Near-Polynomially Competitive Active Logistic Regression

Yihan Zhou
The University of Texas at Austin

Eric Price
The University of Texas at Austin

Trung Nguyen
The University of Texas at Austin

Abstract

We address the problem of active logistic regression in the realizable setting. It is well known that active learning can require exponentially fewer label queries compared to passive learning, in some cases using $\log\frac{1}{\varepsilon}$ rather than $\mathrm{poly}(1/\varepsilon)$ labels to get error ε larger than the optimum.

We present the first algorithm that is polynomially competitive with the optimal algorithm on every input instance, up to factors polylogarithmic in the error and domain size. In particular, if any algorithm achieves label complexity polylogarithmic in ε, so does ours. Our algorithm is based on efficient sampling and can be extended to learn more general class of functions. We further support our theoretical results with experiments demonstrating performance gains for logistic regression compared to existing active learning algorithms.

1 INTRODUCTION

Active learning is a learning paradigm where unlabeled data is abundant and inexpensive, but obtaining labels is costly or time-consuming. The goal is to use as few labeled examples as possible to train an effective model. Unlike passive learning, where the learner receives a fixed set of labeled data, active learning allows the learner to choose which data points to query for labels. It is known that for binary classifiers, active learning algorithms can achieve exponentially better label complexity bounds than passive algorithms in some cases (Cohn et al., 1994). In this paper, we study active logistic regression, which is a special case of active probabilistic classification. Active probabilistic classification extends binary classification by allowing each hypothesis to provide a labeling probability instead of a deterministic label. Formally, the problem is defined as follows.

1.1 Problem Definition and Motivation

Let $X \subseteq \mathbb{R}^d$ be a finite dataset, $Y = \{0,1\}$ be the set of labels and H be a hypothesis class containing hypotheses $h : X \to [0,1]$. In other words, for each $x \in X$ and $h \in H$, $h(x)$ denote the probability that x has the label 1. Let $\mathcal{D}_X$ be a distribution on X and then we use the weighted ℓ_2-distance $\|h_1 - h_2\|_2^{\mathcal{D}_X} := \sqrt{\mathbb{E}_{x\sim\mathcal{D}_X}\left[(h_1(x) - h_2(x))^2\right]}$ with respect to the distribution $\mathcal{D}_X$ to measure the distance between two hypotheses h_1 and h_2. Usually, the distribution $\mathcal{D}_X$ is clear in context and we can simplify the notation by dropping the superscript. There is a ground truth $h^* \in H$ defines the true marginal distribution, i.e., $\Pr[x \text{ has label } 1] = h^*(x)$ for every $x \in X$. We define the error of a hypothesis $h \in H$ by $\mathrm{err}(h) := \|h - h^*\|_2$. We define the problem of realizable active probabilistic classification.

Definition 1.1 (Realizable Active Probabilistic Classification)**.** *Given a finite dataset X and a marginal distribution $\mathcal{D}_X$ over X. The environment chooses a hidden hypothesis $h^* \in H$ and provides an oracle $\mathcal{O}_{h^*} : X \to \{0,1\}$ such that for any query $x \in X$, $\mathcal{O}_{h^*}(x)$ returns 1 with probability $h^*(x)$ and 0 otherwise. The player knows X and $\mathcal{D}_X$ and can query the oracle with any $x \in X$ multiple times. The player's goal is to identify a hypothesis $\hat{h} \in H$ with $\mathrm{err}(\hat{h}) \leqslant \varepsilon$ with probability at least $1-\delta$, using as few queries as possible.*

We use a tuple $P := (X, \mathcal{D}_X, H, \varepsilon, \delta)$ to represent a problem instance. We refer to the number of queries made to the oracle as the *label complexity*. We could characterize the hardness of a specific problem instance by the optimal label complexity, defined in the following.

Definition 1.2 (Optimal Label Complexity)**.** *Given a problem instance $P = (X, \mathcal{D}_X, H, \varepsilon, \delta)$, we say that an algorithm A solves P with $m_A(P)$ queries if, for every $h^* \in H$, the algorithm A uses at most $m_A(P)$ queries and, with probability at least $1-\delta$, returns a hypothesis $\hat{h}$ such that $\mathrm{err}\left(\hat{h}\right) < \varepsilon$. The optimal label complexity of P is the minimal value of $m_A(P)$ over all algorithms A that solve P, denoted by $m^*(P)$.*

In this paper, we focus on a special case of this problem–

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

active logistic regression. In logistic regression, each hypothesis h is parameterized by some $\theta \in \mathbb{R}^d$ such that $h(x) = \sigma\left(\theta^T x\right)$, where $\sigma(a) = \frac{1}{1+e^{-a}}$ is the sigmoid function. To make the connection, we use h_θ to denote the hypothesis h parameterized by θ and use θ_h to denote the vector θ parameterizing h. Similarly, we use H_Θ to denote the hypothesis class parameterized by a class of vectors Θ and Θ_H to denote the class of vectors parameterizing the hypothesis class H. We call Θ_H the parameter space and drop the subscript when the context is clear. Throughout this paper, we make the following standard boundedness assumption.

Assumption 1.3 (Boundedness Assumption). *In logistic regression, both of the dataset and parameter space is bounded, more precisely,*

1. The parameter space is upper bounded by $R_1 \geqslant 1$, i.e., $\|\theta\|_2 \leqslant R_1$ for every $\theta \in \Theta$.

2. The dataset X is upper bounded by $R_2 \geqslant 1$, i.e., $\|x\|_2 \leqslant R_2$ for every $x \in X$.

Similar to the improvement active learners shows on learning binary classifiers, active logistic regression algorithms can significantly outperform their passive counterparts, sometimes by exponential factors, as demonstrated in the following example.

Example 1.4. *Let $X = \{0, 1\}$ and let $\mathcal{D}_X$ be the distribution where 0 occurs with probability $1 - \varepsilon'$ and 1 occurs with probability ε'. Let $R_1 = 10$, set $\varepsilon = \frac{\varepsilon'}{4}$, and choose δ arbitrarily.*

In logistic regression, all hypotheses predict the same value for $x = 0$, providing no additional information from querying this point. However, the predictions for $x = 1$ can vary within the range $[\sigma(-10), \sigma(10)] \supseteq \left[\frac{1}{4}, \frac{3}{4}\right]$. To achieve an error tolerance of ε, a learner must estimate the prediction for $x = 1$ within this constant error bound. A passive learner needs to sample $\Omega\left(\frac{1}{\varepsilon}\right)$ times to observe $x = 1$ with sufficient frequency. In contrast, an active learner can directly query $x = 1$ and estimate its label probability within a constant error margin, requiring only $O(1)$ queries.

1.2 Our Results

Our paper has the following contributions:

1. We present the first active logistic regression algorithm with a provable competitive label complexity upper bound, as shown below.

Theorem 1.5. *Let*

$$m = m^*\left(X, \mathcal{D}_X, H, \frac{\varepsilon^2}{16\sqrt{2}dR_1R_2}, 0.01\right).$$

Under Assumption 1.3, Algorithm 4 returns a hypothesis $\hat{h}$ such that $\operatorname{err}\left(\hat{h}\right) \leqslant 17\varepsilon$ *with probability at least* 0.7*, using a label complexity of*

$$O\left(\text{poly}\,(m)\ \text{polylog}\left(\frac{R_1R_2}{\varepsilon}\right)\right).$$

This bound implies that Algorithm 4 achieves a label complexity that is polynomially competitive with the optimal on any problem instance, up to some polylogarithmic factors in the accuracy and domain size. Furthermore, it demonstrates an exponential improvement over passive algorithms on certain instances, such as Example 1.4.

2. Our algorithm is simple and can be efficiently implemented. In Section 9, we conduct experiments demonstrating its performance, showing our algorithm has potential in real-life application.

3. Our algorithm and analysis can be extended to a wider class of probabilistic binary classifiers, including the exponential family. This extension is discussed in Section 8.

2 RELATED WORK

To the best of our knowledge, our result provides the first known competitive label complexity upper bounds for active logistic regression methods. Our method is adaptive, which means that each query can be chosen based on the outcomes of previous queries. Related works in this area fall into four categories: (i) theory for *passive* logistic regression; (ii) theory for *non-adaptive* active logistic regression; (iii) theory for active learning methods *other than* logistic regression; and (iv) *empirical* results for active logistic regression.

Logistic regression and, more broadly, generalized linear models (GLMs) have been extensively studied in the passive setting. Efficient algorithms for learning GLMs with small ℓ_2 error include ISOTRON (Kalai and Sastry, 2009), GLMtron (Kakade et al., 2011), and Sparsitron (Klivans and Meka, 2017). One can also estimate the parameters under distributional assumptions (Hsu and Mazumdar, 2024). All the aforementioned algorithms are designed for passive learning; therefore, as shown in Example 1.4, they could be exponentially worse than active learning algorithms in some problem instances.

There is a line of work (Munteanu et al., 2018; Mai et al., 2021; Gajjar et al., 2023, 2024; Chowdhury and Ramuhalli, 2024) that employs non-adaptive sampling methods—such as leverage score sampling or Lewis weights—to solve logistic regression, and these techniques can be extended to active learning. However, their results are not directly comparable to ours due to differences in both the setting and the error measures. Specifically, they assume that the dataset X and labels y are fixed in advance, so the labels do not exhibit randomness, and the error bounds they derive are neither simply additive (Gajjar et al., 2023, 2024; Chowdhury and Ramuhalli, 2024) nor measured in the ℓ_2-

distance (Munteanu et al., 2018; Mai et al., 2021). In addition, while non-adaptive methods offer the advantage of faster implementation, their lack of adaptivity makes them unlikely to achieve the near-optimal competitive bounds obtained by our approach.

In the active setting, the theory and algorithms for binary classifiers—where each hypothesis maps every x to a label deterministically—is also well-developed. One class of such algorithms is called disagreement-based active learning, which involves only sampling from the disagreement region, where there exist two hypotheses that have different label predictions (Cohn et al., 1994; Balcan et al., 2006; Hanneke, 2007; Dasgupta et al., 2007). However, in the setting of probabilistic classification, it is not even clear how to define the notion of a disagreement region, so this class of algorithms does not apply. Another class of active learning algorithms is called splitting-based algorithms, where the algorithm quantifies the informativeness of each point and queries the most informative ones (Dasgupta, 2004; Katz-Samuels et al., 2021; Price and Zhou, 2023). Our algorithm falls into this category and can be seen as an extension from deterministic to probabilistic binary prediction. Our algorithm and analysis draw inspiration from the work of Price and Zhou (2023), who developed an algorithm with competitive label complexity bounds for active binary classification. In essence, their approach employs the multiplicative weights framework, which assigns a prior over the hypothesis space. At each iteration, the algorithm selects a set of the most informative points—determined by the current prior—and penalizes hypotheses that yield incorrect predictions. Their analysis shows that, after a sufficient number of queries, the posterior distribution concentrates on the ground truth. However, their method is limited to finite binary hypothesis classes and is not directly applicable to the infinite hypothesis spaces encountered in logistic regression, nor does it naturally extend to probabilistic settings. In contrast, our work overcomes these limitations by adapting the approach to active logistic regression and more general function classes. It is also worth noting that active regression is closely related to active probabilistic classification. However, most existing work in this area—for example, Sabato and Munos (2014); Chen and Price (2019); Musco et al. (2022)—has focused on linear regression, and therefore does not extend to logistic regression.

For our setting of active logistic regression, Yang and Loog (2018) conducted a comprehensive survey of various active learning algorithms and heuristics, benchmarking their empirical performance. However, none of the algorithms considered have label complexity bounds or mathematically rigorous performance guarantees for logistic regression.

3 ALGORITHM: FIRST ATTEMPT

We begin to introduce our algorithm in this section. All the omitted proofs in the paper can be found in Supplementary Materials Section A. In our initial approach, we try to apply the multiplicative weights framework directly. We start by assigning an initial weight $w_1(h_\theta) = 1$ to every hypothesis h_θ parameterized by $\theta \in \Theta$, where $\Theta \subseteq \mathbb{R}^n$ is equipped with the Lebesgue measure. Normalizing these weights with respect to the Lebesgue measure yields a prior distribution λ_1 over the hypothesis space H. Correspondingly, λ_1 is also a uniform distribution over the parameter space Θ. At each iteration i, given the current prior λ_i, we define the informativeness of each point $x \in X$ as follows:

$$r_{\lambda_i}(x) := \mathbb{E}_{h\sim\lambda_i}\left[\mathrm{D}_{\mathrm{KL}}\left(\bar{h}_{\lambda_i}(x) \,\|\, h(x)\right)\right],$$

where $\bar{h}_{\lambda_i}(x) = \mathbb{E}_{h\sim\lambda_i}[h(x)]$ is the average prediction under λ_i, and $\mathrm{D}_{\mathrm{KL}}(p\|q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$ is the binary Kullback-Leibler divergence (KL divergence). This function $r_{\lambda_i}(x)$ measures the expected KL divergence between the average prediction $\bar{h}_{\lambda_i}(x)$ and individual predictions $h(x)$, indicating the level of disagreement among hypotheses at point x. If the prior λ is not overly concentrated, we can relate the information function r of the most informative point to the optimal label complexity $m^*(X, \mathcal{D}_X, H, \varepsilon, 0.01)$, which we will simply denoted as m^* throughout the rest of the paper. Let $B_h(\varepsilon)$ be the ball centered at h with radius ε in the hypothesis space. This relationship is formalized in one of our core lemmas below.

Lemma 3.1 (Lower Bound for Non-concentrated Distribution)**.** *If λ is a distribution over H such that no hypothesis $h \in H$ satisfies $\lambda(B_h(2\varepsilon)) > 0.8$, then:*

$$\max_{x\in X} r_\lambda(x) \gtrsim \frac{1}{m^*(X, \mathcal{D}_X, H, \varepsilon, 0.01)}.$$

Note that computing $r_{\lambda_i}(x)$ exactly is computationally intensive due to the expectation over λ_i. Instead, we approximate it by sampling hypotheses h from λ_i and estimating $r_{\lambda_i}(x)$ using these samples. After the estimation, we query the most informative point x_i, i.e., the one with the highest estimated $r_{\lambda_i}(x)$, and obtain the corresponding label y_i. For each query-label pair (x_i, y_i) and hypothesis h, we use the cross-entropy loss as the penalty:

$$\ell_h(x_i, y_i) = y_i \log\frac{1}{h(x_i)} + (1 - y_i)\log\frac{1}{1 - h(x_i)}.$$

We update the weight of each hypothesis h_θ as:

$$w_{i+1}(h_\theta) := w_i(h_\theta) \cdot \exp\left(-\ell_{h_\theta}(x_i, y_i)\right).$$

Normalizing the weights gives an updated probability density function (PDF) of the distribution λ_{i+1}, but we don't have this normalization step in our algorithm because it is costly and unnecessary. As more queries are made, we expect the distribution λ_i to concentrate around the true hypothesis h^*. Therefore, at the end, we simply sample $\hat{h}$ from the final distribution λ_K. This algorithm is summarized in Algorithm 1.

Algorithm 1: Active Logistic Regression: First Attempt

```
Algorithm ActiveSimple$(P, K)$
    Initialize $w_1(h_\theta) = 1$ for every $\theta \in \Theta$
    for $i = 1$ to $K$ do
        Estimate
          $r_{\lambda_i}(x) = \mathbb{E}_{h\sim\lambda_i}\left[\mathrm{D}_{\mathrm{KL}}\left(\bar{h}_{\lambda_i}(x) \,\|\, h(x)\right)\right]$ for
          all $x \in X$ using samples $h \sim \lambda_i$, obtaining
          estimates $\hat{r}_{\lambda_i}(x)$
        Select $x_i = \arg\max_{x\in X} \hat{r}_{\lambda_i}(x)$
        Query $x_i$ and receive label $y_i$
        Update weights:
           $w_{i+1}(h_\theta) = w_i(h_\theta) \cdot \exp\left(-\ell_{h_\theta}(x_i, y_i)\right)$
          for all $h_\theta \in H$
    end
    return $\hat{h} \sim \lambda_K$
```

3.1 Analysis Attempt

As mentioned earlier, we expect the distribution λ to concentrate around h^* as more queries are made. To formalize this, we define the potential $\psi_i(h^*) := \log \lambda_i(h^*)$. Ideally, we aim to relate the growth in potential to the information function r, and show that r is lower bounded, ensuring progress at each iteration. We begin by calculating the expected potential growth, as outlined in the following lemma.

Lemma 3.2. *Let λ_i be the prior distribution at iteration i, x_i be the queried point, and y_i be its label. Then the expected potential gain is*

$$\mathbb{E}_{y_i}\left[\psi_{i+1}(h^*) - \psi_i(h^*) \,|\, x_i\right] = D_{\mathrm{KL}}\left(h^*(x_i) \,\|\, \bar{h}_{\lambda_i}(x_i)\right).$$

Since the KL divergence is non-negative, we have the nice property that no matter which point we query, the expected potential growth is always non-negative. This property is useful for our analysis. However, several challenges prevent us from proving that Algorithm 1 converges to h^* quickly enough:

1. The KL divergence term depends only on the mean of the distribution. This brings a problem that even if the algorithm queries an informative point, the expected potential gain could still be low, even zero. Consider the following example. Suppose the prior λ assigns a probability of 0.01 to h^* and distributes the remaining probability evenly between hypotheses h_1 and h_2, where $h^*(x) = \frac{1}{2}$, $h_1(x) = 1$ and $h_2(x) = 0$. Here x is an informative point because most hypotheses (h_1 and h_2) disagree with h^* on x, suggesting that querying x should significantly increase $\lambda(h^*)$. However, the mean prediction under λ is $\frac{1}{2}$, matching $h^*(x)$, which results in zero expected potential gain. Thus on this example, Algorithm 1 chooses a good point to query but the potential does not grow.

2. Conversely, Algorithm 1 may also select an *uninformative* point to query, if the distribution is overconcentrated. The algorithm chooses the query point to maximize $r(x)$, which Lemma 3.1 shows how to relate to m^*. But the lemma requires that λ not be too concentrated in a small region; if our prior λ on hypotheses is highly concentrated, the lemma does not apply and we cannot show that the query point is informative because it is not true.

3. The KL divergence is unbounded, introducing complications when attempting to establish concentration inequalities and a high probability bound.

4 ALGORITHM: REFINEMENT

To address the issues described in Section 3.1, we refine our algorithm as follows.

4.1 Double Query

To address challenge 1, we modify our algorithm so that in each iteration, instead of querying x_i once, we query it twice and obtain labels y_i^1 and y_i^2. This adjustment allows us to relate the expected potential growth to the mean and the variance.

Lemma 4.1. *The expected potential growth in iteration i, conditioned on x_i being queried twice, is bounded below by*

$$\begin{aligned}&\mathop{\mathbb{E}}_{y_i^1, y_i^2}\left[\psi_{i+1}(h^*) - \psi_i(h^*) \,|\, x_i\right]\\ &\gtrsim \left(h^*(x_i) - \bar{h}_{\lambda_i}(x_i)\right)^2 + \left(\mathop{\mathrm{Var}}_{h\sim\lambda_i}[h(x_i)]\right)^2.\end{aligned}$$

In the example given in challenge 1, the variance is large, so by performing the double query, the expected potential gain is substantial, as desired. In fact, in Lemma C.5 we can lower bound this variance term by a polynomial in the information function r_{λ_i}. When λ is not overconcentrated, we can then apply Lemma 3.1 to show that the best query x has expected potential growth at least polynomial in $1/m^*$.

4.2 Sampling Procedure

To address Challenge 2 and prevent the algorithm from stalling due to overconcentration, we use the sampling procedure given in Algorithm 2. This procedure samples h_1 from one distribution p, then rejection samples h_2 from another distribution q such that $\|h_1 - h_2\| \geqslant \varepsilon$. It then uniformly randomly outputs h_1 or h_2. The resulting distribution is:

$$\hat{\lambda}(h) := \frac{1}{2}p(h) + \frac{1}{2}\mathbb{E}_{h'\sim p}\left[q^{H\setminus B_{h'}(\varepsilon)}(h)\right],$$

where $q^{H \setminus B_{h'}(\varepsilon)}$ denotes the conditional distribution of q outside the ball $B_{h'}(\varepsilon)$. It can be shown that under the distribution $\hat{\lambda}$, no ball of radius ε in the hypothesis space has probability mass more than 0.8, provided we set the parameters of the sampling procedure properly.

Sampling from Algorithm 2, we could relate the information function r to m^* as desired by Lemma 3.1. However, at the same time, this sampling procedure introduces new questions that need to be answered. How do we choose the distributions p and q? How do we relate the information function r with respect to $\hat{\lambda}$ to the potential change? These questions are indeed tricky. To answer these questions and facilitate our analysis, we use a nested loop structure in our refined algorithm. We refer to the outer loop iterations as "phases" and the inner loop iterations as "iterations". We denote the j-th iteration in phase i by (i, j).

In the refined algorithm, at the beginning of each phase i, we fix a distribution $\lambda^0 = \lambda_i$, which remains unchanged during the phase. We also initialize a distribution $p_{(i,1)}$ to be uniform over Θ at the start of the phase. Then, in iteration j, we use the sampling procedure with $p = \lambda^0$ and $q = p_{(i,j)}$ and query the most informative point $x_{(i,j)}$ with respect to the distribution $\hat{\lambda}_{(i,j)} := \frac{1}{2}\lambda^0 + \frac{1}{2}\lambda^1_{(i,j)}$, where $\lambda^1_{(i,j)} := \mathbb{E}_{h' \sim \lambda^0}\left[q^{H \setminus B_{h'}(\varepsilon)}(h)\right]$. After getting two labels, we update $p_{(i,j+1)}$ at the end of the iteration. At the end of the phase, we query all of the points $\{x_{(i,1)}, \ldots, x_{(i,M)}\}$ twice again and use the fresh labels to update λ_{i+1}. As shown later in Section 5, we expect that in most of the phases, the last iteration distribution $p_{(i,M+1)}$ concentrates around h^*, so we return $\hat{h}$ by sampling from the average of the last iteration of each phase.

4.3 Clipping

To address the unboundedness issue in Challenge 3, we clip the hypothesis class so that each hypothesis $h \in H_\gamma$ satisfies $h(x) \in [\gamma, 1-\gamma]$ for all $x \in X$ and $\gamma \in (0, \frac{1}{2})$. Specifically, we define

$$H_\gamma = \left\{h' : h'(x) = \operatorname{clip}\big(h(x), \gamma\big), \forall h \in H\right\},$$

where $\operatorname{clip}(z, \gamma) = \min\{\max\{z, \gamma\}, 1-\gamma\}$. The refined algorithm then operates on the clipped hypothesis class H_γ.

Clipping may seem problematic, particularly since h^* may not lie in H_γ. In Section 6.2, we address this issue by providing a black-box reduction from unclipped to clipped instance. Additionally, clipping does not change the parameter space, and can be applied directly to h_θ. The refined algorithm is shown in Algorithm 3.

5 ANALYSIS OF ALGORITHM 3

In this section, we provide an overview of the proof for the label complexity of Algorithm 3. In the implementation of Algorithm 3, we sample hypotheses h to estimate the informativeness function r. For the purpose of analysis, we simplify by assuming that we can compute r exactly, thereby neglecting the estimation error. This simplification is justified because, as long as we can efficiently sample h, estimating r with high accuracy is not computationally expensive. As mentioned in Section 4.3, the hypothesis class is clipped, and we keep the realizable assumption by letting the true hypothesis $h^* \in H_\gamma$. Formally, we make the following clipping assumption throughout this section.

Algorithm 2: Sampling Procedure

```
Procedure SAMPLINGPROC$(p, q, \varepsilon)$
    Sample $h_1$ according to the fixed distribution $p$
    Rejection sample $h_2$ according to $q$ until
      $\|h_1 - h_2\| \geqslant \varepsilon$
    return $h_1$ with probability $0.5$ and $h_2$ with
      probability $0.5$
```

Assumption 5.1 (Clipping Assumption). *For every $x \in X$ and $h \in H_\gamma$, it holds that $h(x) \in [\gamma, 1-\gamma]$.*

Let's also define some notations here. Let $\{\mathcal{F}_{(i,j)}\}_{i \in [K], j \in [M]}$ be a filtration and $\mathcal{F}_{(i,j)}$ be the σ-algebra of all the queried points up to (i, j), all the labels for p up to (i, j) and all the labels for λ up to the previous phase $i - 1$. Also recall that we define the potential $\psi_i(h^*) = \log \lambda_i(h^*)$. Now we analyze the potential change in a more fine-grained fashion. If in the current phase (i, j), there exists some queried point $x_{(i,j)}$ satisfies the property that sampling from $\lambda^0 = \lambda_{(i,1)}$, with high probability, $D_{\mathrm{KL}}\left(h^*\left(x_{(i,j)}\right) \| h\left(x_{(i,j)}\right)\right)$ is not small, i.e., a non-trivial proportion of hypotheses is not too close to h^* on the queried point $x_{(i,j)}$, then we could expect $\lambda_{(i,1)}(h^*)$ grows by a non-trivial amount on this iteration. This observation is formal characterized by the following lemma.

Lemma 5.2. *Let $\zeta = \frac{1}{(m^*)^4 \log^5 \frac{1}{\gamma}}$ and let $A_{(i,j)}$ be the event that*

$$\Pr_{h \sim \lambda_0}\left[D_{\mathrm{KL}}\left(h^*\left(x_{(i,j)}\right) \middle\| h\left(x_{(i,j)}\right)\right) \geqslant \zeta \middle| \mathcal{F}_{(i,j)}\right] \geqslant \frac{1}{(m^*)^4 \log^4 \frac{1}{\gamma}},$$

then

$$\mathbb{E}\left[\psi_{i+1}(h^*) - \psi_i(h^*) | \mathcal{F}_{(i,j)}, A_{(i,j)}\right] \gtrsim \frac{1}{(m^*)^{12} \log^{16} \frac{1}{\gamma}}.$$

Otherwise, on every queried point $x_{(i,j)}$ in this phase, the vast majority of h could be quite close to h^*, so the potential growth could be slow for the entire phase. Fortunately, in this case, we could show that $p_{(i,j)}(h^*)$ grows fast relative to the hypotheses that are some distance away from h^*. We introduce a new alternative potential $\tilde{\psi}$ to facilitate such intuition. Let $\tilde{p}_{(i,j)}^{H \backslash B_{h'}(2\varepsilon)}(h) = \frac{p_{(i,j)}(h)}{p_{(i,j)}\left(H \backslash B_{h'}(2\varepsilon)\right)}$. Note that $\tilde{p}_{(i,j)}^{H \backslash B_{h'}(2\varepsilon)}$ is not a proper PDF. We then define the alternative potential $\tilde{\psi}_{(i,j)}(h^*) := \mathbb{E}_{h' \sim \lambda_0}\left[\log \tilde{p}_{(i,j)}^{H \backslash B_{h'}(2\varepsilon)}(h^*)\right]$. Similarly as Lemma 4.1, we could lower bound the expected potential growth of the alternative potential in the following lemma.

Lemma 5.3. *Let*

$$\eta_{(i,j)} = \left(h^*\left(x_{(i,j)}\right) - \bar{h}_{p_{(i,j)}^{H \backslash B_{h'}(2\epsilon)}}\left(x_{(i,j)}\right)\right)^2 + \left(\operatorname*{Var}_{h \sim p_{(i,j)}^{H \backslash B_{h'}(2\epsilon)}}\left[h\left(x_{(i,j)}\right)\right]\right)^2.$$

Then the conditional expected potential growth in iteration (i, j) *is bounded below by*

$$\mathbb{E}\left[\tilde{\psi}_{(i,j+1)}(h^*) - \tilde{\psi}_{(i,j)}(h^*) \middle| \mathcal{F}_{(i,j)}\right] \gtrsim \mathop{\mathbb{E}}_{h' \sim \lambda_0}\left[\eta_{(i,j)} \middle| \mathcal{F}_{(i,j)}\right].$$

Then in this case, we could lower bound the expected growth of the alternative potential.

Lemma 5.4. *Let* $A_{(i,j)}$ *be the event defined in Lemma 5.2, then*

$$\mathbb{E}\left[\tilde{\psi}_{(i,j+1)}(h^*) - \tilde{\psi}_{(i,j)}(h^*) \middle| \mathcal{F}_{(i,j)}, \bar{A}_{(i,j)}\right] \gtrsim \frac{1}{(m^*)^4 \log^4 \frac{1}{\gamma}}.$$

Combining Lemma 5.2 and Lemma 5.4 and set the number of iterations M properly, we get the following guarantee for each phase.

Lemma 5.5 (Phase Potential Guarantee)**.** *At phase* i*, if the number of iterations is set to*

$$M = O\left(\beta + \log \frac{1}{\alpha}\right)(m^*)^8 \log^{10} \frac{1}{\gamma},$$

and the PDF of the initial distribution in this phase satisfies $p_{(i,1)}(h^*) = \alpha$*, then one of the following conditions holds:*

1. $\Pr\left[\tilde{\psi}_{(i,M+1)}(h^*) \geqslant \frac{\beta}{2}\right] \geqslant 0.9$,
2. $\mathbb{E}\left[\psi_{(i+1,1)} - \psi_{(i,1)}\right] \gtrsim \frac{1}{(m^*)^{12} \log^{16} \frac{1}{\gamma}}$.

To summarize, we've established that at each phase, either the growth of the potential is lower bounded, or in the ending iteration of the phase, the alternative potential has a high value.

We are close to completing the proof, but one issue remains: the potential is related only to the PDF of h^*, while we need to show that a small neighborhood around h^* has high probability. To address this, we take a small-radius ball B around h^* in the parameter space. It is true that if the PDF at h^* is high, then B also has high probability. In Lemma 5.5, one possible scenario is that the alternative potential of h^* is high, which implies that a small-radius ball around θ_{h*} in the parameter space has high probability. This ensures that h^* is contained within a heavy ball, as shown in the following lemma.

Lemma 5.6. *Let* $B \subseteq H$ *be inside a ball centered at* h^* *with radius less than* ε *in the hypothesis space. If* $\tilde{\psi}_{(j,i)}(B) \geqslant 10$ *with respect to any choice of* λ^0*, then there exists a ball* C *with radius* 4ε *such that* $\lambda_{(j,i)}(C) \geqslant 0.9$ *and* $h^* \in C$*.*

Therefore, we've shown that in each phase, either the potential grows by a decent amount, or the alternative potential is high in the last iteration, which implies h^* is contained in a heavy ball with radius 4ε with high probability. Then by setting the total number of phases K and total number of iterations M properly, we can show that in almost all the phases, the latter happens. As a result, sampling from the averaged distribution of the last iterations will give a hypothesis with high accuracy with high probability. The label complexity of Algorithm 3 is given as follows.

Lemma 5.7. *Let* $m = m^*\left(X, \mathcal{D}_X, H_\gamma, \varepsilon, 0.01\right)$ *and* d *be the dimension of the parameter space. Under Assumption 1.3 and Assumption 5.1, Algorithm 3 returns* $\hat{h}$ *such that* $\operatorname{err}\left(\hat{h}\right) \leqslant 8\varepsilon$ *with probability greater than* 0.8 *using* $O\left((d)^2 m^{20} \log^{26} \frac{1}{\gamma} \log^2\left(\frac{dR_1R_2m}{\varepsilon}\right)\right)$ *queries.*

6 FINAL LABEL COMPLEXITY BOUND

6.1 Dimension Reduction

Up to this point, we have overlooked an important issue. Algorithm 3 operates in a parameter space of dimension d, resulting in a label complexity that depends on d. However, this dependence can be unnecessary in certain cases. For example, if the entire set X lies within a lower-dimensional subspace, then m^* does not depend on d, and there is no need to run our algorithm in the original high-dimensional parameter space. To address this problem, we perform a dimension reduction at the outset and then execute Algorithm 3 in the reduced parameter space. The dimension reduction algorithm and the relevant lemmas are detailed in Supplementary Materials Section B. Our complete algorithm, which includes both the dimension reduction and the clipping steps, is presented in Algorithm 4.

6.2 Removing the Clipping Assumption

The final step is to eliminate the clipping assumption. By setting γ small enough, the discrepancy between clipped and unclipped hypotheses becomes negligible, as the algorithm does not sample enough points for it to matter. The

Algorithm 3: Active Learning for Logistic Regression on Clipped Instance

```
Algorithm ClippedActive($X, \mathcal{D}_X, H_\gamma, \varepsilon, \delta, K, M$)
    for phases $i = 1$ to $K$ do
        if $i = 1$ then
            Set $\lambda_1$ to be uniform over $\Theta$
            Set $\lambda^0 = \lambda_1$
        else
            Set $\lambda^0 = \lambda_{i-1}$
        end
        Initialize $p_{(i,1)}$ to be uniform over $\Theta$
        for iterations $j = 1$ to $M$ do
            /* Implementation Step */
            Sample $h$ from $\hat{\lambda}_{(i,j)}(h) := \frac{1}{2}\lambda^0(h) + \frac{1}{2}\lambda^1_{(i,j)}$ by calling SamplingProc($\lambda^0, p_{(i,j)}, 2\varepsilon$)
            Estimate $r_{\hat{\lambda}_{(i,j)}}(x) := \mathbb{E}_{h\sim\hat{\lambda}_{(i,j)}} D_{\mathrm{KL}}\left(\bar{h}_{\hat{\lambda}_{(i,j)}}(x) \,\|\, h(x)\right)$ using samples of $h$ to obtain $\tilde{r}_{\hat{\lambda}_{(i,j)}}(x)$
            Query $x_{(i,j)} := \arg\max_{x\in X} \tilde{r}_{\hat{\lambda}_{(i,j)}}(x)$ twice and obtain labels
            /* Analysis Step */
            Query $x_{(i,j)} := \arg\max_{x\in X} r_{\hat{\lambda}_{(i,j)}}(x)$ twice and obtain labels
            Update $p_{(i,j+1)}$ using the query $x_{(i,j)}$ and the labels
        end
        Query every point in $X_i := \{x_{(i,1)}, \dots, x_{(i,M)}\}$ twice again and obtain label set $Y_i$
        Update $\lambda_i$ using the query set $X_i$ and the label set $Y_i$
    end
    return $\hat{h}$ by sampling from $\bar{\lambda} := \frac{1}{K}\sum_{i=1}^{K} p_{(i,M+1)}$
```

following lemma provides a black-box reduction that relates the sample complexities of the clipped and unclipped instances.

Lemma 6.1 (Reduction from Clipped to Unclipped Instances)**.** *Assume that algorithm A can solve a clipped instance with an error tolerance of ε and a success probability greater than 0.8 using $O\left(m \operatorname{polylog}\left(\frac{1}{\gamma}\right)\right)$ labels for any γ. Then, by setting $\gamma = \frac{\varepsilon}{10m}$, algorithm A can solve the unclipped instance with an error tolerance of 2ε and a success probability of 0.7 using $O\left(m \operatorname{polylog}(m) \operatorname{polylog}\left(\frac{1}{\varepsilon}\right)\right)$ labels.*

By applying Lemma 6.1 with proper choice of γ and the results from the dimension reduction, we can bound the label complexity of Algorithm 4.

Algorithm 4: Active Learning for Logistic Regression

```
Algorithm ActiveLogisticRegression($P$)
    $V, S \leftarrow$ DimensionReduction$\left(X, \frac{\sqrt{2}}{R_2\varepsilon}, \frac{\varepsilon^2}{2}\right)$
    Project $X$ onto $S$ to obtain a new dataset $X_S$ and corresponding marginal $\mathcal{D}_{X_S}$
    Construct a new hypothesis class $H' := H_{\mathrm{Span}(V)}$
    Set $d' = \dim(S)$
    Set $\gamma = \Theta\left(\varepsilon\,(d')^{-2}\,(m^*)^{-20}\log^{-2}\left(\frac{d' R_1 R_2 m^*}{\varepsilon}\right)\right)$
    Construct $H'_\gamma = \{h' : h'(x) \leftarrow \mathrm{clip}\big(h(x), \gamma\big), \forall h \in H'\}$, where $\mathrm{clip}(z, \gamma) = \min\{\max\{z, \gamma\}, 1-\gamma\}$
    Set $K = \Theta\left(d'\,(m^*)^{12}\log^{16}\frac{1}{\gamma}\log\left(\frac{d' R_1 R_2 m^*}{\varepsilon}\right)\right)$
    Set $M = \Theta\left(d'\,(m^*)^{8}\log^{10}\frac{1}{\gamma}\log\left(\frac{d' R_1 R_2 m^*}{\varepsilon}\right)\right)$
    return ClippedActive$\big(X_S, \mathcal{D}_{X_S}, H'_\gamma, \varepsilon, \delta, K, M\big)$
```

Theorem 1.5. *Let*

$$m = m^*\left(X, \mathcal{D}_X, H, \frac{\varepsilon^2}{16\sqrt{2}dR_1R_2}, 0.01\right).$$

Under Assumption 1.3, Algorithm 4 returns a hypothesis $\hat{h}$ such that $\mathrm{err}\left(\hat{h}\right) \leqslant 17\varepsilon$ *with probability at least* 0.7*, using a label complexity of*

$$O\left(\operatorname{poly}(m)\operatorname{polylog}\left(\frac{R_1R_2}{\varepsilon}\right)\right).$$

6.3 Discussion

We address some important considerations related to Theorem 1.5.

Extra Polylogarithmic Factors: Let $G \subseteq H$ be any 2ε-packing of the hypothesis class H. Since each query can provide at most one bit of information, we require at least $\Omega(\log|G|)$ queries to solve the problem. While it is not universally true, in many practical and sufficiently complex cases, the size of the largest 2ε-packing scales polynomially with $\frac{1}{\varepsilon}$ and d. Additionally, the boundedness parameters R_1 and R_2 often scale linearly with d in these contexts. Therefore, in such natural and complex scenarios, $\log\left(\frac{R_1R_2}{\varepsilon}\right)$ serves as a meaningful lower bound on m^*. Consequently, the extra polylogarithmic factors in our label complexity are of lower order and negligible.

Polynomial Dependence on m^*: We believe that our current analysis may not be tight. We conjecture that the actual label complexity of our algorithm exhibits a quadratic or even linear dependence on m^*, analogous to the results in Price and Zhou (2023) for the deterministic binary classifiers.

Parameter Setting of Algorithm 4: Properly setting the parameters γ, K, and M in Algorithm 4 appears to require knowledge of m^*. To determine γ, we can utilize an upper bound of m^* from passive learning algorithms; for instance, the $\tilde{O}\left(\frac{R_1^2 d}{\varepsilon^4}\right)$ upper bound provided by Klivans and Meka (2017) allows us to maintain the same label complexity guarantee. However, setting K and M indeed necessitates knowledge of m^*. One potential solution is to directly use the computable information function r as a measure of the expected potential growth, which is an upper bound of $\frac{1}{\mathrm{poly}(m^*)}$.

Boosting the Success Probability: Algorithm 4 succeeds with probability 0.7. We can amplify its success probability to $1-\delta$ by running $O\left(\log\frac{1}{\delta}\right)$ independent copies of the algorithm and returning the center of the heaviest 34ε ball among these hypotheses, as established in the following corollary.

Corollary 6.2. *Let*

$$m = m^*\left(X, \mathcal{D}_X, H, \frac{\varepsilon^2}{16\sqrt{2}dR_1R_2}, 0.01\right).$$

Under Assumption 1.3, there exists an algorithm that returns a hypothesis $\hat{h}$ such that $\mathrm{err}\left(\hat{h}\right) \leqslant 68\varepsilon$ *with probability at least* $1-\delta$*, using a label complexity of*

$$O\left(\mathrm{poly}\left(m\right)\mathrm{polylog}\left(\frac{R_1R_2}{\varepsilon}\right)\log\frac{1}{\delta}\right).$$

7 RUNNING TIME

For each parameter $\theta_h \in \Theta_H$, the penalty function is

$$\ell_{\theta_h}(x,y) = y\log\frac{1}{\sigma(\theta_h^\top x)} + (1-y)\log\frac{1}{1-\sigma(\theta_h^\top x)},$$

where $\sigma(\cdot)$ is the sigmoid function. Since this function is convex, the distribution over Θ_H is log-concave, and prior work shows that sampling from log-concave distributions can be done in polynomial time (Lovász and Simonovits, 1993; Kannan et al., 1997; Lovász and Vempala, 2006; Vempala and Wibisono, 2019). Thus, sampling from λ or p takes polynomial time.

However, there is one tricky aspect of our algorithm's running time: it involves polynomially many calls to Algorithm 2, which does rejection sampling to find two hypotheses that are ε far from each other. We do not know how to show that this takes polynomial time in general. Although it can be inefficient if λ and p concentrate on the same region, we conjecture the double concentration is around the true hypothesis h^* with high probability. So if the rejection step takes too long, the current distribution is likely already concentrated around h^*, allowing us to sample directly.

8 EXTENSIONS

As long as the penalty function is convex, we can efficiently sample from the induced distribution. Importantly, our analysis does not rely on any properties specific to the sigmoid function beyond its Lipschitz continuity. We recall the following definition:

Definition 8.1 (Lipschitz Continuity)**.** *A function* $f: \mathbb{R} \to \mathbb{R}$ *is said to be* L-Lipschitz continuous *if there exists a constant* $L \geqslant 0$ *such that for all* $x, y \in \mathbb{R}$,

$$|f(x) - f(y)| \leqslant L|x-y|.$$

Thus, our algorithm and analysis extend naturally to other probabilistic binary functions by replacing the sigmoid with any function f that satisfies two conditions: (i) the corresponding penalty function

$$\ell_\theta(x,y) = y\log\frac{1}{f(\theta^\top x)} + (1-y)\log\frac{1}{1-f(\theta^\top x)}$$

must be convex, and (ii) the function f must be L-Lipschitz continuous. This generalization covers a broad range of generalized linear models, including those in the exponential family with Lipschitz continuous link functions. In our proofs, the only modification necessary is in Lemma C.6, where the Lipschitz property is used; the corresponding Lipschitz constant will then appear in the final label complexity bound.

9 EXPERIMENTS

We implemented Algorithm 1, clipping the logarithmic ratio at 100 when computing the KL divergence, and refer to this variant as OURS[†]. For sampling from the log-concave distribution of hypotheses, we utilize an implementation of the Metropolis-adjusted Langevin algorithm (MALA) from TensorFlow(Abadi et al., 2015). We compare OURS against three baselines

1. Passive Learning (PASS): Queries are selected according to a random permutation of the training set.

2. Leverage Score Sampling (LSS): Queries are sampled proportionally to the leverage scores of the training set, where for a dataset $X \in \mathbb{R}^{n\times d}$ the leverage score of the ith data point is defined as $\ell_i = \left[X(X^\top X)^{-1}X^\top\right]_{ii}$, which quantifies its importance in capturing the dataset's essential information (Chowdhury and Ramuhalli, 2024; Mahoney et al., 2011).

3. Active Classification using Experimental Design (ACED) (Katz-Samuels et al., 2021): An active learning algorithm that has demonstrated superior empirical performance among active learning algorithms with provable sample complexity.

[†] The code is available at the following GitHub repository: `https://github.com/trung6/ActLogReg`.

Dataset	Training	Number of Queries			
	Performance	OURS	ACED	PASS	LSS
Synthetic	82.5%	601	889	961	1024
Musk	92.5%	249	762	676	656

Table 1: Comparison of the number of queries needed for OURS, ACED, LSS, and PASS to achieve a specific performance.

Our evaluation follows a two-stage pipeline. First, we gather datasets by running the different algorithms, and then we train a logistic regression model on the queried datasets, assessing the model's performance on both the entire training set and a held-out test set. We perform experiments on two datasets: a synthetic dataset (syn_100) and the Musk dataset (Chapman and Jain, 1994) (musk_v2) described below.

1. Synthetic Dataset (syn_100): The synthetic dataset, referred to as syn_100, consists of points sampled uniformly from the hypercube $[-1, 1]^{100} \subseteq \mathbb{R}^{100}$. To enable the logistic regression model to include a bias term, we augment each data point with an additional dimension set to a constant value of 1. We generate a random vector $w^* \in [-1, 1]^{101}$ and assign labels to each data point x according to the probability $\Pr[x \text{ is labeled as } 1] = \frac{1}{1+\exp\left(-(w^*)^\top x\right)}$. Both the training and test sets contain 10000 data points.

2. Musk Dataset (musk_v2) (Chapman and Jain, 1994): This dataset comprises 102 molecules, with 39 identified as musks and 63 as non-musks by human experts. The objective is to predict whether new molecules are musks or non-musks using 166 features that describe the molecules' various conformations. We split the dataset into training and test sets containing 4420 and 2178 data points, respectively.

We measure performance in terms of accuracy on both datasets. Additionally, on the synthetic dataset—where the ground truth is known—we evaluate performance using the weighted ℓ_2-distances defined in Section 1.1. The experimental results are shown in Figure 1[*] and Figure 2[*]. Table 1 demonstrates that OURS achieves comparable training performance to other methods on some target value, while requiring significantly fewer queries on both datasets.

10 CONCLUSION

We presented the first active logistic regression algorithm with provable label complexity bounds that is polynomially competitive with the optimal on any problem instance, up to polylogarithmic factors. In particular, the promise of active learning is that it allows for algorithms like binary search that can, in some cases, improve the sample complexities from $\frac{1}{\varepsilon}$ to $\log \frac{1}{\varepsilon}$. Whenever such an improvement is possible, our algorithm will get a bound of $\mathrm{poly}(\log \frac{1}{\varepsilon})$.

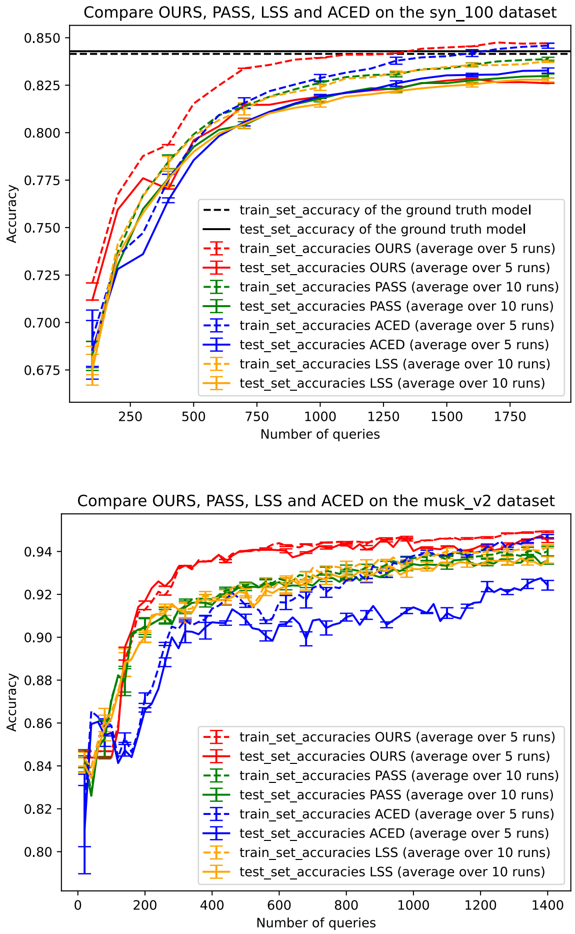

Figure 1: Comparison of OURS with PASS, LSS and ACED on (a) a 100-dimension synthetic dataset and (b) the Musk dataset

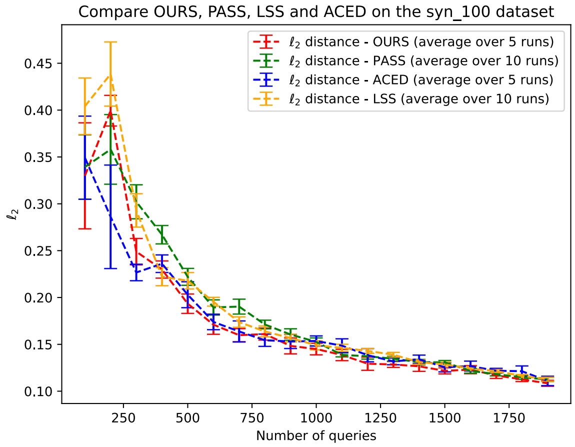

Figure 2: Comparison of OURS with PASS, LSS and ACED in terms of the weighted ℓ_2-distances between estimated hypotheses and the ground truth hypothesis on the synthetic dataset

[*] Error bars represent the standard errors.

ACKNOWLEDGMENTS

We would like to thank Jiawei Li and Zhiyang Xun for their valuable suggestions in polishing the paper. We also thank ChatGPT for its assistance in refining our manuscript. Support was provided by NSF award CCF-1751040 (CAREER) and the NSF AI Institute for Foundations of Machine Learning (IFML).

Reference

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 65–72, 2006.

D. Chapman and A. Jain. Musk (Version 2). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C51608.

X. Chen and E. Price. Active regression via linear-sample sparsification. In *Conference on Learning Theory*, pages 663–695. PMLR, 2019.

A. Chowdhury and P. Ramuhalli. A provably accurate randomized sampling algorithm for logistic regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):11597–11605, Mar. 2024. doi: 10.1609/aaai.v38i10.29042. URL https://ojs.aaai.org/index.php/AAAI/article/view/29042.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine learning*, 15:201–221, 1994.

S. Dasgupta. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 17, 2004.

S. Dasgupta, D. J. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. *Advances in neural information processing systems*, 20, 2007.

A. Gajjar, C. Musco, and C. Hegde. Active learning for single neuron models with lipschitz non-linearities. In *International Conference on Artificial Intelligence and Statistics*, pages 4101–4113. PMLR, 2023.

A. Gajjar, W. M. Tai, X. Xingyu, C. Hegde, C. Musco, and Y. Li. Agnostic active learning of single index models with linear sample complexity. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1715–1754. PMLR, 2024.

S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*, pages 353–360, 2007.

D. Hsu and A. Mazumdar. On the sample complexity of parameter estimation in logistic regression with normal design. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 2418–2437. PMLR, 2024.

S. M. Kakade, V. Kanade, O. Shamir, and A. Kalai. Efficient learning of generalized linear and single index models with isotonic regression. *Advances in Neural Information Processing Systems*, 24, 2011.

A. T. Kalai and R. Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT*, volume 1, page 9, 2009.

R. Kannan, L. Lovász, and M. Simonovits. Random walks and an $o^*(n^5)$ volume algorithm for convex bodies. *Random Structures & Algorithms*, 11(1):1–50, 1997.

J. Katz-Samuels, J. Zhang, L. Jain, and K. Jamieson. Improved algorithms for agnostic pool-based active classification. In *International Conference on Machine Learning*, pages 5334–5344. PMLR, 2021.

A. Klivans and R. Meka. Learning graphical models using multiplicative weights. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 343–354. IEEE, 2017.

T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

L. Lovász and M. Simonovits. Random walks in a convex body and an improved volume algorithm. *Random structures & algorithms*, 4(4):359–412, 1993.

L. Lovász and S. Vempala. Simulated annealing in convex bodies and an $o^*(n^4)$ volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006.

M. W. Mahoney et al. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

T. Mai, C. Musco, and A. Rao. Coresets for classification–simplified and strengthened. *Advances in Neural Information Processing Systems*, 34:11643–11654, 2021.

A. Munteanu, C. Schwiegelshohn, C. Sohler, and D. Woodruff. On coresets for logistic regression. *Advances in Neural Information Processing Systems*, 31, 2018.

C. Musco, C. Musco, D. P. Woodruff, and T. Yasuda. Active linear regression for ℓ_p norms and beyond. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 744–753. IEEE, 2022.

E. Price and Y. Zhou. A competitive algorithm for agnostic active learning. *Advances in Neural Information Processing Systems*, 36, 2023.

S. Sabato and R. Munos. Active regression by stratification. *Advances in Neural Information Processing Systems*, 27, 2014.

S. Vempala and A. Wibisono. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. *Advances in neural information processing systems*, 32, 2019.

Y. Yang and M. Loog. A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83:401–415, 2018.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A OMITTED PROOFS IN THE MAIN PAPER

A.1 Proof of Lemma 3.1

Proof. On a high level, for any algorithm, we aim to construct two hypotheses h_1 and h_2 such that they are far away enough so the algorithm has to distinguish them but at the same time they are hard to distinguish, which gives a lower bound on $m^*(H, \mathcal{D}_X, \varepsilon, 0.01)$. Let's first define some notation. Let A be any algorithm such that with m label queries, it returns a hypothesis that is ε-close to the ground truth with probability at least 0.9 for any choice of the ground truth. Specifically, if two hypotheses h_1 and h_2 satisfy $\|h_1 - h_2\| \geqslant 2\varepsilon$, then we can use A to distinguish h_1 and h_2 with probability at least 0.9. Let the random variable P_h^A be the transcript of algorithm A if h is the ground truth. Note that h could be improper, which means h may not belong to the hypothesis class H. The transcript P_h^A is a collection of queried point and label pairs in the form of $\{(x_1, y_1), (x_2, y_2), \cdots\}$. WLOG, we assume all transcripts have length m because if A terminates before making m queries, we can pad the transcript with arbitrary queries which A would just ignore. Let $\bar{h}$ be the average of hypotheses which is defined as $\bar{h}(x) := \mathbb{E}_{h' \sim \lambda}[h'(x)]$ for every x. Let d denote the marginal of the distribution of $P_{\bar{h}}^A$, in other words, each x is expected to be queried $m \cdot d(x)$ times. We define the query-induced KL distance between two hypotheses h_1 and h_2 with respect to query distribution d as

$$\operatorname{dist}_d(h_1, h_2) = \underset{x \sim d}{\mathbb{E}}\left[D_{\mathrm{KL}}(h_1(x), h_2(x))\right].$$

Constructing two hard-to-distinguish hypotheses. We aim to construct two hypotheses h_1 and h_2 such that they have the following properties

1. The distance between two hypotheses satisfies $\operatorname{dist}_d(h_1, h_2) \geqslant 2\varepsilon$, so $\mathcal{A}$ can distinguish them with probability at least 0.9.
2. The query-induced distance between h_1, h_2 and $\bar{h}$ satisfies $\operatorname{dist}_d\left(\bar{h}, h_1\right), \operatorname{dist}_d\left(\bar{h}, h_2\right) \leqslant 10\, \mathbb{E}_{h' \sim \lambda}\left[\operatorname{dist}_d\left(\bar{h}, h'\right)\right]$, so they are both close to $\bar{h}$ and hard to distinguish.

The mean of query-induced KL distance between $\bar{h}$ and h' sampled from λ is $\mathbb{E}_{h' \sim \lambda}\left[\operatorname{dist}_d\left(\bar{h}, h'\right)\right] = \mathbb{E}_{x \sim d}\, \mathbb{E}_{h' \sim \lambda}\left[D_{\mathrm{KL}}\left(\bar{h}(x), h'(x)\right)\right]$. By Markov's inequality, we know that

$$\Pr_{h' \sim \lambda}\left[\operatorname{dist}_d(\bar{h}, h') > 10 \underset{h' \sim \lambda}{\mathbb{E}}\left[\operatorname{dist}_d\left(\bar{h}, h'\right)\right]\right] \leqslant \frac{1}{10}. \tag{1}$$

We pick h_1 to be any of the hypotheses satisfies $\operatorname{dist}_d(\bar{h}, h') < 10\, \mathbb{E}_{h' \sim \lambda}\left[\operatorname{dist}_d\left(\bar{h}, h'\right)\right]$. Furthermore, by the anti-concentration assumption of this lemma, we know that at least 20% of the hypotheses are at least 2ε from h_1. Combining with (1), there exists a hypothesis h_2 such that $\|h_1 - h_2\| \geqslant 2\varepsilon$ and $\operatorname{dist}_d\left(\bar{h}, h_2\right) \leqslant 10\, \mathbb{E}_{h' \sim \lambda}\left[\operatorname{dist}_d\left(\bar{h}, h'\right)\right]$ as desired.

Implications. We know that A can distinguish h_1 and h_2 with more than 0.9 probability using m queries and by the definition of total variance distance,

$$0.1 \geqslant \Pr\left[A \text{ fails to distinguish } h_1 \text{ and } h_2\right] \geqslant \frac{1}{2}\left(1 - D_{\mathrm{TV}}\left(P_{h_1}^A, P_{h_2}^A\right)\right).$$

From triangle inequality and Pinsker's inequality, we have

$$\begin{aligned} 0.8 \geqslant D_{\mathrm{TV}}\left(P_{h_1}^A, P_{h_2}^A\right) &\leqslant D_{\mathrm{TV}}\left(P_{\bar{h}}^A, P_{h_1}^A\right) + D_{\mathrm{TV}}\left(P_{\bar{h}}^A, P_{h_2}^A\right) \\ &\leqslant \sqrt{\frac{1}{2} D_{\mathrm{KL}}\left(P_{\bar{h}}^A \| P_{h_1}^A\right)} + \sqrt{\frac{1}{2} D_{\mathrm{KL}}\left(P_{\bar{h}}^A \| P_{h_2}^A\right)}. \end{aligned}$$

Decomposing the KL divergence using Lemma C.1 (Lattimore and Szepesvári, 2020)[Lemma 15.1] and the rest follows as

$$\begin{aligned} \sqrt{\frac{1}{2} D_{\mathrm{KL}}\left(P_{\bar{h}}^A \| P_{h_1}^A\right)} + \sqrt{\frac{1}{2} D_{\mathrm{KL}}\left(P_{\bar{h}}^A \| P_{h_2}^A\right)} &= \sqrt{\frac{m}{2} \operatorname{dist}_d\left(\bar{h}, h_1\right)} + \sqrt{\frac{m}{2} \operatorname{dist}_d\left(\bar{h}, h_2\right)} \\ &\leqslant 5\sqrt{m \underset{h' \sim \lambda}{\mathbb{E}}\left[\operatorname{dist}_d\left(\bar{h}, h'\right)\right]} \\ &\leqslant 5\sqrt{m r(x^*)}, \end{aligned}$$

where $x^* = \arg\max_x r_\lambda(x)$. In the above, the first inequality comes from the definition of $\tilde{h}$ and $\hat{h}$. The last step is from the definition of x^*. As a result, we have

$$r(x^*)m \gtrsim 1.$$

Since this result holds for any algorithm, including the optimal one, rearrange and we get

$$m^*(H, \mathcal{D}_X, \varepsilon, 0.1) \gtrsim \frac{1}{r(x^*)}.$$

□

A.2 Proof of Lemma 3.2

Proof. The calculaton is the following.

$$\begin{aligned}
&\mathbb{E}_{y_i}\left[\psi_{i+1}(h^*) - \psi_i(h^*) \,|\, x_i\right] \\
=&\mathbb{E}_{y_i}\left[\log \frac{\lambda_{i+1}(h^*)}{\lambda_i(h^*)} \,\middle|\, x_i\right] \\
=&\mathbb{E}_{y_i}\left[\log\left(\frac{w_{i+1}(h^*)}{w_i(h^*)} \cdot \frac{w_i(H)}{w_{i+1}(H)}\right) \,\middle|\, x_i\right] \\
=&\mathbb{E}_{y_i}\left[\log\left(\frac{w_{i+1}(h^*)}{w_i(h^*)} \cdot \frac{1}{\displaystyle\int_{h\in H} \frac{w_{i+1}(h)}{w_i(h)} dh}\right) \,\middle|\, x_i\right] \\
=&\mathbb{E}_{y_i}\left[\log\left(\exp\left(-\ell_{h*}(x_i, y_i)\right) \cdot \frac{1}{\mathbb{E}_{h\sim\lambda_i}\left[\exp\left(-\ell_h(x_i, y_i)\right)\right]}\right) \,\middle|\, x_i\right] \\
=&\mathbb{E}_{y_i}\left[-\ell_{h*}(x_i, y_i) - \log \mathbb{E}_{h\sim\lambda_i}\left[\exp\left(-\ell_h(x_i, y_i)\right)\right] \,|\, x_i\right] \\
=&h^*(x_i)\log\frac{h^*(x_i)}{\bar{h}_{\lambda_i}(x_i)} + (1 - h^*(x_i))\log\frac{1 - h^*(x_i)}{1 - \bar{h}_{\lambda_i}(x_i)} \\
=&D_{\mathrm{KL}}\left(h^*(x_i) \,\|\, \bar{h}_{\lambda_i}(x_i)\right).
\end{aligned}$$

□

A.3 Proof of Lemma 4.1

Proof. We've already calculated the expected potential growth under a single query in Lemma 3.2. To go from single query to double query, we just need to calculate how the mean $\bar{h}_i(x_i)$ changes after the first query. For the sake of bookkeeping, in the following we drop the queried point x_i because it is unambiguous in the context. Let's use w_i' to denote the weight function after the first query and λ_i' to denote the distribution induced by the weight function. We also use $\bar{h}_i'$ to denote the mean under the distribution λ_i'.

Case 1: The first label $y_i^1 = 1$. Then each hypothesis h gets penalty $\ell_h(x_i, 1) = -\log h(x_i)$ so $w_i'(h) = hw_i(h)$. Therefore,

$$\begin{aligned}
\bar{h}_i' &= \mathop{\mathbb{E}}_{h\sim\lambda_i'}[h] = \int_{h\in H} \lambda_i'(h) h\, dh = \int_{h\in H} \frac{w_i'(h)}{w_i'(H)} h\, dh = \int_{h\in H} \frac{h\, w_i(h)}{\displaystyle\int_{h\in H} h\, w_i(h)\, dh} h\, dh \\
&= \int_{h\in H} \frac{h\,\lambda_i(h)}{\displaystyle\int_{h\in H} h\,\lambda_i(h)\, dh} h\, dh = \int_{h\in H} \frac{h^2\,\lambda_i(h)}{\bar{h}_i} dh = \frac{1}{\bar{h}_i} \mathop{\mathbb{E}}_{h\sim\lambda_i}\left[h^2\right] \\
&= \frac{1}{\bar{h}_i}\left(\mathop{\mathrm{Var}}_{h\sim\lambda_i}[h] + \bar{h}_i^2\right) \\
&= \frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{\bar{h}_i} + \bar{h}_i.
\end{aligned}$$

Case 2: The first label $y_i^1 = 0$. Then each hypothesis h gets penalty $\ell_h(x_i, 1) = -\log(1 - h(x_i))$ so $w_i'(h) = (1-h)w_i(h)$. Therefore,

$$\begin{aligned}
\bar{h}_i' &= \mathop{\mathbb{E}}_{h\sim\lambda_i'}[h] = \int_{h\in H} \lambda_i'(h)h\,dh \\
&= \int_{h\in H} \frac{w_i'(h)}{w_i'(H)} h\,dh = \int_{h\in H} \frac{(1-h)w_i(h)}{\displaystyle\int_{h\in H}(1-h)w_i(h)\,dh} h\,dh = \int_{h\in H} \frac{(1-h)\lambda_i(h)}{\displaystyle\int_{h\in H}(1-h)\lambda_i(h)\,dh} h\,dh \\
&= \int_{h\in H} \frac{(1-h)h\lambda_i(h)}{1-\bar{h}_i}\,dh = \frac{1}{1-\bar{h}_i}\left(\bar{h}_i - \mathop{\mathbb{E}}_{h\sim\lambda_i}\left[h^2\right]\right) = \frac{1}{1-\bar{h}_i}\left(\bar{h}_i - \mathop{\mathrm{Var}}_{h\sim\lambda_i}[h] - \bar{h}_i^2\right) \\
&= -\frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{1-\bar{h}_i} + \bar{h}_i.
\end{aligned}$$

Case 1 happens with probability h^* and case two happens with probability $1-h^*$. Them combined with Lemma 3.2, we proved

$$\begin{aligned}
&\mathop{\mathbb{E}}_{y_i^1, y_i^2}\left[\psi_{i+1}(h^*) - \psi_i(h^*)|x_i\right] \\
=&D_{\mathrm{KL}}\left(h^*(x_i), \bar{h}_i(x_i)\right) + h^*(x_i)D_{\mathrm{KL}}\left(h^*(x_i), \bar{h}_i(x_i) + \frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{\bar{h}_i(x_i)}\right) \\
&+ (1-h^*(x_i))\, D_{\mathrm{KL}}\left(h^*(x_i), \bar{h}_i(x_i) - \frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{1-\bar{h}_i(x_i)}\right)
\end{aligned}$$

Then from Pinsker's inequality, we have

$$\begin{aligned}
&D_{\mathrm{KL}}\left(h^*, \bar{h}_i\right) + h^* D_{\mathrm{KL}}\left(h^*, \bar{h}_i + \frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{\bar{h}_i}\right) + (1-h^*)\, D_{\mathrm{KL}}\left(h^*, \bar{h}_i - \frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{1-\bar{h}_i}\right) \\
\gtrsim& D_{\mathrm{TV}}^2\left(h^*, \bar{h}_i\right) + h^* D_{\mathrm{TV}}^2\left(h^*, \bar{h}_i + \frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{\bar{h}_i}\right) + (1-h^*)\, D_{\mathrm{TV}}^2\left(h^*, \bar{h}_i - \frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{1-\bar{h}_i}\right) \\
\gtrsim& \left(h^* - \bar{h}_i\right)^2 + h^*\left(\frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{\bar{h}_i} - \left(h^* - \bar{h}_i\right)\right)^2 + (1-h^*)\left(\frac{\mathrm{Var}_{h\sim\lambda_i}[h]}{1-\bar{h}_i} - \left(\bar{h}_i - h^*\right)\right)^2 \\
\gtrsim& \left(h^* - \bar{h}_i\right)^2 + \left(\mathop{\mathrm{Var}}_{h\sim\lambda_i}[h]\right)^2.
\end{aligned}$$

The last step comes from the fact that $\max\{h^*, 1-h^*\} \geqslant \frac{1}{2}$ and $a^2 + b^2 \gtrsim (a+b)^2$. □

A.4 Proof of Lemma 5.2

Proof. Notice that for a fixed sequence of queried points, the order does not affect the expected potential change because the randomness only comes from the labels. Therefore, we could move the point $x_{(i,j)}$ to be the first query, i.e, we have

$$\Pr_{h\sim\lambda_0}\left[D_{\mathrm{KL}}\left(h^*\left(x_{(i,1)}\right)\middle\| h\left(x_{(i,1)}\right)\right) \geqslant \frac{1}{(m^*)^4 \log^5 \frac{1}{\gamma}}\middle|\mathcal{F}_{(i,j)}\right] \geqslant \frac{1}{(m^*)^4\log^4\frac{1}{\gamma}}.$$

In the algorithm, we set $\lambda_0 = \lambda_{(i,1)}$. So from Lemma C.3 and Lemma 4.1, we have that the expected potential growth of querying $x_{(i,1)}$ is

$$\mathbb{E}\left[\psi_{i+1}(h^*) - \psi_i(h^*)|x_{(i,1)}\right] \geqslant \frac{1}{(m^*)^{12}\log^{16}\frac{1}{\gamma}}.$$

Note that by definition $\mathcal{F}_{(i,j)}$ contains information of the queried point $x_{(i,1)}$. Moreover, from Lemma 3.2, we know that the expected potential growth of querying any x is non-negative, so our proof finishes. □

A.5 Proof of Lemma 5.3

Proof. For simplicity let's omit the phase index i. Let's first bound the expected potential change of $\log p_j^{H\setminus B_{h'}(2\varepsilon)}(h^*)$ for any fixed h' after one single query,

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{y_j}\left[\log \frac{p_{j+1}^{H\setminus B_{h'}(2\varepsilon)}(h^*)}{p_j^{H\setminus B_{h'}(2\varepsilon)}(h^*)}\middle| x_j\right]\\
=&\mathop{\mathbb{E}}_{y_j}\left[\log\left(\frac{w_{j+1}(h^*)}{w_j(h^*)}\frac{w_j(H\setminus B_{h'}(2\varepsilon))}{w_{j+1}(H\setminus B_{h'}(2\varepsilon))}\right)\middle| x_j\right]\\
=&\mathop{\mathbb{E}}_{y_j}\left[\log\left(\frac{w_{j+1}(h^*)}{w_j(h^*)}\frac{1}{\int_{h\in H\setminus B_{h'}(2\varepsilon)}\frac{w_{j+1}(h)}{w_j(H\setminus B_{h'}(2\varepsilon))}dh}\right)\middle| x_j\right]\\
=&\mathop{\mathbb{E}}_{y_j}\left[\log\left(\exp\left(-\ell_{h^*}(x_j,y_j)\right)\frac{1}{\mathbb{E}_{h\sim p_j^{H\setminus B_{h'}(2\varepsilon)}}\left[\exp\left(-\ell_h(x_j,y_j)\right)\right]}\right)\middle| x_j\right]\\
=&h^*(x_j)\log\frac{h^*(x_j)}{\mathbb{E}_{h\sim p_j^{H\setminus B_{h'}(2\varepsilon)}}\left[h(x_j)\right]}+(1-h^*(x))\log\frac{1-h^*(x_j)}{1-\mathbb{E}_{h\sim p_j^{H\setminus B_{h'}(2\varepsilon)}}\left[h(x_j)\right]}\\
=&D_{\mathrm{KL}}\left(h^*(x_j),\bar{h}_{p_j^{H\setminus B_{h'}(2\varepsilon)}}(x_j)\right).
\end{aligned}
\tag{2}
$$

Then following the same steps as in the proof of Lemma 4.1 and then taking expectation over λ^0, the proof is finished. □

A.6 Proof of Lemma 5.4

Proof. For bookkeeping, we simplify the notations by omitting the queried point $x_{(i,j)}$, the conditions and the phase index i. Up to (5), all results are conditioned on $\mathcal{F}_{(i,j)}$ and the event $\bar{A}_{(i,j)}$. We begin by establishing two important facts.

Fact 1:

$$
\left(\mathop{\mathrm{Var}}_{h\sim\hat{\lambda}_j}[h]\right)^2 \gtrsim \frac{1}{(m^*)^4\log^4\frac{1}{\gamma}}.
$$

Proof of Fact 1. By Lemmas C.4 and 3.1, we can relate $r_{\hat{\lambda}_j}$ to m^* as $r_{\hat{\lambda}_j} \gtrsim \frac{1}{m^*}$. Using Lemma C.5, we have:

$$
\frac{1}{(m^*)^4} \lesssim r_{\hat{\lambda}_j}^4 \lesssim \left(\mathop{\mathrm{Var}}_{h\sim\hat{\lambda}_j}[h]\right)^2\log^4\frac{1}{\gamma}.
$$

Rearranging the inequality yields Fact 1.

Fact 2:

$$
\mathop{\mathbb{E}}_{h\sim\lambda_0}\left[(h-h^*)^2\right] \lesssim \frac{1}{(m^*)^4\log^3\frac{1}{\gamma}}.
$$

Proof of Fact 2. Using Pinsker's inequality $\left((h-h^*)^2 \lesssim D_{\mathrm{KL}}(h^*\|h)\right)$ and Lemma C.2 $\left(D_{\mathrm{KL}}(h^*\|h) \lesssim \log\frac{1}{\gamma}\right)$, we have:

$$
\begin{aligned}
\mathop{\mathbb{E}}_{h\sim\lambda_0}\left[(h-h^*)^2\right] &\lesssim \mathop{\mathbb{E}}_{h\sim\lambda_0}\left[D_{\mathrm{KL}}(h^*,h)\right]\\
&\leqslant \frac{1}{(m^*)^4\log^5\frac{1}{\gamma}} + \Pr_{h\sim\lambda_0}\left[D_{\mathrm{KL}}(h^*,h) \geqslant \frac{1}{(m^*)^4\log^5\frac{1}{\gamma}}\right]\log\frac{1}{\gamma}\\
&\lesssim \frac{1}{(m^*)^4\log^3\frac{1}{\gamma}},
\end{aligned}
$$

where the last step uses the assumption of this lemma.

Lower bound of $\mathbb{E}_{h\sim\lambda_j^1}\left[(h-h^*)^2\right]$. By definition, $\hat{\lambda}_j = \frac{1}{2}\lambda_0 + \frac{1}{2}\lambda_j^1$, so:

$$\bar{h}_{\hat{\lambda}_j} = \frac{1}{2}\bar{h}_{\lambda_0} + \frac{1}{2}\bar{h}_{\lambda_j^1}.$$

Since $\bar{h}_{\hat{\lambda}_j}$ is a convex combination of $\bar{h}_{\lambda_0}$ and $\bar{h}_{\lambda_j^1}$, we have:

$$\left(\bar{h}_{\hat{\lambda}_j} - h^*\right)^2 \leqslant \left(\bar{h}_{\lambda_0} - h^*\right)^2 + \left(\bar{h}_{\lambda_j^1} - h^*\right)^2. \tag{3}$$

Using the inequality $a^2 + b^2 \gtrsim (a+b)^2$ and Fact 1, we have:

$$\frac{1}{(m^*)^2 \log^2 \frac{1}{\gamma}} \lesssim \operatorname*{Var}_{h\sim\hat{\lambda}_j}[h] = \mathop{\mathbb{E}}_{h\sim\hat{\lambda}_j}\left[\left(h - \bar{h}_{\hat{\lambda}_j}\right)^2\right] \lesssim \mathop{\mathbb{E}}_{h\sim\hat{\lambda}_j}\left[(h-h^*)^2\right] + \left(\bar{h}_{\hat{\lambda}_j} - h^*\right)^2.$$

From definition of $\hat{\lambda}_j$,

$$\mathop{\mathbb{E}}_{h\sim\hat{\lambda}_j}\left[(h-h^*)^2\right] = \frac{1}{2}\mathop{\mathbb{E}}_{h\sim\lambda_0}\left[(h-h^*)^2\right] + \frac{1}{2}\mathop{\mathbb{E}}_{h\sim\lambda_j^1}\left[(h-h^*)^2\right]. \tag{4}$$

Combining (3), (4) and Jensen's inequality,

$$\begin{aligned}\frac{1}{(m^*)^2 \log^2 \frac{1}{\gamma}} &\lesssim \frac{1}{2}\mathop{\mathbb{E}}_{h\sim\lambda_0}\left[(h-h^*)^2\right] + \frac{1}{2}\mathop{\mathbb{E}}_{h\sim\lambda_j^1}\left[(h-h^*)^2\right] + \left(\bar{h}_{\lambda_0} - h^*\right)^2 + \left(\bar{h}_{\lambda_j^1} - h^*\right)^2 \\ &\leqslant \frac{1}{2}\mathop{\mathbb{E}}_{h\sim\lambda_0}\left[(h-h^*)^2\right] + \frac{1}{2}\mathop{\mathbb{E}}_{h\sim\lambda_j^1}\left[(h-h^*)^2\right].\end{aligned}$$

Using Fact 2, we conclude that

$$\mathop{\mathbb{E}}_{h\sim\lambda_j^1}\left[(h-h^*)^2\right] \gtrsim \frac{1}{(m^*)^2 \log^2 \frac{1}{\gamma}}.$$

Conclusion. By definition:

$$\mathop{\mathbb{E}}_{h\sim\lambda_j^1}\left[(h-h^*)^2\right] = \mathop{\mathbb{E}}_{h'\sim\lambda_0}\left[\mathop{\mathbb{E}}_{h\sim p_i^{H\setminus B_{h'}(2\varepsilon)}}\left[(h^*-h)^2\right]\right] \gtrsim \frac{1}{(m^*)^2 \log^2 \frac{1}{\gamma}}. \tag{5}$$

From Lemma 5.3, we have:

$$\begin{aligned}&\mathop{\mathbb{E}}_{y_i}\left[\tilde{\psi}_{(i,j+1)}(h^*) - \tilde{\psi}_{(i,j)}(h^*)\middle|\mathcal{F}_{(i,j)}, \bar{A}_{(i,j)}\right] \\ &\gtrsim \mathop{\mathbb{E}}_{h'\sim\lambda_0}\left[\left(h^* - \bar{h}_{p_i^{H\setminus B_{h'}(2\varepsilon)}}\right)^2 + \left(\operatorname*{Var}_{h\sim p_i^{H\setminus B_{h'}(2\varepsilon)}}[h]\right)^2\middle|\mathcal{F}_{(i,j)}, \bar{A}_{(i,j)}\right] \\ &\gtrsim \mathop{\mathbb{E}}_{h'\sim\lambda_0}\left[\left(\left(h^* - \bar{h}_{p_i^{H\setminus B_{h'}(2\varepsilon)}}\right)^2 + \mathop{\mathbb{E}}_{h\sim p_i^{H\setminus B_{h'}(2\varepsilon)}}\left[\left(h - \bar{h}_{p_i^{H\setminus B_{h'}(2\varepsilon)}}\right)^2\right]\right)^2\middle|\mathcal{F}_{(i,j)}, \bar{A}_{(i,j)}\right] \\ &\gtrsim \mathop{\mathbb{E}}_{h'\sim\lambda_0}\left[\left(\mathop{\mathbb{E}}_{h\sim p_i^{H\setminus B_{h'}(2\varepsilon)}}\left[(h-h^*)^2\right]\right)^2\middle|\mathcal{F}_{(i,j)}, \bar{A}_{(i,j)}\right] \\ &\gtrsim \left(\mathop{\mathbb{E}}_{h'\sim\lambda_0}\left[\mathop{\mathbb{E}}_{h\sim p_i^{H\setminus B_{h'}(2\varepsilon)}}\left[(h-h^*)^2\right]\middle|\mathcal{F}_{(i,j)}, \bar{A}_{(i,j)}\right]\right)^2 \\ &\gtrsim \frac{1}{(m^*)^4 \log^4 \frac{1}{\gamma}},\end{aligned}$$

where the third step uses the inequality $a^2 + b^2 \gtrsim (a+b)^2$ and the fourth step uses Jensen's inequality. □

A.7 Proof of Lemma 5.5

Proof. From Lemma 5.2 , we know that

$$\mathbb{E}\left[\psi_{i+1}(h^*) - \psi_i(h^*)|\mathcal{F}_{(i,j)}, A_{(i,j)}\right] \gtrsim \frac{1}{(m^*)^{12}\log^{16}\frac{1}{\gamma}}.$$

On the other hand, from Lemma 5.4, we know that

$$\mathbb{E}\left[\tilde{\psi}_{(i,j+1)}(h^*) - \tilde{\psi}_{(i,j)}(h^*)\middle|\mathcal{F}_{(i,j)}, \bar{A}_{(i,j)}\right] \gtrsim \frac{1}{(m^*)^{4}\log^{4}\frac{1}{\gamma}}.$$

Let $Q_j = \mathbb{1}_{\mathbb{E}\left[\psi_{i+1}(h^*)-\psi_i(h^*)|\mathcal{F}_{(i,j)}\right]\gtrsim\frac{1}{(m^*)^{12}\log^{16}\frac{1}{\gamma}}}$ where $\mathbb{1}_A$ is the indicator of event A and $\tilde{\Delta}_j = \tilde{\psi}_{(i,j+1)}(h^*) - \tilde{\psi}_{(i,j)}(h^*)$, then

$$\mathbb{E}\left[Q_j + \tilde{\Delta}_j\middle|\mathcal{F}_{(i,j)}\right] \gtrsim \frac{1}{(m^*)^{4}\log^{4}\frac{1}{\gamma}}. \tag{6}$$

Let $X_j = \sum_{l=1}^{j-1} Q_l + \tilde{\psi}_{(i,j)}(h^*)$, $\mu_j = \sum_{l=1}^{j-1}\mathbb{E}[X_{l+1} - X_l|\mathcal{F}_{(i,l)}]$ and $Y_j = X_j - \mu_j$. Then $\{Y_i\}_{i\geqslant 1}$ is a martingale because

$$\mathbb{E}\left[Y_{j+1} - Y_j|\mathcal{F}_{(i,j)}\right] = \mathbb{E}[X_{j+1} - X_j|\mathcal{F}_{(i,j)}] - \mathbb{E}[X_{j+1} - X_j|\mathcal{F}_{(i,j)}] = 0.$$

Moreover, this martingale has the property that

$$Y_1 = \tilde{\psi}_{(i,1)}(h^*) \geqslant \log\alpha$$

by definition of $\tilde{\psi}$ and the assumption $p_{(i,1)}(h^*) = \alpha$. From (6), we have for any $j \in [M]$,

$$\mu_j \geqslant \frac{j-1}{(m^*)^8\log^8\frac{1}{\gamma}}.$$

We can also bound the absolute increment of $|Y_{j+1} - Y_j|$ by

$$|Y_{j+1} - Y_j| \leqslant 2\left|Q_j + \tilde{\Delta}_j(h^*)\right| \leqslant 2\cdot\left(1 + 2\log\frac{1}{\gamma}\right) \leqslant 6\log\frac{1}{\gamma},$$

by the clipping assumption and equation (2). Then by applying Azuma-Hoeffding, we have

$$\begin{aligned}
&\Pr\left[Y_{M+1} - Y_1 \leqslant -\frac{1}{2}\mu_{M+1}\right] \\
&= \Pr\left[\sum_{j=1}^{M} Q_j + \tilde{\psi}_{(i,M+1)}(h^*) - \tilde{\psi}_{(i,1)}(h^*) \leqslant \mu_{M+1} - \frac{1}{2}\mu_{M+1}\right] \\
&\leqslant \exp\left(\frac{-\frac{\mu_{M+1}^2}{4}}{12M\log^2\frac{1}{\gamma}}\right) \\
&\leqslant \exp\left(-\frac{M}{48\,(m^*)^8\log^{10}\frac{1}{\gamma}}\right).
\end{aligned}$$

By picking $M = O\left(\left(\beta + \log\frac{1}{\alpha}\right)(m^*)^8\log^{10}\frac{1}{\gamma}\right)$ with proper constant, we showed that

$$\Pr\left[\sum_{j=1}^{M} Q_j + \tilde{\psi}_{(i,M+1)}(h^*) - \tilde{\psi}_{(i,1)}(h^*) \geqslant \beta\right] \geqslant 0.99.$$

Therefore, either we have

$$\Pr\left[\tilde{\psi}_{(i,M+1)}(h^*) - \tilde{\psi}_{(i,1)}(h^*) \geqslant \frac{\beta}{2}\right] \geqslant 0.9,$$

or we have

$$\Pr\left[\sum_{j=1}^{M} Q_j \geqslant \frac{\beta}{2}\right] \geqslant 0.09. \tag{7}$$

Since Q_j's are indicators, $\sum_{j=1}^{M} Q_j \geqslant \frac{\beta}{2}$ means there exists some j such that

$$\mathbb{E}\left[\psi_{i+1}(h^*) - \psi_i(h^*) | \mathcal{F}_{(i,j)}\right] \gtrsim \frac{1}{(m^*)^{12} \log^{16} \frac{1}{\gamma}}.$$

Because the expected potential gain is non-negative for any queried points, taking expectation over the σ-algebra and we get (7) implies

$$\mathbb{E}\left[\psi_{i+1}(h^*) - \psi_i(h^*)\right] \gtrsim \frac{1}{(m^*)^{12} \log^{16} \frac{1}{\gamma}}.$$

□

A.8 Proof of Lemma 5.6

Proof. Consider any ball B' with radius 2ε whose center is at least 3ε away from h^*, then B' does not intersect B, implying that $\tilde{\lambda}_{(j,i)}^{H\backslash B'}(B) \leqslant 1$ so $\log \tilde{\lambda}_{(j,i)}^{H\backslash B'}(B) \leqslant 0$. Equivalently, if $\log \tilde{\lambda}_{(j,i)}^{H\backslash C'}(B) > 0$ and $\tilde{\lambda}_{(j,i)}^{H\backslash C'}(B) > 1$ for some radius 2ε ball C', then the center of C' must be at most 3ε from h^*. Assume, for the sake of contradiction, that any radius 4ε ball containing h^* has a probability mass less than 0.9. Then for any radius 2ε ball C' whose center is at most 3ε from h^*, $\tilde{\lambda}_{(j,i)}^{H\backslash C'}(B) = \frac{\lambda_{(j,i)}(B)}{\lambda_{(j,i)}(H\backslash C')} \leqslant \frac{\lambda_{(j,i)}(C'')}{\lambda_{(j,i)}(H\backslash C'')} \leqslant \frac{0.9}{\lambda_{(j,i)}(H\backslash C'')}$. Here, C'' and C' share the same center, but C'' has a radius of 4ε so C'' contains B by definition. Moreover, $\lambda_{(j,i)}(H\backslash C'') \geqslant 1 - 0.9 = 0.1$ by our assumption so $\log \tilde{\lambda}_{(j,i)}^{H\backslash C'}(B) \leqslant \log 9$. This means that if the the assumption is true,

$$\tilde{\psi}_{(j,i)}(B) = \mathop{\mathbb{E}}_{h'\sim\lambda_0}\left[\log \lambda_{(j,i)}^{H\backslash B_{h'}(2\varepsilon)}(B)\right] \leqslant \Pr_{h'\sim\lambda_0}\left[\left\|h' - h^*\right\|_2 \leqslant 3\varepsilon\right] \cdot \log 9 < 10.$$

This is a contradiction so there exists a ball C with radius 4ϵ containing h^* and $\lambda_{(j,i)}(C) \geqslant 0.9$. □

A.9 Proof of Lemma 5.7

Proof. Let ξ be as defined in Lemma C.6 and we set the parameters as the following:

- Total number of phases $K = \Theta\left(d\,(m^*)^{12} \log^{16} \frac{1}{\gamma}\left(\log \frac{1}{\xi\gamma} + \log \frac{1}{\alpha}\right)\right)$.
- The parameter $\beta = 2d \log\left(\frac{1}{\xi\gamma}\right)$ in Lemma 5.5.

Then the total number of queries are

$$T = O\left(d^2\,(m^*)^{20} \log^{26} \frac{1}{\gamma} \log^2 \frac{1}{\xi\alpha}\right). \tag{8}$$

Lower Bounding Success Probability for Algorithm 3 Let E_i be the event that $\mathbb{E}\left[\psi_{i+1}(h^*) - \psi_i(h^*)\right] \gtrsim \frac{1}{(m^*)^{12}\log^{16}\frac{1}{\gamma}}$. Let $Z_i = \mathbb{1}_{E_i}$ be the indicator of E_i. First note that $\psi_{K+1}(h^*) \lesssim 2d\log\left(\frac{1}{\xi\gamma}\right)$. Otherwise, applying property 2 of Lemma C.6,

$$\begin{aligned}
\log \lambda_{K+1}(B) &\gtrsim \log \lambda_{K+1}(h^*) - \frac{T\xi R_2}{\log^{50} \frac{1}{\gamma} \log^2 \frac{1}{\xi}} - d \log\left(\frac{1}{\xi\gamma}\right) \\
&\gtrsim 2d \log\left(\frac{1}{\xi\gamma}\right) - o(1) - d\log\left(\frac{1}{\xi\gamma}\right) \\
&\gtrsim 2d \log\left(\frac{1}{\xi\gamma}\right) \\
&\gtrsim 0.
\end{aligned}$$

However, this is impossible because $\lambda_{K+1}(B) \leqslant 1$ by definition. By definition of Z_i, we have

$$Z_i \lesssim (m^*)^{12} \log^{16}\left(\frac{1}{\gamma}\right) \mathbb{E}\left[\psi_{i+1}(h^*) - \psi_i(h^*)\right].$$

Therefore,

$$\begin{aligned}
&\sum_{i=1}^{K} Z_i \\
\lesssim &(m^*)^{12} \log^{16}\left(\frac{1}{\gamma}\right) \sum_{i=1}^{K} \mathbb{E}\left[\psi_{i+1}(h^*) - \psi_i(h^*)\right] \\
\lesssim &(m^*)^{12} \log^{16}\left(\frac{1}{\gamma}\right) \mathbb{E}\left[\psi_{K+1}(h^*) - \psi_1(h^*)\right] \\
\lesssim &d\,(m^*)^{12} \log^{16}\frac{1}{\gamma}\left(\log\frac{1}{\xi\gamma} + \log\frac{1}{\alpha}\right).
\end{aligned}$$

By picking the proper constants, we can show

$$\sum_{i=1}^{K} Z_i \leqslant \frac{K}{10}.$$

Since we pick $\beta = 2d\log\left(\frac{1}{\xi\gamma}\right)$, from Lemma 5.5, for more than $\frac{9}{10}$ of all phases i,

$$\Pr\left[\tilde{\psi}_{(i,M+1)}(h^*) \geqslant 2d\log\left(\frac{1}{\xi\gamma}\right)\right] \geqslant 0.99. \tag{9}$$

Again by applying property 3 of Lemma C.6, we have if $\tilde{\psi}_{(i,M+1)}(h^*) \geqslant 2d\log\left(\frac{1}{\xi\gamma}\right)$, then

$$\begin{aligned}
\tilde{\psi}_{(i,M+1)}(B) &\gtrsim \tilde{\psi}_{(i,i)}\left(h^*\right) - \frac{T\xi R_2}{\log^{50}\frac{1}{\gamma}\log^2\frac{1}{\xi}} - d\log\left(\frac{1}{\xi\gamma}\right) \\
&\gtrsim d\log\left(\frac{1}{\xi\gamma}\right) \\
&> 10.
\end{aligned}$$

Because of property 1 of Lemma C.6, we can apply Lemma 5.6 so the above implies

$$p_{(i,M+1)}\left(B_{h*}(8\varepsilon)\right) \geqslant 0.9.$$

Therefore, (9) implies that for more than $\frac{9}{10}$ of all phases,

$$\Pr\left[p_{(i,M+1)}\left(B_{h*}(8\varepsilon)\right) \geqslant 0.9\right] \geqslant 0.99.$$

Taking expectation and we get for more than $\frac{9}{10}$ of all phases,

$$\mathbb{E}\left[p_{(i,M+1)}\left(B_{h*}(8\varepsilon)\right)\right] \geqslant 0.9 \cdot 0.99.$$

Since $\bar{\lambda} = \frac{1}{K}\sum_{i=1}^{K} p_{(i,M+1)}$,

$$\mathbb{E}\left[\bar{\lambda}\left(B_{h*}(8\varepsilon)\right)\right] = \frac{1}{K}\sum_{i=1}^{K} \mathbb{E}\left[p_{(i,M+1)}\left(B_{h*}(8\varepsilon)\right)\right] \geqslant 0.9 \cdot 0.99 \cdot 0.9 \geqslant 0.8.$$

On the other hand,

$$\begin{aligned}
\Pr_{\hat{h}\sim\bar{\lambda}}\left[\operatorname{err}\left(\hat{h}\right) \leqslant 8\varepsilon\right] &= \mathbb{E}\left[\Pr_{\hat{h}\sim\bar{\lambda}}\left[\hat{h} \in B_{h*}(8\varepsilon)\middle|\bar{\lambda}\right]\right] \\
&= \mathbb{E}\left[\bar{\lambda}\left(B_{h*}(8\varepsilon)\right)\right] \\
&\geqslant 0.8.
\end{aligned}$$

Conclusion The volume of the parameterized space is $O\left(R_1^d\right)$ so $\alpha = \Omega\left(R_1^{-d}\right)$ and $\log\frac{1}{\alpha} = O\left(d\log R_1\right)$. Plugging this in (8) and we get the total number of queries

$$T = O\left(d^2\left(m^*\right)^{20}\log^{26}\frac{1}{\gamma}\log^2\left(\frac{dR_1R_2m^*}{\varepsilon}\right)\right).$$

□

A.10 Proof of Lemma 6.1

Proof. WLOG, we assume $m \geqslant 1$. Let $\gamma = \frac{\varepsilon}{100m} \leqslant \frac{1}{100}$, where m is the sample complexity of A. For any $x \in X$ and $h^* \in H$, the probability that the clipped and unclipped versions of h^* give different labels is at most $\gamma \leqslant \frac{1}{100}$. The probability that they give different labels across all m queries is then less than 0.1, since

$$(1-\gamma)^m \geqslant \frac{19}{20}e^{-\gamma} \geqslant 0.9.$$

This holds because $1 - x \geqslant \frac{19}{20}e^{-x}$ for $x \in \left[0, \frac{1}{100}\right]$. If the labels match for all queries, the algorithm A cannot distinguish between the clipped and unclipped versions of h^*. Thus, by the union bound and the definition of A, with probability at least 0.8, A returns a hypothesis $\hat{h}$ within ε of the clipped h^*. Since clipping can reduce the error by at most γ, the true hypothesis $\hat{h}$ will be within $\varepsilon + \gamma \leqslant 2\varepsilon$. Substituting $\gamma = \frac{\varepsilon}{100m}$, the sample complexity of A becomes

$$O\left(m\operatorname{polylog}(m)\operatorname{polylog}\left(\frac{1}{\varepsilon}\right)\right).$$

□

A.11 Proof of Theorem 1.5

Proof. We bound the error, sample complexity and success probability of Algorithm 4 as below.

Bounding the Error. Let θ_S^* be the projection of θ^* onto S. From Lemma B.2, we have

$$\left\|h_{\theta*} - h_{\theta_S^*}\right\|_2^{\mathcal{D}_X} \leqslant \varepsilon. \tag{10}$$

Let h' be the clipped version of $h_{\theta_S^*}$. Then, by Lemma 5.7,

$$\left\|\hat{h} - h'\right\|_2^{\mathcal{D}_{X_S}} \leqslant 8\varepsilon.$$

Applying Lemma 6.1, we obtain

$$\left\|\hat{h} - h_{\theta_S^*}\right\|_2^{\mathcal{D}_X} = \left\|\hat{h} - h_{\theta_S^*}\right\|_2^{\mathcal{D}_{X_S}} \leqslant 16\varepsilon.$$

Using the triangle inequality with (10), we get

$$\left\|\hat{h} - h_{\theta*}\right\|_2^{\mathcal{D}_X} \leqslant 17\varepsilon.$$

Bounding the Sample Complexity. Clipping and dimension reduction simplify the problem, as they reduce the distances between hypotheses. Thus, a solution on the original or unclipped instance is also valid for the dimension-reduced or clipped instance. Furthermore, smaller error tolerance and lower failure probability make the problem harder. Let $m = m^*\left(X, \mathcal{D}_X, H, \frac{\varepsilon^2}{16\sqrt{2}dR_1R_2}, 0.01\right)$ as stated in Theorem 1.5. Then,

$$m^*\left(X_S, \mathcal{D}_{X_S}, H'_\gamma, \varepsilon, 0.01\right) \leqslant m^*\left(X_S, \mathcal{D}_{X_S}, H', \varepsilon, 0.01\right) \leqslant m^*\left(X, \mathcal{D}_X, H, \varepsilon, 0.01\right) \leqslant m.$$

By Lemma 5.7, the sample complexity is

$$O\left((d')^2m^{20}\log^{25}\frac{1}{\gamma}\log^2\left(\frac{dR_1R_2}{\varepsilon}\right)\right), \tag{11}$$

where $d' = \dim(S)$. From Lemma B.4 with the parameters C and κ chosen as in Algorithm 4, and given that

$$\frac{C\kappa}{\sqrt{d}R_1} = \frac{\varepsilon}{\sqrt{2d}R_1R_2} \leqslant 1,$$

(which satisfies the required condition in Lemma B.4), we also have

$$d' \lesssim m^*\left(X_S, \mathcal{D}_{X_S}, H_S, \frac{\varepsilon^2}{16\sqrt{2d}R_1R_2}, 0.01\right) \leqslant m^*\left(X, \mathcal{D}_X, H, \frac{\varepsilon^2}{16\sqrt{2d}R_1R_2}, 0.01\right) \leqslant m. \tag{12}$$

Substituting the value of γ as chosen in Algorithm 4 and (12) to (11), we get sample complexity of Algorithm 4 is

$$O\left(\operatorname{poly}(m)\operatorname{polylog}\left(\frac{R_1R_2}{\varepsilon}\right)\right).$$

Bounding the Success Probability. By Lemma 5.7 and 6.1, the success probability of Algorithm 4 is at least 0.7. □

A.12 Proof of Corollary 6.2

Proof. We run $O\left(\log \frac{1}{\delta}\right)$ independent copies of Algorithm 4. Let A be the event where more than 60% of the returned hypotheses $\hat{h}_i$ satisfy $\operatorname{err}\left(\hat{h}_i\right) \leqslant 17\varepsilon$. By Theorem 1.5 and the Chernoff bound, event A occurs with probability at least $1-\delta$.

Conditioned on event A, any ball of radius 34ε centered at a hypothesis more than 68ε away from h^* has probability at most 0.4, as it contains no hypothesis with error at most 17ε from h^*. Conversely, a ball of radius 34ε centered at a hypothesis with error at most 17ε has probability greater than 0.6. Therefore, conditioned on A, selecting the hypothesis whose center forms the heaviest 34ε ball ensures it is at most 68ε away from h^*. □

B DIMENSION REDUCTION

As mentioned in Section 6.1, our algorithm operates in a parameter space that depends on the dimension d, which is unnecessary. To address this, we first apply a dimension reduction procedure Algorithm 5 and then run Algorithm 3 on the resulting subspace. We use $\operatorname{dist}(x, S) := \arg\min_{s\in S} \|x - s\|_2$ to denote the distance between x and the subspace S. We then define the (C, κ)-significant subspace as follows.

Definition B.1 ((C,κ)-Significant Subspace)**.** *A subspace $S \subseteq \mathbb{R}^d$ is called (C,κ)-significant if*

$$\Pr_{x\sim\mathcal{D}_X}\left[\operatorname{dist}(x, S) \geqslant C\kappa\right] \leqslant \kappa.$$

Intuitively, this definition implies that most points in X are close to the subspace S. Therefore, learning the best hypothesis within this subspace would result in a small prediction error. The following lemma quantifies this observation.

Lemma B.2. *Let S be a $\left(\frac{\sqrt{2}}{R_2\varepsilon}, \frac{\varepsilon^2}{2}\right)$-significant subspace and θ' be the projection of θ onto S for any $\theta \in \mathbb{R}^d$, then under Assumption 1.3,*

$$\|h_\theta - h_{\theta'}\|_2^{\mathcal{D}_X} \leqslant \varepsilon.$$

Proof. For x satisfies $\operatorname{dist}(x, S) \leqslant \frac{\varepsilon}{\sqrt{2}R_2}$,

$$|h_\theta(x) - h_{\theta'}(x)| = \left|\left(\theta - \theta'\right)^\top x\right| \leqslant \left\|\theta - \theta'\right\|_2 \|x\|_2 \leqslant R_2\frac{\varepsilon}{\sqrt{2}R_2} = \frac{\varepsilon}{\sqrt{2}},$$

where the first inequality is Cauchy–Schwarz. Therefore,

$$\|h_\theta - h_{\theta'}\|_2^{\mathcal{D}_X} \leqslant \sqrt{\frac{\varepsilon^2}{2} + \left(1 - \frac{\varepsilon^2}{2}\right)\left(\frac{\varepsilon}{\sqrt{2}}\right)^2} \leqslant \varepsilon.$$

□

Algorithm 5: Dimension Reduction Algorithm

```
Algorithm DimensionReduction$(X, C, \kappa)$
    $i \leftarrow 0$
    $S_i \leftarrow \{0\}$
    $V_i \leftarrow \varnothing$
    while $S_i$ is not a $(C, \kappa)$-subspace do
        Pick an orthonormal basis $\{b_1, \cdots, b_{d-i}\}$ for $S_i^{\perp}$
        $v_{i+1} := \arg\max_{j \in [d-i]} \Pr_{x \in \mathcal{D}_X}\left[\langle x, b_j \rangle \geqslant \frac{C\kappa}{\sqrt{d}}\right]$
        $V_i \leftarrow V_i \cup \{v_{i+1}\}$
        $S_i \leftarrow \operatorname{span}(V_i)$
        $i \leftarrow i + 1$
    end
    return $V_i$ and $S_i$
```

The full description of the dimension reduction algorithm is given below as Algorithm 5, where we use $S_i^{\perp}$ to denote the complement of S_i.

The following lemma shows the correctness of Algorithm 5 and one useful property of the returned basis.

Lemma B.3. *Algorithm 5 returns a (C, κ)-significant subspace. Let V be the basis of the subspace S returned by Algorithm 5, then each vector v_i in the basis V satisfies*

$$\Pr_{x \sim \mathcal{D}_X}\left[\langle x, v_i \rangle \geqslant \frac{C\kappa}{\sqrt{d}}\right] \geqslant \frac{\kappa}{d}.$$

Proof. It is evident that Algorithm 5 terminates and returns a (C, κ)-subspace, since it increases the dimension by one in every iteration. During each iteration, when the current subspace S_i is not a (C, κ)-subspace, it means that

$$\Pr_{x \in \mathcal{D}_X}\left[\left\|x - \operatorname*{proj}_{S_i}(x)\right\|_2 \geqslant C\kappa\right] \geqslant \kappa,$$

where $\operatorname{proj}_{S_i}(x)$ denote the projection of x onto S_i. Since $x - \operatorname{proj}_{S_i}(x) \in S_i^{\perp}$, it is a linearly combination of $\{b_1, \cdots, b_{d-i}\}$. By Pigeonhole Principle, for every x satisfying $\left\|x - \operatorname{proj}_{S_i}(x)\right\|_2 \geqslant C\kappa$, there exists a b_j^x in the basis such that $\left\langle x, b_j^x \right\rangle \geqslant \frac{C\kappa}{\sqrt{d}}$. Since $|\{b_1, \cdots, b_{d-i}\}| \leqslant d$, again by Pigeonhole Principle, there exists a b_j such that, among all x satisfies $\left\|x - \operatorname{proj}_{S_i}(x)\right\| \geqslant C\kappa$, at least $\frac{1}{d}$ fraction satisfies $\langle x, b_j \rangle \geqslant \frac{C\kappa}{\sqrt{d}}$. Therefore, there exists a $b_j \in \{b_1, \cdots, b_{d-i}\}$ such that

$$\Pr_{x \sim \mathcal{D}_X}\left[\langle x, b_j \rangle \geqslant \frac{C\kappa}{\sqrt{d}}\right] \geqslant \frac{\kappa}{d}.$$

Because we pick v_{i+1} maximize such probability, it also has this property. □

Furthermore, we can relate the dimension of the subspace S to m^*, the optimal query complexity, as shown below.

Lemma B.4. *Let S be the subspace returned by* DimensionReduction(X, C, κ)*. Define H_S as the hypothesis class parameterized by vectors in S, and let X_S be the projection of X onto S. Further, assume that $\frac{C\kappa}{\sqrt{d}R_1} \leqslant 1$ and then,*

$$m^*\left(X_S, \mathcal{D}_{X_S}, H_S, \frac{C\kappa^{\frac{3}{2}}}{dR_1}, 0.45\right) \gtrsim \dim(S).$$

Proof. Let V be the orthogonal basis returned by DimensionReduction(X, C, κ), and define $V' = \frac{1}{R_1}V$ with $d' = \dim(S)$. To simplify the proof, we introduce a two-player game in Definition C.7. Note that for every $x \in X$, we have

$$\sum_{i \in [d']} \left(x^\top v_i\right)^2 \leqslant \left(\frac{\|x\|_2}{R_1}\right)^2 \leqslant 1,$$

as required in Definition C.7. This game is strictly easier than our active learning problem, as the player can query any $x \in \mathbb{R}^{d'}$, whereas in active learning, the learner can only query $x \in X$. Thus, we apply Lemma C.8 and conclude that with fewer than $\frac{d'}{200}$ queries, no algorithm can separate the hypothesis class $H_{V'}$, parameterized by V', from $h_{\mathbf{0}}$ with probability greater than 0.55.

By Lemma B.3, each $h_v \in H_{V'}$ satisfies

$$\|h_v - h_{\mathbf{0}}\|_2^{\mathcal{D}_{X_S}} \geqslant \sqrt{\frac{\kappa}{d} \cdot \left(\sigma\left(\frac{C\kappa}{\sqrt{d}R_1}\right) - \frac{1}{2}\right)^2}.$$

Note that for $|x| \leqslant 1$, $\left(\sigma(x) - \frac{1}{2}\right) \geqslant \frac{x^2}{32}$ so

$$\|h_v - h_{\mathbf{0}}\|_2^{\mathcal{D}_{X_S}} \geqslant \frac{C\kappa^{\frac{3}{2}}}{4\sqrt{2}dR_1}.$$

Therefore, from the definition of optimal query complexity m^*,

$$m^*\left(X_S, \mathcal{D}_{X_S}, H_S, \frac{C\kappa^{\frac{3}{2}}}{8\sqrt{2}dR_1}, 0.45\right) \geqslant \frac{d'}{200}.$$

□

C COMPLEMENTARY LEMMAS AND DEFINITIONS

This divergence decomposition lemma is adapted from Lattimore and Szepesvári (2020, Lemma 15.1). Although originally stated in the context of bandit problems, it applies directly to our setting, as our problem can be viewed as a special case of the bandit problem, where each x corresponds to an arm with a Bernoulli distribution.

Lemma C.1 (Divergence Decomposition)**.** *Let $\nu = (P_1, \cdots, P_k)$ be the reward distributions associated with one k-armed bandit, and let $\nu' = (P'_1, \cdots, P'_k)$ be the reward distributions associated with another k-armed bandit. Fix some policy π and let $\mathbb{P}_\nu = \mathbb{P}_{\nu\pi}$ and $\mathbb{P}_{\nu'} = \mathbb{P}_{\nu'\pi}$ be the probability measures on the canonical bandit model (Section 4.6 in Lattimore and Szepesvári (2020)) induced by the n-round interconnection of π and ν (respectively, π and ν'). Then,*

$$D_{\mathrm{KL}}\left(\mathbb{P}_\nu \| \mathbb{P}_{\nu'}\right) = \sum_{i=1}^{k} \mathop{\mathbb{E}}_{\nu}\left[T_i(n)\right] D_{\mathrm{KL}}\left(P_i \| P'_i\right).$$

The following lemma gives an upper bound of the KL divergence under the clipping assumption.

Lemma C.2. *For $p, q \in [\gamma, 1-\gamma]$,*

$$D_{\mathrm{KL}}\left(p \| q\right) \lesssim |p - q| \log \frac{1}{\gamma}.$$

Proof. WLOG, we assume that $q \leqslant p$. Then there are two cases.

Case one: $p - q \leqslant \frac{1}{2}p$. In this case, we have

$$D_{\mathrm{KL}}(p\|q) \leqslant p \log \frac{p}{q} = -p \log\left(\frac{p - (p-q)}{p}\right) = -p \log\left(1 - \frac{p-q}{p}\right) \leqslant 2\left|p - q\right|.$$

The last inequality comes from $1 - x \geqslant \exp(-2x)$ for $x \leqslant \frac{1}{2}$.

Case two: $p - q > \frac{1}{2}p$. In this case, we have

$$D_{\mathrm{KL}}\left(p\|q\right) \leqslant p \log \frac{p}{q} \leqslant p \log \frac{1}{\gamma} \leqslant 2\left|p - q\right| \log \frac{1}{\gamma}.$$

□

The following lemma gives a relation of the proportion of “bad hypotheses” (far away from h^*) and the potential growth.

Lemma C.3. *Under Assumption 5.1, if x satisfies*

$$\Pr_{h \sim \lambda}\left[D_{\mathrm{KL}}\left(h^*(x) \,\|\, h(x)\right) \geqslant \alpha\right] \geqslant \beta,$$

then

$$\left(\bar{h}_\lambda(x) - h^*(x)\right)^2 + \left(\operatorname*{Var}_{h \sim \lambda}\left[h(x)\right]\right)^2 \geqslant \left(\operatorname*{Var}_{h \sim \lambda}\left[h(x)\right]\right)^2 \gtrsim \frac{\beta \alpha^2}{\log^2\left(\frac{1}{\gamma}\right)}.$$

Proof. To simplify the notation, we drop the parameter x. If $\left(\bar{h} - h^*\right)^2 \geqslant \frac{\alpha^2}{4 \log^2 \frac{1}{\gamma}}$, then the statement is true, so we assume $\left|\bar{h} - h^*\right| < \frac{\alpha}{2 \log \frac{1}{\gamma}}$. Otherwise, note that

$$\begin{aligned}
&D_{\mathrm{KL}}\left(\bar{h} \| h\right) - D_{\mathrm{KL}}\left(h^* \| h\right) \\
&= \left(\bar{h} \log \frac{\bar{h}}{h} - h^* \log \frac{h^*}{h}\right) + \left(\left(1 - \bar{h}\right) \log \frac{1 - \bar{h}}{1 - h} - \left(1 - h^*\right) \log \frac{1 - h^*}{1 - h}\right) \\
&\geqslant - \left|h - h^*\right| \max\left\{\log \frac{\bar{h}}{h}, \log \frac{1 - \bar{h^*}}{1 - h}, \log \frac{1 - \bar{h}}{1 - h}, \log \frac{1 - h^*}{1 - h}\right\} \\
&\geqslant - \left|h - h^*\right| \log \frac{1}{\gamma}.
\end{aligned}$$

Then it follows that

$$D_{\mathrm{KL}}\left(\bar{h} \| h\right) = \geqslant D_{\mathrm{KL}}\left(h^* \| h\right) - |\bar{h} - h^*| \log \frac{1}{\gamma} \geqslant D_{\mathrm{KL}}\left(h^* \| h\right) - \frac{\alpha}{2}.$$

By applying Lemma C.2, we get if $D_{\mathrm{KL}}\left(h^* \| h\right) \geqslant \alpha$, then

$$\left|\bar{h} - h\right| \gtrsim \frac{D_{\mathrm{KL}}\left(\bar{h} \| h\right)}{\log \frac{1}{\gamma}} \geqslant \frac{D_{\mathrm{KL}}\left(h^* \| h\right) - \frac{\alpha}{2}}{\log \frac{1}{\gamma}} \gtrsim \frac{\alpha}{\log \frac{1}{\gamma}}.$$

Therefore, we know that there are more than β fraction of the h satisfying $\left|\bar{h} - h\right| \gtrsim \frac{\alpha}{\log \frac{1}{\gamma}}$. Then by definition, we have

$$\left(\operatorname*{Var}_{h \sim \lambda}\left[h\right]\right)^2 \gtrsim \frac{\beta \alpha^2}{\log^2 \frac{1}{\gamma}}.$$

□

The following lemma shows $\hat{\lambda}_{(i,j)}$ is not too concentrated for any (i, j).

Lemma C.4. *Let $\hat{\lambda}_{(i,j)} = \frac{1}{2}\lambda^0 + \frac{1}{2}\lambda^1_{(i,j)}$ be the distribution defined in Algorithm 3. Then, for any $h \in H$, the probability that $\hat{\lambda}_{(i,j)}$ assigns to the ball $B_h(\varepsilon)$ is at most 0.8; that is,*

$$\hat{\lambda}_{(i,j)}\left(B_h(\varepsilon)\right) \leqslant 0.8.$$

Proof. We interpret the sampling procedure as follows:

- With some probability distribution, we obtain a pair of hypotheses (h_1, h_2).
- We then output one of these hypotheses, chosen uniformly at random.

For every such pair (h_1, h_2), the hypotheses are at least 2ε apart; that is, $\|h_1 - h_2\| \geqslant 2\varepsilon$. This implies that neither h_1 nor h_2 lies within a radius-ε ball centered at the other hypothesis. Consider any fixed radius-ε ball $B \subseteq H$. Given a pair (h_1, h_2), the probability that a randomly selected hypothesis from the pair lies within B is at most 0.5. This is because at most one of h_1 or h_2 can be in B, since they are at least 2ε apart. Taking the expectation over all possible pairs (h_1, h_2) and applying the law of total probability, we conclude that for any $h \in H$:

$$\hat{\lambda}_{(i,j)}\left(B_h(\varepsilon)\right) \leqslant 0.5 < 0.8.$$

This completes the proof. □

The following lemma relates the information function r_λ to the lower bound of expected potential gain in Lemma 4.1.

Lemma C.5. *Under Assumption 5.1, for any distribution λ over H_γ and any $x \in X$:*

$$\left(\operatorname*{Var}_{h \sim \lambda} [h(x)] \right)^2 \gtrsim \frac{r_\lambda^4(x)}{\log^4 \frac{1}{\gamma}}.$$

Proof. For bookkeeping, in the following we omit x and use h to denote $h(x)$. By Lemma C.2 and Jensen's inequality,

$$r^2 = \left(\mathop{\mathbb{E}}_{h \sim \lambda} \left[D_{\mathrm{KL}} \left(\bar{h}, h \right) \right] \right)^2 \leqslant \mathop{\mathbb{E}}_{h \sim \lambda} \left[D_{\mathrm{KL}}^2 \left(\bar{h}, h \right) \right] \lesssim \mathop{\mathbb{E}}_{h \sim \lambda} \left[\left(\bar{h} - h \right)^2 \log^2 \frac{1}{\gamma} \right] = \operatorname*{Var}_{h \sim \lambda} [h] \log^2 \frac{1}{\gamma}.$$

Consequently,

$$\left(\operatorname*{Var}_{h \sim \lambda} [h] \right)^2 \gtrsim \frac{r^4}{\log^4 \frac{1}{\gamma}}.$$

□

The following lemma relates the probability of a small radius ball centered at θ_{h*} in the parameter space to the PDF of h^*. Recall that $B_h(r)$ denote a ball in the hypothesis class with center h and radius r where the distance is measured by the weighted ℓ_2-distance $\|\cdot\|_2^{\mathcal{D}_X}$. On the other hand, $B_\theta(r)$ denotes a ball in the parameter space with center θ and radius r where the distance is measured by the ℓ_2-distance $\|\cdot\|_2$.

Lemma C.6. *Let $\xi = \frac{\varepsilon}{2R_1 R_2 (m*)^{50} d^4 \log^2 \frac{1}{\alpha}}$, $B = B_{\theta*} \left(\frac{\xi\gamma}{\log^{50} \frac{1}{\gamma} \log^2 \frac{1}{\xi}} \right)$ and d be the dimension of parameter space, where $B_\theta(r)$ denotes a ball centered at θ with radius r measure in ℓ_2 distance in the parameter space. Under Assumption 5.1, the following three properties are true:*

1. *$B \subseteq B_{h*}(\varepsilon)$.*
2. *Let T be the number of queries made and λ be any distribution on H,*

$$\log \lambda(B) \gtrsim \log \lambda\left(h^*\right) - \frac{T\xi R_2}{\log^{50} \frac{1}{\gamma} \log^2 \frac{1}{\xi}} - d \log \left(\frac{1}{\xi\gamma} \right).$$

3. *For any phase j and iteration i,*

$$\tilde{\psi}_{(j,i)}(B) \gtrsim \tilde{\psi}_{(j,i)}\left(h^*\right) - \frac{T\xi R_2}{\log^{50} \frac{1}{\gamma} \log^2 \frac{1}{\xi}} - d \log \left(\frac{1}{\xi\gamma} \right).$$

Proof. The sigmoid function has the property that $|\sigma(a) - \sigma(b)| \leqslant |a - b|$. Then for any $\theta \in B$ and $x \in X$, we have

$$|h_\theta(x) - h_{\theta*}(x)| = \left| \sigma(\theta^\top x) - \sigma((\theta^*)^\top x) \right| \leqslant \left| \theta^\top x - (\theta^*)^\top x \right| \leqslant \|\theta - \theta^*\| \, \|x\|,$$

where the last inequality follows from the Cauchy-Schwarz inequality. By definition of B, we have $\|\theta - \theta^*\| \leqslant \frac{\varepsilon}{2R_1 R_2}$, so $\|\theta - \theta^*\| \, \|x\| \leqslant \varepsilon$. Therefore, $\|h_\theta - h_{\theta*}\|_2 \leqslant \varepsilon$ and $B \subseteq B_{h*}(\varepsilon)$, which proves the first property.

Using the same argument, we can show a stronger upper bound for any $\theta \in B$ and $x \in X$,

$$|h_\theta(x) - h_{\theta*}(x)| \leqslant \frac{\xi\gamma R_2}{\log^{50} \frac{1}{\gamma} \log^2 \frac{1}{\xi}}.$$

Consequently, by the definition of the penalty function and Assumption 5.1, for any query and label pair (x, y), we have for any $\theta \in B$,

$$\begin{aligned} \exp\left(-\ell_{h_\theta}(x, y)\right) &\geqslant \exp\left(-\ell_{h_{\theta*}}(x, y)\right) - \frac{\xi\gamma R_2}{\log^{50} \frac{1}{\gamma} \log^2 \frac{1}{\xi}} \\ &\geqslant \left(1 - \frac{\xi R_2}{\log^{50} \frac{1}{\gamma} \log^2 \frac{1}{\xi}} \right) \exp\left(-\ell_{h_{\theta*}}(x, y)\right). \end{aligned}$$

This means for any $\theta \in B$,

$$w(h_\theta) \geqslant \left(1 - \frac{\xi R_2}{\log^{50}\frac{1}{\gamma}\log^2\frac{1}{\xi}}\right)^{\top} w(h_{\theta *}),$$

where $w(h)$ is the weight of h incurred by the T queries. Let $\lambda(h)$ denote the PDF of h after normalizing the weights, then

$$\lambda(h_\theta) \geqslant \left(1 - \frac{\xi R_2}{\log^{50}\frac{1}{\gamma}\log^2\frac{1}{\xi}}\right)^{\top} \lambda(h_{\theta *}).$$

Note that B has volume $\Omega\left(\left(\frac{\xi\gamma}{d\log^{50}\frac{1}{\gamma}\log^2\frac{1}{\xi}}\right)^d\right)$. Therefore, by integrating over the ball, we have

$$\lambda(B) \gtrsim \left(\frac{\xi\gamma}{d\log^{50}\frac{1}{\gamma}\log^2\frac{1}{\xi}}\right)^d \cdot \left(1 - \frac{\xi R_2}{\log^{50}\frac{1}{\gamma}\log^2\frac{1}{\xi}}\right)^{\top} \lambda(h_{\theta *}).$$

Taking the log of both side and we get

$$\log\lambda(B) \gtrsim \log\lambda\left(h_{\theta *}\right) - \frac{T\xi R_2}{\log^{50}\frac{1}{\gamma}\log^2\frac{1}{\xi}} - d\log\left(\frac{1}{\xi\gamma}\right).$$

Note that the above inequality holds for any function λ proportional to a probability density (i.e., not necessarily normalized), so that for any h'

$$\log\tilde{\lambda}_{(j,i)}^{H\setminus B_{h'}(2\varepsilon)}(B) \gtrsim \log\tilde{\lambda}_{(j,i)}^{H\setminus B_{h'}(2\varepsilon)}\left(h_{\theta *}\right) - \frac{T\xi R_2}{\log^{50}\frac{1}{\gamma}\log^2\frac{1}{\xi}} - d\log\left(\frac{1}{\xi\gamma}\right).$$

Taking the expectation of h' from distribution λ^0 and we finish the proof of the third property. □

We define the following two-player game, which is employed in the proof of Lemma B.4.

Definition C.7 (Two-Player Game). *Let $\{b_i\}_{i=1}^{d'}$ be an orthonormal basis for a subspace $S \subseteq \mathbb{R}^d$ of dimension d', and define*

$$\Theta := \{b_i, -b_i : i \in [d']\}.$$

The environment selects a ground truth parameter $\theta^* \in \Theta \cup \{\mathbf{0}\}$. In each round, the player chooses an arbitrary query vector $a \in \mathbb{R}^d$ subject to the constraint

$$\|a\|_2^2 = \sum_{i=1}^{d'}\left(a^\top b_i\right)^2 \leqslant 1.$$

Subsequently, the environment returns the label 1 with probability $\sigma(a^\top\theta^*)$, and 0 otherwise, where σ denotes the sigmoid function. The player's objective is to determine whether θ^* is the zero vector.

The following lemma establishes that any strategy achieving a success probability greater than 0.55 must make a number of queries that grows linearly with d'.

Lemma C.8. *In the game defined in Definition C.7, any player strategy that correctly determines whether θ^* is the zero vector with probability exceeding 0.55 must issue at least $\frac{d'}{200}$ queries.*

Proof. Suppose we have an algorithm A that uses m queries to determine whether the ground truth θ^* is $\mathbf{0}$ or not and it succeed with probability more than 0.55 on every θ^* the environment chooses. Let X_0 and Y_0 be the queries and responses of A when the ground truth is $\mathbf{0}$. Let X_b and Y_b be the queries and responses when the ground truth is $b \in \Theta$. Together, (X_b, Y_b) and (X_0, Y_0) denote the transcript under b and $\mathbf{0}$ respectively. For any query a, let Y_0^a be the response if the ground truth is $\mathbf{0}$, and let Y_b^a be the response if the ground truth is b. By the definition of the KL divergence, for any query vector a, we have

$$\begin{aligned} D_{\mathrm{KL}}\left(Y_0^a \| Y_b^a\right) &= \log\frac{1}{2} + \frac{1}{2}\log\frac{1}{\sigma(a^\top b)\left(1 - \sigma(a^\top b)\right)} \\ &\leqslant 4\left(\sigma(a^\top b) - \frac{1}{2}\right)^2, \end{aligned}$$

where the final inequality follows from the fact that $\sigma(a^\top b)$ is confined to the interval $\left[\sigma(-1), \sigma(1)\right]$ and by applying an upper bound on the function $\log \frac{1}{x(1-x)}$ for x in this range. Using a first-order approximation of the sigmoid function σ, we have:

- If $a^\top \theta \geqslant 0$, then
$$\sigma\left(a^\top \theta\right) \leqslant \frac{1}{2} + a^\top \theta.$$
- If $a^\top \theta \leqslant 0$, then
$$\sigma\left(a^\top \theta\right) \geqslant \frac{1}{2} + a^\top \theta,$$

which implies $\sigma\left(a^\top \theta\right) \in \left[\frac{1}{2} - \left|a^\top \theta\right|, \frac{1}{2} + \left|a^\top \theta\right|\right]$. Therefore, for any query a,

$$D_{\mathrm{KL}}\left(Y_0^a \| Y_b^a\right) \leqslant 4\left(a^\top b\right)^2. \tag{13}$$

For any $b \in \Theta$, by the definition of total variation distance,

$$\Pr\left[A \text{ cannot distinguish whether } \theta^* = \mathbf{0}\right] \geqslant \frac{1}{2}\left(1 - D_{\mathrm{TV}}\left((X_0, Y_0), (X_b, Y_b)\right)\right).$$

Since A has failure probability less than 0.45, $D_{\mathrm{TV}}\left((X_0, Y_0), (X_b, Y_b)\right) \leqslant 0.1$. Applying Pinsker's inequality, for any $b \in \Theta$:

$$\frac{1}{50} \leqslant 2D_{\mathrm{TV}}^2\left((X_0, Y_0), (X_b, Y_b)\right) \leqslant D_{\mathrm{KL}}\left((X_0, Y_0) \| (X_b, Y_b)\right).$$

Let $T_0(a)$ denote the expected number of times query a is made when running algorithm A for m queries and the ground truth is $\mathbf{0}$. Then, by Lemma C.1 (Lattimore and Szepesvári, 2020)[Lemma 15.1], we have:

$$D_{\mathrm{KL}}\left((X_0, Y_0) \| (X_b, Y_b)\right) = \int_{\mathbb{R}^d} T_0(a) D_{\mathrm{KL}}\left(Y_0^a \| Y_b^a\right) da.$$

Taking average over $b \in \Theta$,

$$\begin{aligned}
\frac{1}{50} &\leqslant \frac{1}{2d'} \sum_{b \in \Theta} D_{\mathrm{KL}}\left((X_0, Y_0) \| (X_b, Y_b)\right) \\
&\leqslant \frac{2}{d'} \sum_{b \in \Theta} \int_{\mathbb{R}^d} T_0(a)\left(a^\top b\right)^2 da \\
&= \frac{2}{d'} \int_{\mathbb{R}^d} T_0(a) \sum_{b \in \Theta}\left(a^\top b\right)^2 da \\
&\leqslant \frac{4}{d'} \int_{\mathbb{R}^d} T_0(a)\, da \\
&= \frac{4m}{d'},
\end{aligned}$$

where the last inequality comes from the assumption of a. Rearrange and we conclude

$$m \geqslant \frac{d'}{200}.$$

□