm{Unif}([n])$ first and then sampling from $\mathcal{D}'_{i^*}$. Under $\overline{\mathcal{D}}$, for any $j \in a_t$, we have

$$\begin{aligned}\Pr(\Delta_t = j) =& \Pr(i^* \in a_t \setminus \{j\}) \Pr(\Delta_t = j | i^* \in a_t \setminus \{j\}) + \Pr(i^* = j) \Pr(\Delta_t = j | i^* = j) \\ &+ \Pr(i^* \notin a_t) \Pr(\Delta_t = \mathbf{e}_j | i^* \notin a_t) \\ &= \frac{k_t - 1}{n} \cdot \frac{1-p}{k_t} + \frac{1}{n} \cdot p + \frac{n-k_t}{n} \cdot \frac{1-p}{k_t+1} = \frac{(nk_t-1)(1-p)}{nk_t(k_t+1)} + \frac{p}{n}\,,\end{aligned}$$

and

$$\begin{aligned}\Pr(\Delta_t = 0) &= \Pr(i^* \in a_t) \Pr(\Delta_t = 0 | i^* \in a_t) + \Pr(i^* \notin a_t) \Pr(\Delta_t = 0 | i^* \notin a_t) \\ &= \frac{k_t}{n} \cdot \frac{1-p}{k_t} + \frac{n-k_t}{n} \cdot (p + \frac{1-p}{k_t+1}) = \frac{(n+1)(1-p)}{n(k_t+1)} + \frac{(n-k_t)p}{n}\,.\end{aligned}$$

Thus, the distribution of $\Delta_t$ under $\overline{\mathcal{D}}$ is

$$\Delta_t = \begin{cases} j & \text{w.p. } \frac{(nk_t-1)(1-p)}{nk_t(k_t+1)} + \frac{p}{n} \text{ for } j \in a_t \\ 0 & \text{w.p. } \frac{(n+1)(1-p)}{n(k_t+1)} + \frac{(n-k_t)p}{n}\,. \end{cases}$$

We denote this distribution by $\overline{P}(a_t)$. Next we will compute the KL divergence of $P_{\notin}(a_t)$ and $P_{\in}(a_t)$ from $\overline{P}(a_t)$. Since $p = \frac{\varepsilon}{16n^2} \le \frac{1}{16n^2}$, we have $\frac{(nk_t-1)(1-p)}{nk_t(k_t+1)} + \frac{p}{n} \le \frac{1-p}{k_t+1}$ and $\frac{(n+1)(1-p)}{n(k_t+1)} + \frac{(n-k_t)p}{n} \le \frac{1}{k_t} + p$. We will also use $\log(1+x) \le x$ for $x \ge 0$ in the following calculation. For any $i^* \in a_t$, we have

$$\begin{aligned}&\mathrm{D}_{\mathrm{KL}}(\overline{P}(a_t) \| P_{\in}(a_t, i^*)) \\ =&(k_t - 1)\left(\frac{(nk_t-1)(1-p)}{nk_t(k_t+1)} + \frac{p}{n}\right) \log\left((\frac{(nk_t-1)(1-p)}{nk_t(k_t+1)} + \frac{p}{n}) \cdot \frac{k_t}{1-p}\right) \\ &+ \left(\frac{(nk_t-1)(1-p)}{nk_t(k_t+1)} + \frac{p}{n}\right) \log\left((\frac{(nk_t-1)(1-p)}{nk_t(k_t+1)} + \frac{p}{n}) \cdot \frac{1}{p}\right) \\ &+ \left(\frac{(n+1)(1-p)}{n(k_t+1)} + \frac{(n-k_t)p}{n}\right) \log\left(\left(\frac{(n+1)(1-p)}{n(k_t+1)} + \frac{(n-k_t)p}{n}\right) \cdot \frac{k_t}{1-p}\right) \\ \le&(k_t - 1)\left(\frac{(nk_t-1)(1-p)}{nk_t(k_t+1)} + \frac{p}{n}\right) \log(\frac{1-p}{k_t+1} \cdot \frac{k_t}{1-p}) + \frac{1-p}{k_t+1} \log(1 \cdot \frac{1}{p}) \\ &+ (\frac{1}{k_t} + p) \cdot \log(1 + pk_t) \\ \le& 0 + \frac{1}{k_t+1} \log(\frac{1}{p}) + \frac{2}{k_t} \cdot pk_t = \frac{1}{k_t+1} \log(\frac{1}{p}) + 2p\,. \end{aligned} \tag{22}$$

For $P_{\notin}(a_t)$, we have

$$
\begin{aligned}
&\mathrm{D_{KL}}(\overline{P}(a_t)\|P_{\notin}(a_t))\\
=&k_t\left(\frac{(nk_t-1)(1-p)}{nk_t(k_t+1)}+\frac{p}{n}\right)\log\left((\frac{(nk_t-1)(1-p)}{nk_t(k_t+1)}+\frac{p}{n})\cdot\frac{k_t+1}{1-p}\right)\\
&+\left(\frac{(n+1)(1-p)}{n(k_t+1)}+\frac{(n-k_t)p}{n}\right)\log\left(\left(\frac{(n+1)(1-p)}{n(k_t+1)}+\frac{(n-k_t)p}{n}\right)\cdot\frac{1}{p+\frac{1-p}{k_t+1}}\right)\\
\leq&k_t\left(\frac{(nk_t-1)(1-p)}{nk_t(k_t+1)}+\frac{p}{n}\right)\log(\frac{1-p}{k_t+1}\cdot\frac{k_t+1}{1-p})\\
&+(\frac{1}{k_t}+p)\log\left(\left(\frac{(n+1)(1-p)}{n(k_t+1)}+\frac{(n-k_t)p}{n}\right)\cdot\frac{1}{p+\frac{1-p}{k_t+1}}\right)\\
=&0+(\frac{1}{k_t}+p)\log(1+\frac{1-p(k_t^2+k_t+1)}{n(1+k_tp)})\\
\leq&(\frac{1}{k_t}+p)\frac{1}{n(1+k_tp)}=\frac{1}{nk_t}\,.
\end{aligned}
\tag{23}
$$

**Lower bound of the information** Now we adopt the similar framework used in the proofs of Theorem 7 and 8. For notation simplicity, for all $i\in[n]$, let $\mathbf{P}_i$ denote the dynamics of $(x_1,f_1,\Delta_1,y_1,\widehat{y}_1,\ldots,x_T,f_T,\Delta_T,y_T,\widehat{y}_T)$ under $\mathcal{D}'_i$ and and $\overline{\mathbf{P}}$ denote the dynamics under $\overline{\mathcal{D}}$. Let $B_t$ denote the event of $\{f_t=2\mathbb{1}_{a_t}-1$ for some non-empty $a_t\subset[n]\}$. As discussed before, for any $a_t$, conditional on $\neg B_t$ or $y_t=+1$, $(x_t,\Delta_t,y_t,\widehat{y}_t)$ are identical in all $\{\mathcal{D}'_i|i\in[n]\}$, and therefore, also identical in $\overline{\mathcal{D}}$. We can only obtain information at rounds when $B_t\wedge(y_t=-1)$ occurs. In such rounds, we know that $x_t$ is always 0, $f_t$ is fully determined by history (possibly with external randomness , which does not depend on data distribution), $y_t=-1$ and $\widehat{y}_t$ is fully determined by $\Delta_t$ ($\widehat{y}_t=+1$ iff. $\Delta_t\neq 0$).

Therefore, conditional the history $H_{t-1}=(x_1,f_1,\Delta_1,y_1,\widehat{y}_1,\ldots,x_{t-1},f_{t-1},\Delta_{t-1},y_{t-1},\widehat{y}_{t-1})$ before time $t$, we have

$$
\begin{aligned}
&\mathrm{D_{KL}}(\overline{\mathbf{P}}(x_t,f_t,\Delta_t,y_t,\widehat{y}_t|H_{t-1})\|\mathbf{P}_i(x_t,f_t,\Delta_t,y_t,\widehat{y}_t|H_{t-1}))\\
=&\overline{\mathbf{P}}(B_t\wedge y_t=-1)\mathrm{D_{KL}}(\overline{\mathbf{P}}(\Delta_t|H_{t-1},B_t\wedge y_t=-1)\|\mathbf{P}_i(\Delta_t|H_{t-1},B_t\wedge y_t=-1))\\
=&6\varepsilon\overline{\mathbf{P}}(B_t)\mathrm{D_{KL}}(\overline{\mathbf{P}}(\Delta_t|H_{t-1},B_t\wedge y_t=-1)\|\mathbf{P}_i(\Delta_t|H_{t-1},B_t\wedge y_t=-1))\,,
\end{aligned}
\tag{24}
$$

where the last equality holds due to that $y_t\sim\mathrm{Rad}(1-6\varepsilon)$ and does not depend on $B_t$.

For any algorithm that can successfully identify $i$ under the data distribution $\mathcal{D}_i$ with probability $\frac{3}{4}$ for all $i\in[n]$, then $\mathbf{P}_{\mathcal{D}_i}(i_{\mathrm{out}}=i)\geq\frac{3}{4}$ and $\mathbf{P}_{\mathcal{D}_j}(i_{\mathrm{out}}=i)\leq\frac{1}{4}$ for all $j\neq i$. Recall that $\mathcal{D}_i$ and $\mathcal{D}'_i$ are very close when the mixture parameter $p$ is small. Combining with Eq (21), we have

$$
\begin{aligned}
&|\mathbf{P}_i(i_{\mathrm{out}}=i)-\mathbf{P}_j(i_{\mathrm{out}}=i)|\\
\geq&\left|\mathbf{P}_{\mathcal{D}_i}(i_{\mathrm{out}}=i)-\mathbf{P}_{\mathcal{D}_j}(i_{\mathrm{out}}=i)\right|-|\mathbf{P}_{\mathcal{D}_i}(i_{\mathrm{out}}=i)-\mathbf{P}_i(i_{\mathrm{out}}=i)|-\left|\mathbf{P}_{\mathcal{D}_j}(i_{\mathrm{out}}=i)-\mathbf{P}_j(i_{\mathrm{out}}=i)\right|\\
\geq&\frac{1}{2}-\frac{1}{4}=\frac{1}{4}\,.
\end{aligned}
$$

Then we have the total variation distance between $\mathbf{P}_i$ and $\mathbf{P}_j$

$$
\mathrm{TV}(\mathbf{P}_i,\mathbf{P}_j)\geq|\mathbf{P}_i(i_{\mathrm{out}}=i)-\mathbf{P}_j(i_{\mathrm{out}}=i)|\geq\frac{1}{4}\,.
\tag{25}
$$

Then we have

$$
\begin{aligned}
&\mathbb{E}_{i\sim\mathrm{Unif}([n])}\left[\mathrm{TV}^2(\mathbf{P}_i,\mathbf{P}_{(i+1)\bmod n})\right] \le 4\mathbb{E}_{i\sim\mathrm{Unif}([n])}\left[\mathrm{TV}^2(\mathbf{P}_i,\overline{\mathbf{P}})\right]\\
\le& 2\mathbb{E}_i\left[\mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}\|\mathbf{P}_i)\right] && \text{(Pinsker's ineq)}\\
=& 2\mathbb{E}_i\left[\sum_{t=1}^T \mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(x_t,f_t,\Delta_t,y_t,\widehat{y}_t|H_{t-1})\|\mathbf{P}_i(x_t,f_t,\Delta_t,y_t,\widehat{y}_t|H_{t-1}))\right] && \text{(Chain rule)}\\
\le& 12\varepsilon\mathbb{E}_i\left[\sum_{t=1}^T \overline{\mathbf{P}}(B_t)\mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(\Delta_t|H_{t-1},B_t\wedge y_t=-1)\|\mathbf{P}_i(\Delta_t|H_{t-1},B_t\wedge y_t=-1))\right]\\
& && \text{(Apply Eq (24))}\\
\le& \frac{12\varepsilon}{n}\sum_{t=1}^T \overline{\mathbf{P}}(B_t)\sum_{i=1}^n \mathrm{D}_{\mathrm{KL}}(\overline{\mathbf{P}}(\Delta_t|H_{t-1},B_t\wedge y_t=-1)\|\mathbf{P}_i(\Delta_t|H_{t-1},B_t\wedge y_t=-1))\\
=& \frac{12\varepsilon}{n}\mathbb{E}_{a_{1:T}\sim\overline{\mathbf{P}}}\left[\sum_{t=1}^T \mathbb{1}(B_t)\left(\sum_{i:i\in a_t}\mathrm{D}_{\mathrm{KL}}(\overline{P}(a_t)\|P_{\in}(a_t))+\sum_{i:i\notin a_t}\mathrm{D}_{\mathrm{KL}}(\overline{P}(a_t)\|P_{\notin}(a_t))\right)\right]\\
\le& \frac{12\varepsilon}{n}\mathbb{E}_{a_{1:T}\sim\overline{\mathbf{P}}}\left[\sum_{t:\mathbb{1}(B_t)=1}\left(\sum_{i:i\in a_t}\left(\frac{1}{k_t+1}\log(\frac{1}{p})+2p\right)+\sum_{i:i\notin a_t}\frac{1}{nk_t}\right)\right] && \text{(Apply Eq (22),(23))}\\
\le& \frac{12\varepsilon}{n}\sum_{t=1}^T(\log(\frac{1}{p})+2np+1)\\
\le& \frac{12T\varepsilon(\log(16n^2/\varepsilon)+2)}{n}\,.
\end{aligned}
$$

Combining with Eq (25), we have that there exists a universal constant $c$ such that $T \ge \frac{cn}{\varepsilon(\log(n/\varepsilon)+1)}$.

$\square$