Noise-Tolerant Interactive Learning Using Pairwise Comparisons

Yichong Xu[*], Hongyang Zhang[*], Kyle Miller[†], Aarti Singh[*], and Artur Dubrawski[†]

[*]Machine Learning Department, Carnegie Mellon University, USA
[†]Auton Lab, Carnegie Mellon University, USA
{yichongx, hongyanz, aarti, awd}@cs.cmu.edu,
mille856@andrew.cmu.edu

Abstract

We study the problem of interactively learning a binary classifier using noisy labeling and pairwise comparison oracles, where the comparison oracle answers which one in the given two instances is more likely to be positive. Learning from such oracles has multiple applications where obtaining direct labels is harder but pairwise comparisons are easier, and the algorithm can leverage both types of oracles. In this paper, we attempt to characterize how the access to an easier comparison oracle helps in improving the label and total query complexity. We show that the comparison oracle reduces the learning problem to that of learning a threshold function. We then present an algorithm that interactively queries the label and comparison oracles and we characterize its query complexity under Tsybakov and adversarial noise conditions for the comparison and labeling oracles. Our lower bounds show that our label and total query complexity is almost optimal.

1 Introduction

Given high costs of obtaining labels for big datasets, interactive learning is gaining popularity in both practice and theory of machine learning. On the practical side, there has been an increasing interest in designing algorithms capable of engaging domain experts in two-way queries to facilitate more accurate and more effort-efficient learning systems (c.f. [26, 31]). On the theoretical side, study of interactive learning has led to significant advances such as exponential improvement of query complexity over passive learning under certain conditions (c.f. [5, 6, 7, 15, 19, 27]). While most of these approaches to interactive learning fix the form of an oracle, e.g., the labeling oracle, and explore the best way of querying, recent work allows for multiple diverse forms of oracles [12, 13, 16, 33]. The focus of this paper is on this latter setting, also known as active dual supervision [4]. We investigate how to recover a hypothesis h that is a good approximator of the optimal classifier h^*, in terms of expected 0/1 error $\Pr_X[h(X) \neq h^*(X)]$, given limited access to labels on individual instances $X \in \mathcal{X}$ and pairwise comparisons about which one of two given instances is more likely to belong to the +1/-1 class.

Our study is motivated by important applications where comparisons are easier to obtain than labels, and the algorithm can leverage both types of oracles to improve label and total query complexity. For example, in material design, synthesizing materials for specific conditions requires expensive experimentation, but with an appropriate algorithm we can leverage expertize of material scientists, for whom it may be hard to accurately assess the resulting material properties, but who can quickly compare different input conditions and suggest which ones are more promising. Similarly, in clinical settings, precise assessment of each individual patient's health status can be difficult, expensive and/or risky (e.g. it may require application of invasive sensors or diagnostic surgeries), but comparing relative statuses of two patients at a time may be relatively easy and accurate. In both these scenarios we may have access to a modest amount of individually labeled data, but the bulk of more accessible training information is available via pairwise comparisons. There are many other examples where

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

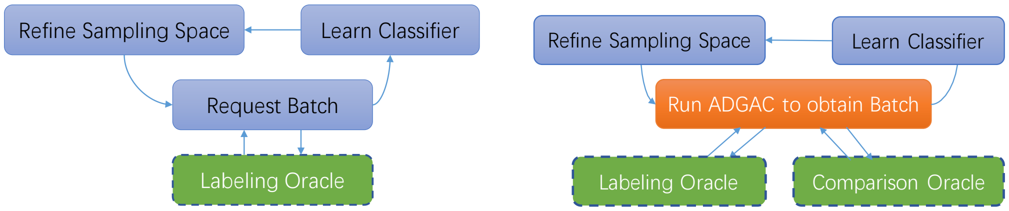

Figure 1: Explanation of work flow of ADGAC-based algorithms. **Left:** Procedure of typical active learning algorithms. **Right:** Procedure of our proposed ADGAC-based interactive learning algorithm which has access to both pairwise comparison and labeling oracles.

Table 1: Comparison of various methods for learning of generic hypothesis class (Omitting $\log(1/\varepsilon)$ factors).

Label Noise	Work	# Label	# Query	$\mathsf{Tol}_{\mathrm{comp}}$
Tsybakov (κ)	[18]	$\tilde{\mathcal{O}}\left(\left(\frac{1}{\varepsilon}\right)^{2\kappa-2} d\theta\right)$	$\tilde{\mathcal{O}}\left(\left(\frac{1}{\varepsilon}\right)^{2\kappa-2} d\theta\right)$	N/A
Tsybakov (κ)	**Ours**	$\tilde{\mathcal{O}}\left(\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\right)$	$\tilde{\mathcal{O}}\left(\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\theta + d\theta\right)$	$\mathcal{O}(\varepsilon^{2\kappa})$
Adversarial ($\nu = \mathcal{O}(\varepsilon)$)	[19]	$\tilde{\mathcal{O}}(d\theta)$	$\tilde{\mathcal{O}}(d\theta)$	N/A
Adversarial ($\nu = \mathcal{O}(\varepsilon)$)	**Ours**	$\tilde{\mathcal{O}}(1)$	$\tilde{\mathcal{O}}(d\theta)$	$\mathcal{O}(\varepsilon^2)$

humans find it easier to perform pairwise comparisons rather than providing direct labels, including content search [17], image retrieval [31], ranking [21], etc.

Despite many successful applications of comparison oracles, many fundamental questions remain. One of them is how to design *noise-tolerant*, *cost-efficient* algorithms that can approximate the unknown target hypothesis to arbitrary accuracy while having access to pairwise comparisons. On one hand, while there is theoretical analysis on the pairwise comparisons concerning the task of learning to rank [3, 22], estimating ordinal measurement models [28] and learning combinatorial functions [11], much remains unknown how to extend these results to more generic hypothesis classes. On the other hand, although we have seen great progress on using single or multiple oracles with the same form of interaction [9, 16], classification using both comparison and labeling queries remains an interesting open problem. Independently of our work, Kane et al. [23] concurrently analyzed a similar setting of learning to classify using both label and comparison queries. However, their algorithms work only in the noise-free setting.

Our Contributions: Our work addresses the aforementioned issues by presenting a new algorithm, Active Data Generation with Adversarial Comparisons (ADGAC), which learns a classifier with both noisy labeling and noisy comparison oracles.

- We analyze ADGAC under Tsybakov (TNC) [30] and adversarial noise conditions for the labeling oracle, along with the adversarial noise condition for the comparison oracle. Our general framework can augment any active learning algorithm by replacing the batch sampling in these algorithms with ADGAC. Figure 1 presents the work flow of our framework.
- We propose A^2-ADGAC algorithm, which can learn an arbitrary hypothesis class. The label complexity of the algorithm is as small as learning a threshold function under both TNC and adversarial noise condition, independently of the structure of the hypothesis class. The *total query complexity* improves over previous best-known results under TNC which can only access the labeling oracle.
- We derive Margin-ADGAC to learn the class of halfspaces. This algorithm has the same label and total query complexity as A^2-ADGAC, but is computationally efficient.
- We present lower bounds on total query complexity for any algorithm that can access both labeling and comparison oracles, and a noise tolerance lower bound for our algorithms. These lower bounds demonstrate that our analysis is nearly optimal.

An important quantity governing the performance of our algorithms is the adversarial noise level of comparisons: denote by $\mathsf{Tol}_{\mathrm{comp}}(\varepsilon, \delta, \mathcal{A})$ the adversarial noise tolerance level of comparisons that guarantees an algorithm $\mathcal{A}$ to achieve an error of ε, with probability at least $1-\delta$. Table 1 compares our results with previous work in terms of label complexity, total query complexity, and $\mathsf{Tol}_{\mathrm{comp}}$ for generic hypothesis class $\mathbb{C}$ with error ε. We see that our results significantly improve over prior

Table 2: Comparison of various methods for learning of halfspaces (Omitting $\log(1/\varepsilon)$ factors).

Label Noise	Work	# Label	# Query	$\mathsf{Tol}_{\text{comp}}$	Efficient?
Massart	[8]	$\tilde{\mathcal{O}}(d)$	$\tilde{\mathcal{O}}(d)$	N/A	No
Massart	[5]	poly(d)	poly(d)	N/A	Yes
Massart	**Ours**	$\tilde{\mathcal{O}}(1)$	$\tilde{\mathcal{O}}(d)$	$\mathcal{O}(\varepsilon^2)$	Yes
Tsybakov (κ)	[19]	$\tilde{\mathcal{O}}((\frac{1}{\varepsilon})^{2\kappa-2} d\theta)$	$\tilde{\mathcal{O}}((\frac{1}{\varepsilon})^{2\kappa-2} d\theta)$	N/A	No
Tsybakov (κ)	**Ours**	$\tilde{\mathcal{O}}\left((\frac{1}{\varepsilon})^{2\kappa-2}\right)$	$\tilde{\mathcal{O}}\left((\frac{1}{\varepsilon})^{2\kappa-2}+d\right)$	$\mathcal{O}(\varepsilon^{2\kappa})$	Yes
Adversarial $(\nu=\mathcal{O}(\varepsilon))$	[34]	$\tilde{\mathcal{O}}(d)$	$\tilde{\mathcal{O}}(d)$	N/A	No
Adversarial $(\nu=\mathcal{O}(\varepsilon))$	[6]	$\tilde{\mathcal{O}}(d^2)$	$\tilde{\mathcal{O}}(d^2)$	N/A	Yes
Adversarial $(\nu=\mathcal{O}(\varepsilon))$	**Ours**	$\tilde{\mathcal{O}}(1)$	$\tilde{\mathcal{O}}(d)$	$\mathcal{O}(\varepsilon^2)$	Yes

work with the extra comparison oracle. Denote by d the VC-dimension of $\mathbb{C}$ and θ the disagreement coefficient. We also compare the results in Table 2 for learning halfspaces under isotropic log-concave distributions. In both cases, our algorithms enjoy small label complexity that is independent of θ and d. This is helpful when labels are very expensive to obtain. Our algorithms also enjoy better total query complexity under both TNC and adversarial noise condition for efficiently learning halfspaces.

2 Preliminaries

Notations: We study the problem of learning a classifier $h : \mathcal{X} \to \mathcal{Y} = \{-1, 1\}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the instance space and label space, respectively. Denote by $\mathcal{P}_{\mathcal{XY}}$ the distribution over $\mathcal{X} \times \mathcal{Y}$ and let $\mathcal{P}_{\mathcal{X}}$ be the marginal distribution over $\mathcal{X}$. A hypothesis class $\mathbb{C}$ is a set of functions $h : \mathcal{X} \to \mathcal{Y}$. For any function h, define the error of h under distribution D over $\mathcal{X} \times \mathcal{Y}$ as $\mathsf{err}_D(h) = \Pr_{(X,Y)\sim D}[h(X) \neq Y]$. Let $\mathsf{err}(h) = \mathsf{err}_{\mathcal{P}_{\mathcal{XY}}}(h)$. Suppose that $h^* \in \mathbb{C}$ satisfies $\mathsf{err}(h^*) = \inf_{h\in\mathbb{C}} \mathsf{err}(h)$. For simplicity, we assume that such an h^* exists in class $\mathbb{C}$.

We apply the concept of disagreement coefficient from Hanneke [18] for generic hypothesis class in this paper. In particular, for any set $V \subseteq \mathbb{C}$, we denote by $\mathsf{DIS}(V) = \{x \in \mathcal{X} : \exists h_1, h_2 \in V, h_1(x) \neq h_2(x)\}$. The disagreement coefficient is defined as $\theta = \sup_{r>0} \frac{\Pr[\mathsf{DIS}(B(h^*,r))]}{r}$, where $B(h^*, r) = \{h \in \mathbb{C} : \Pr_{X\sim\mathcal{P}_{\mathcal{X}}}[h(X) \neq h^*(X)] \leq r\}$.

Problem Setup: We analyze two kinds of noise conditions for the labeling oracle, namely, adversarial noise condition and Tsybakov noise condition (TNC). We formally define them as follows.

Condition 1 (Adversarial Noise Condition for Labeling Oracle)**.** *Distribution $\mathcal{P}_{\mathcal{XY}}$ satisfies adversarial noise condition for labeling oracle with parameter $\nu \geq 0$, if $\nu = \Pr_{(X,Y)\sim\mathcal{P}_{\mathcal{XY}}}[Y \neq h^*(X)]$.*

Condition 2 (Tsybakov Noise Condition for Labeling Oracle)**.** *Distribution $\mathcal{P}_{\mathcal{XY}}$ satisfies Tsybakov noise condition for labeling oracle with parameters $\kappa \geq 1, \mu \geq 0$, if $\forall h : \mathcal{X} \to \{-1, 1\}, \mathsf{err}(h) - \mathsf{err}(h^*) \geq \mu \Pr_{X\sim\mathcal{P}_{\mathcal{X}}}[h(X) \neq h^*(X)]^\kappa$. Also, h^* is the Bayes optimal classifier, i.e., $h^*(x) = sign(\eta(x) - 1/2)$.* [1] *where $\eta(x) = \Pr[Y = 1|X = x]$. The special case of $\kappa = 1$ is also called Massart noise condition.*

In the classic active learning scenario, the algorithm has access to an unlabeled pool drawn from $\mathcal{P}_{\mathcal{X}}$. The algorithm can then query the labeling oracle for any instance from the pool. The goal is to find an $h \in \mathbb{C}$ such that the error $\Pr[h(X) \neq h^*(X)] \leq \varepsilon^2$. The labeling oracle has access to the input $x \in \mathcal{X}$, and outputs $y \in \{-1, 1\}$ according to $\mathcal{P}_{\mathcal{XY}}$. In our setting, however, an extra comparison oracle is available. This oracle takes as input a pair of instances $(x, x') \in \mathcal{X} \times \mathcal{X}$, and returns a variable $Z(x, x') \in \{-1, 1\}$, where $Z(x, x') = 1$ indicates that x is more likely to be positive, while $Z(x, x') = -1$ otherwise. In this paper, we discuss an adversarial noise condition for the comparison oracle. We discuss about dealing with TNC on the comparison oracle in appendix.

Condition 3 (Adversarial Noise Condition for Comparison Oracle)**.** *Distribution $\mathcal{P}_{\mathcal{XXZ}}$ satisfies adversarial noise with parameter $\nu' \geq 0$, if $\nu' = \Pr[Z(X, X')(h^*(X) - h^*(X')) < 0]$.*

[1] The assumption that h^* is Bayes optimal classifier can be relaxed if the approximation error of h^* can be quantified under assumptions on the decision boundary (c.f. [15]).

[2] Note that we use the disagreement $\Pr[h(X) \neq h^*(X)]$ instead of the excess error $\mathsf{err}(h) - \mathsf{err}(h^*)$ in some of the other literatures. The two conditions can be linked by assuming a two-sided version of Tsybakov noise (see e.g., Audibert 2004).

Table 3: Summary of notations.

Notation	Meaning	Notation	Meaning
$\mathbb{C}$	Hypothesis class	κ	Tsybakov noise level (labeling)
$X, \mathcal{X}$	Instance & Instance space	ν	Adversarial noise level (labeling)
$Y, \mathcal{Y}$	Label & Label space	ν'	Adversarial noise level (comparison)
$Z, \mathcal{Z}$	Comparison & Comparison space	$\mathsf{err}_D(h)$	Error of h on distribution D
d	VC dimension of $\mathbb{C}$	$\mathsf{SC}_{\text{label}}$	Label complexity
θ	Disagreement coefficient	$\mathsf{SC}_{\text{comp}}$	Comparison complexity
h^*	Optimal classifier in $\mathbb{C}$	$\mathsf{Tol}_{\text{label}}$	Noise tolerance (labeling)
g^*	Optimal scoring function	$\mathsf{Tol}_{\text{comp}}$	Noise tolerance (comparison)

Note that we do not make any assumptions on the randomness of Z: $Z(X, X')$ can be either random or deterministic as long as the joint distribution $P_{\mathcal{X}\mathcal{X}\mathcal{Z}}$ satisfies Condition 3.

For an interactive learning algorithm $\mathcal{A}$, given error ε and failure probability δ, let $\mathsf{SC}_{\text{comp}}(\varepsilon, \delta, \mathcal{A})$ and $\mathsf{SC}_{\text{label}}(\varepsilon, \delta, \mathcal{A})$ be the comparison and label complexity, respectively. The query complexity of $\mathcal{A}$ is defined as the sum of label and comparison complexity. Similar to the definition of $\mathsf{Tol}_{\text{comp}}(\varepsilon, \delta, \mathcal{A})$, define $\mathsf{Tol}_{\text{label}}(\varepsilon, \delta, \mathcal{A})$ as the maximum ν such that algorithm $\mathcal{A}$ achieves an error of at most ε with probability $1 - \delta$. As a summary, $\mathcal{A}$ learns an h such that $\Pr[h(X) \neq h^*(X)] \leq \varepsilon$ with probability $1 - \delta$ using $\mathsf{SC}_{\text{comp}}(\varepsilon, \delta, \mathcal{A})$ comparisons and $\mathsf{SC}_{\text{label}}(\varepsilon, \delta, \mathcal{A})$ labels, if $\nu \leq \mathsf{Tol}_{\text{label}}(\varepsilon, \delta, \mathcal{A})$ and $\nu' \leq \mathsf{Tol}_{\text{comp}}(\varepsilon, \delta, \mathcal{A})$. We omit the parameters of $\mathsf{SC}_{\text{comp}}, \mathsf{SC}_{\text{label}}, \mathsf{Tol}_{\text{comp}}, \mathsf{Tol}_{\text{label}}$ if they are clear from the context. We use $\mathcal{O}(\cdot)$ to express sample complexity and noise tolerance, and $\tilde{O}(\cdot)$ to ignore the $\log(\cdot)$ terms. Table 3 summarizes the main notations throughout the paper.

3 Active Data Generation with Adversarial Comparisons (ADGAC)

The hardness of learning from pairwise comparisons follows from the error of comparison oracle: the comparisons are *noisy*, and can be *asymmetric* and *intransitive*, meaning that the human might give contradicting preferences like $x_1 \preccurlyeq x_2 \preccurlyeq x_1$ or $x_1 \preccurlyeq x_2 \preccurlyeq x_3 \preccurlyeq x_1$ (here $\preccurlyeq$ is some preference). This makes traditional methods, e.g., defining a function class $\{h : h(x) = Z(x, \hat{x}), \hat{x} \in \mathcal{X}\}$, fail, because such a class may have infinite VC dimension.

In this section, we propose a novel algorithm, ADGAC, to address this issue. Having access to both comparison and labeling oracles, ADGAC generates a labeled dataset by techniques inspired from group-based binary search. We show that ADGAC can be combined with any active learning procedure to obtain interactive algorithms that can utilize both labeling and comparison oracles. We provide theoretical guarantees for ADGAC.

3.1 Algorithm Description

To illustrate ADGAC, we start with a general active learning framework in Algorithm 1. Many active learning algorithms can be adapted to this framework, such as A^2 [7] and margin-based active algorithms [6, 5]. Here U represents the querying space/disagreement region of the algorithm (i.e., we reject an instance x if $x \notin U$), and V represents a version space consisting of potential classifiers. For example, A^2 algorithm can be adapted to Algorithm 1 straightforwardly by keeping U as the sample space and V as the version space. More concretely, A^2 algorithm [7] for adversarial noise can be characterized by

$$U_0 = \mathcal{X},\ V_0 = \mathbb{C},\ f_V(U, V, W, i) = \{h : |W|\mathsf{err}_W(h) \leq n_i\varepsilon_i\},\ f_U(U, V, W, i) = \mathsf{DIS}(V),$$

where ε_i and n_i are parameters of the A^2 algorithm, and $\mathsf{DIS}(V) = \{x \in \mathcal{X} : \exists h_1, h_2 \in V, h_1(x) \neq h_2(x)\}$ is the disagreement region of V. Margin-based active learning [6] can also be fitted into Algorithm 1 by taking V as the halfspace that (approximately) minimizes the hinge loss, and U as the region within the margin of that halfspace.

To efficiently apply the comparison oracle, we propose to replace step 4 in Algorithm 1 with a subroutine, ADGAC, that has access to both comparison and labeling oracles. Subroutine 2 describes ADGAC. It takes as input a dataset S and a sampling number k. ADGAC first runs Quicksort algorithm on S using feedback from comparison oracle, which is of form $Z(x, x')$. Given that the comparison oracle $Z(\cdot, \cdot)$ might be asymmetric w.r.t. its two arguments, i.e., $Z(x, x')$ may not equal to $Z(x', x)$, for each pair (x_i, x_j), we randomly choose (x_i, x_j) or (x_j, x_i) as the input to $Z(\cdot, \cdot)$. After Quicksort, the algorithm divides the data into multiple groups of size $\alpha m = \varepsilon|\tilde{S}|$, and does

Algorithm 1 Active Learning Framework

```
Input: $\varepsilon, \delta$, a sequence of $n_i$, functions $f_U, f_V$.
1: Initialize $U \leftarrow U_0 \subseteq \mathcal{X}, V \leftarrow V_0 \subseteq \mathbb{C}$.
2: for $i = 1, 2, ..., \log(1/\varepsilon)$ do
3:     Sample unlabeled dataset $\tilde{S}$ of size $n_i$. Let $S \leftarrow \{x : x \in \tilde{S}, x \in U\}$.
4:     Request the labels of $x \in S$ and obtain $W \leftarrow \{(x_i, y_i) : x_i \in S\}$.
5:     Update $V \leftarrow f_V(U, V, W, i)$, $U \leftarrow f_U(U, V, W, i)$.
Output: Any classifier $\hat{h} \in V$.
```

Subroutine 2 Active Data Generation with Adversarial Comparison (ADGAC)

```
Input: Dataset $S$ with $|S| = m$, $n, \varepsilon, k$.
1: $\alpha \leftarrow \frac{\varepsilon n}{2m}$.
2: Define preference relation on $S$ according to $Z$. Run Quicksort on $S$ to rank elements in an
   increasing order. Obtain a sorted list $S = (x_1, x_2, ..., x_m)$.
3: Divide $S$ into groups of size $\alpha m$: $S_i = \{x_{(i-1)\alpha m+1}, ..., x_{i\alpha m}\}, i = 1, 2, ..., 1/\alpha$ .
4: $t_{\min} \leftarrow 1, t_{\max} \leftarrow 1/\alpha$.
5: while $t_{\min} < t_{\max}$ do                                   ▷ Do binary search
6:     $t = (t_{\min} + t_{\max})/2$.
7:     Sample $k$ points uniformly without replacement from $S_t$ and obtain the labels $Y =
       \{y_1, ..., y_k\}$.
8:     If $\sum_{i=1}^{k} y_i \geq 0$, then $t_{\max} = t$; else $t_{\min} = t + 1$.
9: For $t' > t$ and $x_i \in S_{t'}$, let $\hat{y}_i \leftarrow 1$.
10: For $t' < t$ and $x_i \in S_{t'}$, let $\hat{y}_i \leftarrow -1$.
11: For $x_i \in S_t$, let $\hat{y}_i$ be the majority of labeled points in $S_t$.
Output: Predicted labels $\hat{y}_1, \hat{y}_2, ..., \hat{y}_m$.
```

group-based binary search by sampling k labels from each group and determining the label of each group by majority vote.

For active learning algorithm $\mathcal{A}$, let $\mathcal{A}$-ADGAC be the algorithm of replacing step 4 with ADGAC using parameters $(S_i, n_i, \varepsilon_i, k_i)$, where ε_i, k_i are chosen as additional parameters of the algorithm. We establish results for specific $\mathcal{A}$: A^2 and margin-based active learning in Sections 4 and 5, respectively.

3.2 Theoretical Analysis of ADGAC

Before we combine ADGAC with active learning algorithms, we provide theoretical results for ADGAC. By the algorithmic procedure, ADGAC reduces the problem of labeling the whole dataset S to binary searching a threshold on the sorted list S. One can show that the conflicting instances cannot be too many within each group S_i, and thus binary search performs well in our algorithm. We also use results in [3] to give an error estimate of Quicksort. We have the following result based on the above arguments.

Theorem 4. *Suppose that Conditions 2 and 3 hold for $\kappa \geq 1, \nu' \geq 0$, and $n = \Omega\left(\left(\frac{1}{\varepsilon}\right)^{2\kappa-1}\log(1/\delta)\right)$. Assume a set $\tilde{S}$ with $|\tilde{S}| = n$ is sampled i.i.d. from $\mathcal{P}_{\mathcal{X}}$ and $S \subseteq \tilde{S}$ is an arbitrary subset of $\tilde{S}$ with $|S| = m$. There exist absolute constants C_1, C_2, C_3 such that if we run Subroutine 2 with $\varepsilon < C_1$, $\nu' \leq C_2\varepsilon^{2\kappa}\delta$, $k = k^{(1)}(\varepsilon, \delta) := C_3 \log\left(\frac{\log(1/\varepsilon)}{\delta}\right)\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}$, it will output a labeling of S such that $|\{x_i \in S : \hat{y}_i \neq h^*(x_i)\}| \leq \varepsilon n$, with probability at least $1 - \delta$. The expected number of comparisons required is $\mathcal{O}(m \log m)$, and the number of sample-label pairs required is $\mathsf{SC}_{\text{label}}(\varepsilon, \delta) = \tilde{\mathcal{O}}\left(\log\left(\frac{m}{\varepsilon n}\right)\log(1/\delta)\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\right)$.*

Similarly, we analyze ADGAC under adversarial noise condition w.r.t. labeling oracle with $\nu = \mathcal{O}(\varepsilon)$.

Theorem 5. *Suppose that Conditions 1 and 3 hold for $\nu, \nu' \geq 0$, and $n = \Omega\left(\frac{1}{\varepsilon}\log(1/\delta)\right)$. Assume a set $\tilde{S}$ with $|\tilde{S}| = n$ is sampled i.i.d. from $\mathcal{P}_{\mathcal{X}}$ and $S \subseteq \tilde{S}$ is an arbitrary subset of $\tilde{S}$ with $|S| = m$. There exist absolute constants C_1, C_2, C_3, C_4 such that if we run Subroutine 2 with $\varepsilon < C_1$, $\nu' \leq C_2\varepsilon^2\delta$, $k = k^{(2)}(\varepsilon, \delta) := C_3 \log\left(\frac{\log(1/\varepsilon)}{\delta}\right)$, and $\nu \leq C_4\varepsilon$, it will output a labeling* *of S such that $|\{x_i \in S : \hat{y}_i \neq h^*(x_i)\}| \leq \varepsilon n$, with probability at least $1-\delta$. The expected number of comparisons required is $\mathcal{O}(m \log m)$, and the number of sample-label pairs required is* $\mathsf{SC}_{\text{label}}(\varepsilon, \delta) = \mathcal{O}\left(\log\left(\frac{m}{\varepsilon n}\right) \log\left(\frac{\log(1/\varepsilon)}{\delta}\right)\right)$.

Proof Sketch. We call a pair (x_i, x_j) an *inverse pair* if $Z(x_i, x_j) = -1, h^*(x_i) = 1, h^*(x_j) = -1$, and an *anti-sort pair* if $h^*(x_i) = 1, h^*(x_j) = -1$, and $i < j$. We show that the expectation of inverse pairs is $n(n-1)\varepsilon^*$. By the results in [3] the numbers of inverse pairs and anti-sort pairs have the same expectation, and the actual number of anti-sort pairs can be bounded by Markov's inequality. Then we show that the majority label of each group must be all -1 starting from beginning the list, and changes to all 1 at some point of the list. With a careful choice of k, we may obtain the true majority with k labels under Tsybakov noise; we will thus end up in the turning point of the list. The error is then bounded by the size of groups. See appendix for the complete proof.

Theorems 4 and 5 show that ADGAC gives a labeling of dataset with arbitrary small error using label complexity *independent* of the data size. Moreover, ADGAC is computationally efficient since it only involves binary search. These nice properties of ADGAC lead to improved query complexity when we combine ADGAC with other active learning algorithms.

4 A^2-ADGAC: Learning of Generic Hypothesis Class

In this section, we combine ADGAC with A^2 algorithm to learn a generic hypothesis class. We use the framework in Algorithm 1: let A^2-ADGAC be the algorithm that replaces step 4 in Algorithm 1 with ADGAC of parameters $(S, n_i, \varepsilon_i, k_i)$, where n_i, ε_i, k_i are parameters to be specified later. Under TNC, we have the following result.

Theorem 6. *Suppose that Conditions 2 and 3 hold, and $h^*(x) = sign(\eta(x) - 1/2)$. There exist global constants C_1, C_2 such that if we run A^2-ADGAC with $\varepsilon < C_1, \delta$, $\nu' \leq \mathsf{Tol}_{\text{comp}}(\varepsilon, \delta) = C_2\varepsilon^{2\kappa}\delta$, $\varepsilon_i = 2^{-(i+2)}, n_i = \Omega\left(\frac{1}{\varepsilon_i}(d\log(1/\varepsilon)) + \left(\frac{1}{\varepsilon_i}\right)^{2\kappa-1}\log(1/\delta)\right)$, $k_i = k^{(1)}\left(\varepsilon_i, \frac{\delta}{4\log(1/\varepsilon)}\right)$ with $k^{(1)}$ specified in Theorem 4, with probability at least $1-\delta$, the algorithm will return a classifier $\hat{h}$ with $\Pr[\hat{h}(X) \neq h^*(X)] \leq \varepsilon$ with comparison and label complexity*

$$\mathbb{E}[\mathsf{SC}_{\text{comp}}] = \tilde{\mathcal{O}}\left(\theta \log^2\left(\frac{1}{\varepsilon}\right)\log(d\theta)\left(\left(d\log\left(\frac{1}{\varepsilon}\right)\right) + \left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\log(1/\delta)\right)\right),$$

$$\mathsf{SC}_{\text{label}} = \tilde{\mathcal{O}}\left(\log\left(\frac{1}{\varepsilon}\right)\log\left(\min\left\{\frac{1}{\varepsilon}, \theta\right\}\right)\log(1/\delta)\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\right).$$

The dependence on $\log^2(1/\varepsilon)$ in $\mathsf{SC}_{\text{comp}}$ can be reduced to $\log(1/\varepsilon)$ under Massart noise.

We can prove a similar result for adversarial noise condition.

Theorem 7. *Suppose that Conditions 1 and 3 hold. There exist global constants C_1, C_2, C_3 such that if we run A^2-ADGAC with $\varepsilon < C_1, \delta$, $\nu' \leq \mathsf{Tol}_{\text{comp}}(\varepsilon, \delta) = C_2\varepsilon^2\delta, \nu \leq \mathsf{Tol}_{\text{label}}(\varepsilon, \delta) = C_3\varepsilon$, $\varepsilon_i = 2^{-(i+2)}, n_i = \tilde{\Omega}\left(\frac{1}{\varepsilon_i}d\log\left(\frac{1}{\varepsilon_i}\right)\log(1/\delta)\right), k_i = k^{(2)}\left(\varepsilon_i, \frac{\delta}{4\log(1/\varepsilon)}\right)$ with $k^{(2)}$ specified in Theorem 5, with probability at least $1-\delta$, the algorithm will return a classifier $\hat{h}$ with $\Pr[\hat{h}(X) \neq h^*(X)] \leq \varepsilon$ with comparison and label complexity*

$$\mathbb{E}[\mathsf{SC}_{\text{comp}}] = \tilde{\mathcal{O}}\left(\theta d\log(\theta d)\log\left(\frac{1}{\varepsilon_i}\right)\log(1/\delta)\right),$$

$$\mathsf{SC}_{\text{label}} = \tilde{\mathcal{O}}\left(\log\left(\frac{1}{\varepsilon}\right)\log\left(\min\left\{\frac{1}{\varepsilon}, \theta\right\}\right)\log(1/\delta)\right).$$

Proof of Theorems 6 and 7 uses Theorem 4 and Theorem 5 with standard manipulations in VC theory. Theorems 6 and 7 show that having access to even a biased comparison function can reduce the problem of learning a classifier in high-dimensional space to that of learning a threshold classifier in one-dimensional space as the label complexity matches that of actively learning a threshold classifier. Given the fact that comparisons are usually easier to obtain, A^2-ADGAC will save a lot in practice due to its small label complexity. More importantly, we improve the total query complexity under TNC by separating the dependence on d and ε; The query complexity is now the sum of the two terms instead of the product of them. This observation shows the power of pairwise comparisons for learning classifiers. Such small label/query complexity is impossible without access to a comparison oracle, since query complexity with only labeling oracle is at least $\Omega\left(d\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\right)$ and $\Omega\left(d\log\left(\frac{1}{\varepsilon}\right)\right)$ under TNC and adversarial noise conditions, respectively [19]. Our results also matches the lower bound of learning with labeling and comparison oracles up to log factors (see Section 6).

We note that Theorems 6 and 7 require rather small $\mathsf{Tol}_{\text{comp}}$, equal to $\mathcal{O}(\varepsilon^{2\kappa}\delta)$ and $\mathcal{O}(\varepsilon^{2}\delta)$, respectively. We will show in Section 6.3 that it is necessary to require $\mathsf{Tol}_{\text{comp}} = \mathcal{O}(\varepsilon^2)$ in order to obtain a classifier of error ε, if we restrict the use of labeling oracle to only learning a threshold function. Such restriction is able to reach the near-optimal label complexity as specified in Theorems 6 and 7.

5 Margin-ADGAC: Learning of Halfspaces

In this section, we combine ADGAC with margin-based active learning [6] to efficiently learn the class of halfspaces. Before proceeding, we first mention a naive idea of utilizing comparisons: we can i.i.d. sample pairs (x_1, x_2) from $\mathcal{P}_{\mathcal{X}} \times \mathcal{P}_{\mathcal{X}}$, and use $Z(x_1, x_2)$ as the label of $x_1 - x_2$, where Z is the feedback from comparison oracle. However, this method cannot work well in our setting without additional assumption on the noise condition for the labeling $Z(x_1, x_2)$.

Before proceeding, we assume that $\mathcal{P}_{\mathcal{X}}$ is isotropic log-concave on $\mathbb{R}^d$; i.e., $\mathcal{P}_{\mathcal{X}}$ has mean 0, covariance I and the logarithm of its density function is a concave function [5, 6]. The hypothesis class of halfspaces can be represented as $\mathbb{C} = \{h : h(x) = \text{sign}(w \cdot x), w \in \mathbb{R}^d\}$. Denote by $h^*(x) = \text{sign}(w^* \cdot x)$ for some $w^* \in \mathbb{R}^d$. Define $l_\tau(w, x, y) = \max\left(1 - y(w \cdot x)/\tau, 0\right)$ and $l_\tau(w, W) = \frac{1}{|W|}\sum_{(x,y)\in W} l_\tau(w, x, y)$ as the hinge loss. The expected hinge loss of w is $L_\tau(w, D) = \mathbb{E}_{x\sim D}[l_\tau(w, x, \text{sign}(w^* \cdot x))]$.

Margin-based active learning [6] is a concrete example of Algorithm 1 by taking V as (a singleton set of) the hinge loss minimizer, while taking U as the margin region around that minimizer. More concretely, take $U_0 = \mathcal{X}$ and $V_0 = \{w_0\}$ for some w_0 such that $\theta(w_0, w^*) \le \pi/2$. The algorithm works with constants $M \ge 2, \kappa < 1/2$ and a set of parameters r_i, τ_i, b_i, z_i that equal to $\Theta(M^{-i})$ (see proof in Appendix for formal definition of these parameters). V always contains a single hypothesis. Suppose $V = \{w_{i-1}\}$ in iteration $i-1$. Let v_i satisfies $l_{\tau_i}(v_i, W) \le \min_{v:\|v-w_{i-1}\|_2\le r_i, \|v\|_2\le 1} l_{\tau_i}(v, W) + \kappa/8$, where w_i is the content of V in iteration i. We also have $f_V(V, W, i) = \{w_i\} = \left\{\frac{v_i}{\|v_i\|_2}\right\}$ and $f_U(U, V, W, i) = \{x : |w_i \cdot x| \le b_i\}$.

Let Margin-ADGAC be the algorithm obtained by replacing the sampling step in margin-based active learning with ADGAC using parameters $(S, n_i, \varepsilon_i, k_i)$, where n_i, ε_i, k_i are additional parameters to be specified later. We have the following results under TNC and adversarial noise conditions, respectively.

Theorem 8. *Suppose that Conditions 2 and 3 hold, and $h^*(x) = sign(w^* \cdot x) = sign(\eta(x) - 1/2)$. There are settings of $M, \kappa, r_i, \tau_i, b_i, \varepsilon_i, k_i$, and constants C_1, C_2 such that for all $\varepsilon \le C_1, \nu' \le \mathsf{Tol}_{\text{comp}}(\varepsilon, \delta) = C_2\varepsilon^{2\kappa}\delta$, if we run Margin-ADGAC with w_0 such that $\theta(w_0, w^*) \le \pi/2$, and $n_i = \tilde{O}\left(\frac{1}{\varepsilon_i} d\log^3(dk/\delta) + \left(\frac{1}{\varepsilon}\right)^{2\kappa-1}\log(1/\delta)\right)$, it finds $\hat{w}$ such that* $\Pr[$*sign*$(\hat{w} \cdot X) \ne$ *sign*$(w^* \cdot X)] \le \varepsilon$ *with probability at least* $1 - \delta$. *The comparison and label complexity are*

$$\mathbb{E}[\mathsf{SC}_{\text{comp}}] = \tilde{\mathcal{O}}\left(\log^2(1/\varepsilon)\left(d\log^4(d/\delta) + \left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\log(1/\delta)\right)\right),$$

$$\mathsf{SC}_{\text{label}} = \tilde{\mathcal{O}}\left(\log(1/\varepsilon)\log(1/\delta)\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\right).$$

The dependence on $\log^2(1/\varepsilon)$ *in* $\mathsf{SC}_{\text{comp}}$ *can be reduced to* $\log(1/\varepsilon)$ *under Massart noise.*

Theorem 9. *Suppose that Conditions 1 and 3 hold. There are settings of $M, \kappa, r_i, \tau_i, b_i, \varepsilon_i, k_i$, and constants C_1, C_2, C_3 such that for all $\varepsilon \le C_1, \nu' \le \mathsf{Tol}_{\text{comp}}(\varepsilon, \delta) = C_2\varepsilon^{2\kappa}\delta, \nu \le \mathsf{Tol}_{\text{comp}}(\varepsilon, \delta) = C_3\varepsilon$, if we run Margin-ADGAC with $n_i = \tilde{\mathcal{O}}\left(\frac{1}{\varepsilon_i} d\log^3(dk/\delta)\right)$ and w_0 such that $\theta(w_0, w^*) \le \pi/2$, it finds $\hat{w}$ such that* $\Pr[$*sign*$(\hat{w} \cdot X) \ne$ *sign*$(w^* \cdot X)] \le \varepsilon$ *with probability at least* $1 - \delta$. *The comparison and label complexity are*

$$\mathbb{E}[\mathsf{SC}_{\text{comp}}] = \tilde{\mathcal{O}}\left(\log(1/\varepsilon)\left(d\log^4(d/\delta)\right)\right), \quad \mathsf{SC}_{\text{label}} = \tilde{\mathcal{O}}\left(\log(1/\varepsilon)\log(1/\delta)\right).$$

The proofs of Theorems 8 and 9 are different from the conventional analysis of margin-based active learning in two aspects: a) Since we use labels generated by ADGAC, which is not independently sampled from the distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$, we require new techniques that can deal with adaptive noises; b) We improve the results of [6] over the dependence of d by new Rademacher analysis.

Theorems 8 and 9 enjoy better label and query complexity than previous results (see Table 2). We mention that while Yan and Zhang [32] proposed a perceptron-like algorithm with label complexity as small as $\tilde{\mathcal{O}}(d\log(1/\varepsilon))$ under Massart and adversarial noise conditions, their algorithm works only under uniform distributions over the instance space. In contrast, our algorithm Margin-ADGAC works under broad log-concave distributions. The label and total query complexity of Margin-ADGAC improves over that of traditional active learning. The lower bounds in Section 6 show the optimality of our complexity.

6 Lower Bounds

In this section, we give lower bounds on learning using labeling and pairwise comparison. In Section 6.1, we give a lower bound on the optimal label complexity $\mathsf{SC}_{\text{label}}$. In Section 6.2 we use this result to give a lower bound on the total query complexity, i.e., the sum of comparison and label complexity. Our two methods match these lower bounds up to log factors. In Section 6.3, we additionally give an information-theoretic bound on $\mathsf{Tol}_{\text{comp}}$, which matches our algorithms in the case of Massart and adversarial noise.

Following from [19, 20], we assume that there is an underlying score function g^* such that $h^*(x) = \text{sign}(g^*(x))$. Note that g^* does not necessarily have relation with $\eta(x)$; We only require that $g^*(x)$ represents how likely a given x is positive. For instance, in digit recognition, $g^*(x)$ represents how an image looks like a 7 (or 9); In the clinical setting, $g^*(x)$ measures the health condition of a patient. Suppose that the distribution of $g^*(X)$ is continuous, i.e., the probability density function exists and for every $t \in \mathbb{R}$, $\Pr[g^*(X) = t] = 0$.

6.1 Lower Bound on Label Complexity

The definition of g^* naturally induces a comparison oracle Z with $Z(x, x') = \text{sign}(g^*(x) - g^*(x'))$. We note that this oracle is invariant to shifting w.r.t. g^*, i.e., g^* and $g^* + t$ lead to the same comparison oracle. As a result, we cannot distinguish g^* from $g^* + t$ without labels. In other words, pairwise comparisons do not help in improving label complexity when we are learning a threshold function on $\mathbb{R}$, where all instances are in the natural order. So the label complexity of any algorithm is lower bounded by that of learning a threshold classifier, and we formally prove this in the following theorem.

Theorem 10. *For any algorithm $\mathcal{A}$ that can access both labeling and comparison oracles, sufficiently small ε, δ, and any score function g that takes at least two values on $\mathcal{X}$, there exists a distribution $P_{\mathcal{X}\mathcal{Y}}$ satisfying Condition 2 such that the optimal function is in the form of $h^*(x) = sign(g(x) + t)$ for some $t \in \mathbb{R}$ and*

$$\mathsf{SC}_{\text{label}}(\varepsilon, \delta, \mathcal{A}) = \Omega\left((1/\varepsilon)^{2\kappa-2}\log(1/\delta)\right). \tag{1}$$

If $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$ satisfies Condition 1 with $\nu = O(\varepsilon)$, $\mathsf{SC}_{\text{label}}$ satisfies (1) *with $\kappa = 1$.*

The lower bound in Theorem 10 matches the label complexity of A^2-ADGAC and Margin-ADGAC up to a log factor. So our algorithm is near-optimal.

6.2 Lower Bound on Total Query Complexity

We use Theorem 10 to give lower bounds on the total query complexity of any algorithm which can access both comparison and labeling oracles.

Theorem 11. *For any algorithm $\mathcal{A}$ that can access both labeling and comparison oracles, and sufficiently small ε, δ, there exists a distribution $P_{\mathcal{X}\mathcal{Y}}$ satisfying Condition 2, such that*

$$\mathsf{SC}_{\text{comp}}(\varepsilon, \delta, \mathcal{A}) + \mathsf{SC}_{\text{label}}(\varepsilon, \delta, \mathcal{A}) = \Omega\left((1/\varepsilon)^{2\kappa-2}\log(1/\delta) + d\log(1/\varepsilon)\right). \tag{2}$$

If $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$ satisfies Condition 1 with $\nu = O(\varepsilon)$, $\mathsf{SC}_{\text{comp}} + \mathsf{SC}_{\text{label}}$ satisfies (2) *with $\kappa = 1$.*

The first term of (2) follows from Theorem 10, whereas the second term follows from transforming a lower bound of active learning with access to only the labeling oracle. The lower bounds in Theorem 11 match the performance of A^2-ADGAC and Margin-ADGAC up to log factors.

6.3 Adversarial Noise Tolerance of Comparisons

Note that label queries are typically expensive in practice. Thus it is natural to ask the following question: what is the minimal requirement on ν', given that we are only allowed to have access to minimal label complexity as in Theorem 10? We study this problem in this section. More concretely, we study the requirement on ν' when we learn a threshold function using labels. Suppose that the comparison oracle gives feedback using a scoring function $\hat{g}$, i.e., $Z(x, x') = \text{sign}(\hat{g}(x) - \hat{g}(x'))$, and has error ν'. We give a sharp minimax bound on the risk of the optimal classifier in the form of $h(x) = \text{sign}(\hat{g}(x) - t)$ for some $t \in \mathbb{R}$ below.

Theorem 12. *Suppose that* $\min\{\Pr[h^*(X) = 1], \Pr[h^*(X) = -1]\} \geq \sqrt{\nu'}$ *and both* $\hat{g}(X)$ *and* $g^*(X)$ *have probability density functions. If* $\hat{g}(X)$ *induces an oracle with error* ν'*, then we have* $\min_t \max_{\hat{g}, g^*} \Pr[\textit{sign}(\hat{g}(X) - t) \neq h^*(X)] = \sqrt{\nu'}$.

The proof is technical and omitted. By Theorem 12, we see that the condition of $\nu' = \varepsilon^2$ is necessary if labels from g^* are only used to learn a threshold on $\hat{g}$. This matches our choice of ν' under Massart and adversarial noise conditions for labeling oracle (up to a factor of δ).

7 Conclusion

We presented a general algorithmic framework, ADGAC, for learning with both comparison and labeling oracles. We proposed two variants of the base algorithm, A^2-ADGAC and Margin-ADGAC, to facilitate low query complexity under Tsybakov and adversarial noise conditions. The performance of our algorithms matches lower bounds for learning with both oracles. Our analysis is relevant to a wide range of practical applications where it is easier, less expensive, and/or less risky to obtain pairwise comparisons than labels.

There are multiple directions for future works. One improvement over our work is to show complexity bounds for excess risk $\mathsf{err}(h) - \mathsf{err}(h^*)$ instead of $\Pr[h \neq h^*]$. Also, our bound on comparison complexity is in expectation due to limits of quicksort; deriving concentration inequalities on the comparison complexity would be helpful. Also, an adaptive algorithm that applies to different levels of noise w.r.t. labels and comparisons would be interesting; i.e., use labels when comparisons are noisy and use comparisons when labels are noisy. Other directions include using comparisons (or more broadly, rankings) for other ML tasks like regression or matrix completion.

Acknowledgments

This research is supported in part by AFRL grant FA8750-17-2-0212. We thank Chicheng Zhang for insightful ideas on improving results in [6] using Rademacher complexity.

References

[1] S. Agarwal and P. Niyogi. Stability and generalization of bipartite ranking algorithms. In *Annual Conference on Learning Theory*, pages 32–47, 2005.

[2] S. Agarwal and P. Niyogi. Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research*, 10:441–474, 2009.

[3] N. Ailon and M. Mohri. An efficient reduction of ranking to classification. *arXiv preprint arXiv:0710.2889*, 2007.

[4] J. Attenberg, P. Melville, and F. Provost. A unified approach to active dual supervision for labeling features and examples. In *Machine Learning and Knowledge Discovery in Databases*, pages 40–55. Springer, 2010.

[5] P. Awasthi, M.-F. Balcan, N. Haghtalab, and H. Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Annual Conference on Learning Theory*, pages 152–192, 2016.

[6] P. Awasthi, M.-F. Balcan, and P. M. Long. The power of localization for efficiently learning linear separators with noise. *Journal of the ACM*, 63(6):50, 2017.

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

[8] M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Annual Conference On Learning Theory*, pages 35–50, 2007.

[9] M.-F. Balcan and S. Hanneke. Robust interactive learning. In *COLT*, pages 20–1, 2012.

[10] M.-F. Balcan and P. M. Long. Active and passive learning of linear separators under log-concave distributions. In *Annual Conference on Learning Theory*, pages 288–316, 2013.

[11] M.-F. Balcan, E. Vitercik, and C. White. Learning combinatorial functions from pairwise comparisons. *arXiv preprint arXiv:1605.09227*, 2016.

[12] M.-F. Balcan and H. Zhang. Noise-tolerant life-long matrix completion via adaptive sampling. In *Advances in Neural Information Processing Systems*, pages 2955–2963, 2016.

[13] A. Beygelzimer, D. J. Hsu, J. Langford, and C. Zhang. Search improves label for active learning. In *Advances in Neural Information Processing Systems*, pages 3342–3350, 2016.

[14] S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375, 2005.

[15] R. M. Castro and R. D. Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353, 2008.

[16] O. Dekel, C. Gentile, and K. Sridharan. Selective sampling and active learning from single and multiple teachers. *Journal of Machine Learning Research*, 13:2655–2697, 2012.

[17] J. Fürnkranz and E. Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer, 2010.

[18] S. Hanneke. Adaptive rates of convergence in active learning. In *COLT*. Citeseer, 2009.

[19] S. Hanneke. Theory of active learning, 2014.

[20] S. Hanneke and L. Yang. Surrogate losses in passive and active learning. *arXiv preprint arXiv:1207.3772*, 2012.

[21] R. Heckel, N. B. Shah, K. Ramchandran, and M. J. Wainwright. Active ranking from pairwise comparisons and the futility of parametric assumptions. *arXiv preprint arXiv:1606.08842*, 2016.

[22] K. G. Jamieson and R. Nowak. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems*, pages 2240–2248, 2011.

[23] D. M. Kane, S. Lovett, S. Moran, and J. Zhang. Active classification with comparison queries. *arXiv preprint arXiv:1704.03564*, 2017.

[24] A. Krishnamurthy. *Interactive Algorithms for Unsupervised Machine Learning*. PhD thesis, Carnegie Mellon University, 2015.

[25] L. Lovász and S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007.

[26] S. Maji and G. Shakhnarovich. Part and attribute discovery from relative annotations. *International Journal of Computer Vision*, 108(1-2):82–96, 2014.

[27] S. Sabato and T. Hess. Interactive algorithms: from pool to stream. In *Annual Conference On Learning Theory*, pages 1419–1439, 2016.

[28] N. B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. Wainwright. When is it better to compare than to score? *arXiv preprint arXiv:1406.6618*, 2014.

[29] N. Stewart, G. D. Brown, and N. Chater. Absolute identification by relative judgment. *Psychological review*, 112(4):881, 2005.

[30] A. B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *Annals of Statistics*, pages 135–166, 2004.

[31] C. Wah, G. Van Horn, S. Branson, S. Maji, P. Perona, and S. Belongie. Similarity comparisons for interactive fine-grained categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–866, 2014.

[32] S. Yan and C. Zhang. Revisiting perceptron: Efficient and label-optimal active learning of halfspaces. *arXiv preprint arXiv:1702.05581*, 2017.

[33] L. Yang and J. G. Carbonell. Cost complexity of proactive learning via a reduction to realizable active learning. Technical report, CMU-ML-09-113, 2009.

[34] C. Zhang and K. Chaudhuri. Beyond disagreement-based agnostic active learning. In *Advances in Neural Information Processing Systems*, pages 442–450, 2014.

A Our Techniques

Intransitivity: The main challenge of learning with pairwise comparisons is that the comparisons might be *asymmetric* or *intransitive*. If we construct a classifier $h(x)$ by simply comparing x with a fixed instance $\hat{x}$ by comparison oracle, then the concept class of classifiers $\{h : h(x) = Z(x, \hat{x}), \hat{x} \in \mathcal{X}\}$ will have infinite VC dimension, so the complexity will be as high as infinite if we apply the traditional tools of VC theory. To resolve the issue, we conduct a group-based binary search in ADGAC. The intuition is that by dividing the dataset into several ranked groups $S_1, S_2, ...$, the majority of labels in each group can be stably decided if we sample enough examples from that group. Therefore, we are able to reduce the original problem in the high-dimensional space to the problem of learning a "threshold" function in one-dimension space. Then some straightforward approaches such as binary search learns the thresholding function.

Combining with Active Learning Algorithms: If the labels follow Tsybakov noise (i.e., Condition 2), the most straightforward method to combine ADGAC with existing algorithms is to combine ADGAC with an algorithm that uses the label oracle only and works under TNC. However, we cannot save query complexity if we follow this method. To see this, notice that in each round we need roughly $n_i = \tilde{\mathcal{O}}\left(d\theta\left(\frac{1}{\varepsilon_i}\right)^{2\kappa-1}\right)$ samples and $m_i = \tilde{\mathcal{O}}\left(d\theta\left(\frac{1}{\varepsilon_i}\right)^{2\kappa-2}\right)$ labels; if we use ADGAC, we can obtain a labeling of n_i samples with at most $\varepsilon_i n_i \approx m_i$ errors with low label complexity. Suppose N is the set of labels that ADGAC makes error on. However, since the outside active learning algorithm works under TNC, we will need to query labels in N to make sure that the ADGAC labels follow TNC. That means our label complexity is still m_i, the same as the original algorithm. To avoid this problem, we combine ADGAC with algorithms under adversarial noise in all cases including TNC. This eliminates the need to query additional labels, and also reduces the query complexity.

Handling Independence: We mostly follow previous works on combining ADGAC with existing algorithms. However, since we now obtain labels from ADGAC instead of $\mathcal{P}_{\mathcal{XY}}$, the labels are not independently sampled, and we need to adapt the proof to our case. We use different methods for A^2-ADGAC and Margin-ADGAC: For the former, we use results from PAC learning to bound the error on all n_i samples; for the latter, we decompose the error of any classifier h on labels generated by ADGAC into two parts: The first part is caused by the error of ADGAC itself, and second is by h on truthful labels. Using the above techniques enables us to circumvent the independence problem.

Lower Bounds: It is typically hard to provide a unified lower bound for multi-query learning framework, as several quantities are simultaneously involved in the analysis, e.g., the comparison complexity, the label complexity, the noise tolerance, etc. So traditional proof techniques for active learning, e.g., Le Cam's and Fano's bounds [15, 19], cannot be trivially applied to our setting. Instead, we prove lower bounds on one quantity by allowing arbitrary budgets of other quantities. Another non-trivial technique is in the proof of minimax bound for the adversarial noise level of comparison oracle (see Theorem 12): In the proof of upper bound, we divide the integral region w.r.t. the expectation into n segments, each of size $1/n$, and the expectation is thus the limit when $n \to \infty$. We upper bound the discrete approximation of the integral by a careful calibration of noise on each segment for a fixed n, and then let $n \to \infty$. The proof then leads to a general inequality (Lemma 21), and it might be of independent interest.

B Additional Related Work

It is well known that people are better at comparison than labeling [29, 28]. It has been widely used to tackle problems in classification [26], clustering [24] and ranking [2, 17]. Balcan et al. [11] studied using pairwise comparisons to learn submodular functions on sets. Another related problem is bipartite ranking [1], which exactly does the opposite of our problem: Given a group of binary labels, learn a ranking function that rank positive samples higher than negative ones.

Interactive learning has wide application in the field of computer vision and natural language processing (see e.g., [31]). There are also abundant literatures on interactive ways to improve unsupervised and semi-supervised learning [24]. However, there lacks a general statistical analysis of interactive learning for traditional classification tasks. Balcan and Hanneke [9] analyze class conditional queries

(CCQ), where the user gives counterexamples to a given classification. Beygelzimer et al. [13] used a similar idea using search queries. However, their interactions requires a oracle that is usually stronger than the traditional labelers (i.e., we can simulate traditional active learning using such oracles), and is generally hard to deploy in practice. There turns out to be little general analysis on using a ”weaker” interaction between human and computer. Balcan and Hanneke[9] studied an abstract query based notions from exact learning, but their analysis cannot handle queries that gives relation between samples (as comparisons do). Our work fits in this blank.

We compare our work to traditional label-based active learning [19], which has drawn a lot of attention in the society in recent years. Disagreement-based active learning has been shown to reach a near-optimal rate on classification problems [18]. Another line of research is margin-based active learning [5], which aims at computational efficiency of learning halfspaces, under the large-margin assumption.

C Learning under TNC for Comparisons

In this section we justify our choice of analyzing adversarial noise model for the comparison oracle. In fact, any algorithm using adversarial comparisons can be transformed into an algorithm using TNC comparisons, by treating learning comparison functions as a separate learning problem. Let $\mathbb{C}'$ be a hypothesis class consisting of comparison functions $f : \mathcal{X} \times \mathcal{X} \to \{-1, 1\}$. Suppose the optimal comparison function is $f^*(x, x') = \text{sign}(g^*(x) - g^*(x'))$, and Tsybakov noise condition holds for $((X, X'), Z)$ with some constant μ', κ'; i.e., for any $f \in \mathbb{C}'$ we have

$$\Pr[f(X, X') \neq Z] - \Pr[f^*(X, X') \neq Z] \geq \mu' \Pr[f(X, X') \neq f^*(X, X')]^{\kappa'}.$$

Also suppose $f^*(x, x') = \text{sign}(\Pr[Z = 1|X = x, X' = x'] - 1/2)$. Assume $\mathbb{C}'$ has VC-dimension d' and disagreement coefficient θ', standard active learning requires $\Phi(\nu') = \tilde{O}\left(\theta' \left(\frac{1}{\nu'}\right)^{2\kappa'-2} (d' \log(\theta') + \log(1/\delta)) \log\left(\frac{1}{\nu'}\right)\right)$ samples to learn a comparison function of error ν' with probability $1 - \delta$. So an algorithm $\mathcal{A}$ for adversarial noise on comparisons can be automatically transformed into an algorithm $\mathcal{A}'$ for TNC on comparisons with $\mathsf{SC}_{\text{label}}(\mathcal{A}') = \mathsf{SC}_{\text{label}}(\mathcal{A})$ and $\mathsf{SC}_{\text{comp}}(\mathcal{A}) = \Phi(\mathsf{Tol}_{\text{comp}}(\mathcal{A}))$. So we only analyze adversarial noise for comparison in other parts of this paper.

D Proof of Theorem 4

Proof. We only prove the theorem for $\kappa > 1$, the case of $\kappa = 1$ holds with a similar proof. An equivalent condition (see e.g., page 341, [14]) for Condition 2 under $\kappa > 1$ is that there exists constant $\tilde{\mu} > 0$ such that for all $t > 0$ we have

$$\Pr(|\eta(x) - 1/2| < t) \leq \tilde{\mu} t^{1/(\kappa-1)}. \tag{3}$$

We use (3) instead of Condition 2 through out the proof.

To bound the error in labeling by ADGAC, we first bound the number of incorrectly sorted pairs due to noise/bias of the comparison oracle. We call (x_i, x_j) an *inverse pair* if $h^*(x_i) = 1, h^*(x_j) = -1, x_i \preccurlyeq x_j$ (the partial order is decided by randomly querying $Z(x_i, x_j)$ or $Z(x_j, x_i)$). Also, we call (x_i, x_j) an *anti-sort pair* if $h^*(x_i) = 1, h^*(x_j) = -1, i < j$ (after sorting by Quicksort). Let T be the set of all anti-sort pairs, T' be the set of all inverse pairs in S, and $\tilde{T}'$ be the set of all inverse pairs in $\tilde{S}$. We first bound $|T|$ using $|T'|$. Let s be the random bits supplied for Quicksort in its process, by Theorem 3 in [3] we have

$$\mathbb{E}_s[|T|] = |T'|.$$

Notice that sampling a pair of (X, X') is equivalent to sample a set $\tilde{S}$ of n points and then uniformly pick two different points in it. Also, number of inverse pairs in S is less than that in $\tilde{S}$. So we have

$$\mathbb{E}_S\left[\mathbb{E}_s[|T|]\right] = \mathbb{E}_S[|T'|] \leq \mathbb{E}_{\tilde{S}}[|\tilde{T}'|] = n(n-1)\nu' \leq n^2\nu'.$$

By Markov inequality we have

$$\Pr\left(|T| \geq \frac{2\nu'}{\delta} n^2\right) \leq \frac{\delta}{2}. \tag{4}$$

Suppose $|T| < \frac{2\nu'}{\delta}n^2$ (which holds with probability $> 1 - \delta/2$). We now proceed to bound the number of labeling errors made by ADGAC. First, notice that in Algorithm 2, we divide all samples into groups of size $\alpha m = \varepsilon n/2$. For every set S_i, let

$$\begin{aligned} q(S_i) &= \frac{1}{|S_i|} \min\left\{ \sum_{x \in S_i} I(h^*(x) = 1), \sum_{x \in S_i} I(h^*(x) = -1) \right\} \\ &= \min\left\{ \Pr_{X \sim S_i}(h^*(x) = -1), \Pr_{X \sim S_i}(h^*(x) = 1) \right\} \end{aligned}$$

where $X \sim S_i$ denote the empirical distribution that X is drawn uniformly at random from the finite collection of points in S_i. Let

$$\beta = \frac{2}{\varepsilon}\sqrt{\frac{\nu'}{\delta}} \le C\varepsilon^{\kappa - 1}$$

for some constant C. Suppose ε is small enough such that $\beta \le 1/2$. Then we claim that there is at most 1 set S_i such that $q(S_i) \ge \beta$. Otherwise, suppose two such sets exist; let them be S_i and S_j. So there are at least $\alpha\beta m$ points $x \in S_i$ with $h^*(x) = -1$, and $\alpha\beta m$ points $x \in S_i$ with $h^*(x) = 1$; the same holds for S_j. These -1s and 1s would indicate at least

$$2\alpha^2\beta^2 m^2 = \frac{2\nu'}{\delta} n^2$$

anti-sort pairs, which violates our claim of $|T|$.

Since ADGAC uses group binary search, we first analyze some properties of the majority label of the Bayes optimal classifier within each group/set. For each set S_i, let $\mu(S_i) = \text{sign}(\sum_{x \in S_i} h^*(x_i))$ be the majority Bayes optimal label. We can show that $\mu(S_i)$ is monotonic: that is, for every $i < j$ we have $\mu(S_i) \le \mu(S_j)$. To see this, suppose there exist two sets $S_i, S_j, i < j$ such that $\mu(S_i) = 1$ and $\mu(S_j) = -1$. That would indicate at least $\alpha^2 m^2/2 > \alpha^2\beta^2 m^2$ anti-sort pairs, which violates our assumption. So there must be a boundary l such that $\mu(S_i) = -1$ for $i < l$, and $\mu(S_i) = 1$ for $i \ge l$. We call S_l to be the *boundary set*. Now consider two cases:

- Case 1: there exists a set $S_{l'}$ such that $q(S_{l'}) \ge \beta$ (recall that from previous arguments, there is only one such set). If $l \ne l'$, the sets S_l and $S_{l'}$ generates at least $2(\alpha m/2)(\alpha\beta m) \ge 2\alpha^2\beta^2 m^2$ anti-sort pairs, which violates our assumption for $|T|$. So $l = l'$.
- Case 2: for all sets S_i, we have $q(S_i) < \beta$.

In both cases, we have $q(S_i) < \beta$ for all $i \ne l$.

Now we prove that the majority vote of the noisy labels agrees with the majority vote of the Bayes optimal classifier $\mu(S_i)$ for each set S_i that we visit, and hence we will find the boundary set S_l. Suppose $q(S_i) < \beta$. Take

$$t = \left(\frac{\varepsilon}{16\tilde{\mu}}\right)^{\kappa - 1}.$$

For small enough ε, we have $t \le 1/2$, and $\Pr(x : |\eta(x) - 1/2| \le t) \le \varepsilon/16$. Let $U = \{x_i \in S : |\eta(x_i) - 1/2| \le t\}$. By relative form of Chernoff bound we have

$$\Pr\left(|U| > 3\log(4/\delta) + n\varepsilon/8\right) \le \exp\left(-\frac{3\log(4/\delta) + \varepsilon n/16}{3}\right) \le \frac{\delta}{4}.$$

Suppose $|U|/n \le \varepsilon/8$, so at most 1/4 of each S_i is in U.

For each set S_i, let $\bar{S}_i = \{x \in S_i : h^*(x) \neq \mu(S_i)\}$ and $S_i' = \{x \in S_i : |\eta(x) - 1/2| \le t\}$. So for each set such that $q(S_i) \le \beta$, we have

$$\begin{aligned}\Pr(Y \neq \mu(S_i)|X \sim S_i) \le & \Pr(Y \neq \mu(S_i)|X \in S_i') \Pr(X \in S_i'|X \sim S_i) + \\ & \Pr(Y \neq \mu(S_i)|X \in \bar{S}_i) \Pr(X \in \bar{S}_i|X \sim S_i) + \\ & \Pr(Y \neq \mu(S_i)|X \notin S_i', X \notin \bar{S}_i) \Pr(X \notin S_i', X \notin \bar{S}_i|X \sim S_i) \\ \le & \left(\frac{1}{2} + t\right) \cdot \frac{1}{4} + 1 \cdot \beta + \left(\frac{1}{2} - t\right) \cdot \left(\frac{3}{4} - \beta\right) \\ = & \frac{1}{2} - \frac{1}{2}t + \left(\frac{1}{2} + t\right)\beta.\end{aligned}$$

Pick ν' small enough such that $\beta \le \frac{1}{4}t$:

$$\frac{2}{\varepsilon}\sqrt{\frac{\nu'}{\delta}} \le \frac{1}{4}\left(\frac{\varepsilon}{16\tilde{\mu}}\right)^{\kappa-1}.$$

This yields

$$\nu' \le \frac{\varepsilon^{2\kappa}\delta}{32(16\tilde{\mu})^{2\kappa-2}}.$$

Note that this also guarantees $\beta \le 1/2$ above, since $t \le \frac{1}{2}$. Now we have

$$\Pr\left[Y \neq \mu(S_i)|X \sim S_i\right] \le \frac{1}{2} - \frac{1}{2}t + \frac{1}{4}t\,(t + 1/2) \le \frac{1}{2} - \frac{1}{4}t.$$

In the algorithm, suppose we pick $X_1, X_2, ..., X_k \in S_i$ as the points for which to query the label and the labels are $Y_1, ..., Y_k$. Note that since $n = \Omega\left(\left(\frac{1}{\varepsilon}\right)^{2\kappa-1}\log(1/\delta)\right)$, we have $|S_i| = \alpha n \ge k$ for every i, so we will not run out of samples for each set. By Hoeffding's inequality, we have

$$\Pr\left[\text{sign}\left(\sum_{j=1}^{k} Y_j\right) = \mu(S_i)\right] = \Pr\left[\frac{1}{k}\sum_{j=1}^{n} I(Y_j = \mu(S_i)) > \frac{1}{2}\right] \le \exp\left(-\frac{1}{8}kt^2\right).$$

The choice of k yields that the majority vote of the noisy labels agrees with the majority vote $\mu(S_i)$ of the Bayes optimal classifier for each S_i with $q(S_i) \le \beta$ we visit, with probability $\frac{\delta}{8\log(2/\varepsilon)}$.

Suppose the binary search output set S_t (i.e., the value of t at step 9). Now we analyze the errors we made in the final output. We consider the two cases:

- Case 1: If $q(S_l) \ge \beta$, then with probability $1 - \delta$, we have $t \in \{l-1, l, l+1\}$ since we might behave arbitrarily in set S_l. In this case, we have $q(S_l)|S_l| \ge \alpha\beta m$, and so $|\{x : x \in S_{t'}, t' < l, h^*(x) = 1\} \le \alpha\beta m$, because otherwise we have $\alpha^2\beta^2 m^2$ anti-sort pairs, which violates our assumption on $|T|$. Similarly, $|\{x : x \in S_{t'}, t' > l, h^*(x) = -1\}| \le \alpha\beta m$. Counting also the possible errors made on S_l, the total number of errors is
$$|\{\hat{y}_i : \hat{y}_i \neq h^*(x_i)\}| \le \alpha m + 2\alpha\beta m \le \frac{\varepsilon n}{2}\left(1 + \frac{1}{2}t\right) \le \varepsilon n.$$
- Case 2: If $q(S_i) < \beta$ for all i, then we have $t \in \{l-1, l\}$. Now note that we have $|\{x \in S_{l-1} : h^*(x) = -1\}| \ge \alpha m/2$, and so $|\{x \in S_{t'} : t' < l-1, h^*(x) = 1\}| \le \alpha\beta m$ since otherwise at least $\alpha^2\beta m/2$ anti-sort pairs are present. So $|\{x \in S_{t'} : t' \le l - 1, h^*(x) = 1\}| \le 2\alpha\beta m$ considering $q(S_{l-1}) < \beta$. Similarly, $|\{x \in S_{t'} : t' \ge l, h^*(x) = -1\}| \le 2\alpha\beta m$. So the total number of errors is
$$|\{\hat{y}_i : \hat{y}_i \neq h^*(x_i)\}| \le 4\alpha\beta m \le \varepsilon n.$$

So we have at most εn error under both cases. Now we examine the total query complexity: The expected comparison complexity is $O(m\log m)$ even if we have noisy comparisons, see [3]. For label complexity, it takes $k = \tilde{\mathcal{O}}\left(\log(1/\delta)\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\right)$ queries for each set S_t, and we do this for $\mathcal{O}(\log(1/\alpha)) = \mathcal{O}\left(\log\left(\frac{2m}{\varepsilon n}\right)\right)$ times. So the total query complexity is

$$\tilde{\mathcal{O}}\left(\log\left(\frac{2m}{\varepsilon n}\right)\log(1/\delta)\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\right).$$

□

E Proof of Theorem 5

Proof. The first part of proof is exactly the same as that of Theorem 4. We now bound $\Pr[Y \neq \mu(S_i)|X \sim S_i]$. Suppose $q(S_i) < \beta$. Let $V = \{x : \Pr[Y \neq h^*(X)|X = x] > 1/4\}$ and $U = \{x_i : \Pr[Y \neq h^*(X)|X = x_i] > 1/4\}$. We have $P(V) \leq 4\nu$. By a relative Chernoff bound, if $\nu \leq C_1\varepsilon$ for a small enough constant C_1 we have

$$\Pr[|U| \leq 8n\nu + 3\log(4/\delta)] \leq \exp\left(-\frac{3\log(4/\delta) + 4\nu n}{3}\right) \leq \delta/4.$$

So if $\nu \leq \frac{1}{64}\varepsilon$ we have $|U|/n \leq \varepsilon/8$ with probability $\delta/4$. In this case, at most 1/4 of each S_i is in U.

For each set S_i, let $\bar{S}_i = \{x \in S_i : h^*(x) \neq \mu(S_i)\}$ and $\tilde{S}_i = \{x \in S_i, x \in U\}$. So for each set such that $q(S_i) \leq \beta$, we have

$$\begin{aligned}\Pr(Y \neq \mu(S_i)|X \sim S_i) \leq& \Pr(Y \neq \mu(S_i)|X \in \tilde{S}_i)\Pr(X \in \tilde{S}_i|X \sim S_i)+\\ &\Pr(Y \neq \mu(S_i)|X \in \bar{S}_i)\Pr(X \in \bar{S}_i|X \sim S_i)+\\ &\Pr(Y \neq \mu(S_i)|X \notin \tilde{S}_i, X \notin \bar{S}_i)\Pr(X \notin \tilde{S}_i, X \notin \bar{S}_i|X \sim S_i)\\ \leq& 1\cdot\frac{1}{4} + 1\cdot\beta + \left(\frac{3}{4} - \beta\right)\frac{1}{4}\\ =& \frac{7}{16} + \frac{3}{4}\beta.\end{aligned}$$

So there exists constant C_2 such that if $\nu' \leq C_2\varepsilon^2\delta$ we have $\beta \leq \frac{1}{24}$, $\Pr(Y \neq \mu(S_i)|X \sim S_i)] \leq \frac{1}{2} - \frac{1}{32}$. Thus by Hoeffding's inequality, the choice of k yields that we recover $\mu(S_i)$ for each i we visit with probability $\frac{\delta}{8\log(2/\varepsilon)}$.

By similar analysis as the proof of Theorem 4, we can show that the number of errors (i.e., $|\{\hat{y}_i : \hat{y}_i \neq h^*(x_i)\}|$) is at most εn.

Now examine the total query complexity: It takes $k = \mathcal{O}(\log(\log(1/\varepsilon)/\delta))$ queries for each set S_t, and we do this for $\mathcal{O}(\log(1/\alpha)) = \mathcal{O}(\log\left(\frac{2m}{\varepsilon n}\right))$ times. So the total query complexity is

$$\mathcal{O}\left(\log\left(\frac{2m}{\varepsilon n}\right)\log\left(\frac{\log(1/\varepsilon)}{\delta}\right)\right).$$

□

F Proof for A^2-ADGAC

We use the following lemma adapted from [19]:

Lemma 13 ([19], Lemma 3.1). *Suppose that $\mathcal{D} = \{x_1, x_2, ..., x_n\}$ is i.i.d. sampled from $\mathcal{P}_\mathcal{X}$, and $h^* \in \mathbb{C}$. There is a universal constant $c_0 \in (1, \infty)$ such that for any $\gamma \in (0, 1)$, and any $n \in \mathbb{N}$, letting*

$$U(n, \gamma) = c_0\frac{d\log(n/d) + \log(1/\gamma)}{n},$$

with probability at least $1 - \gamma$, $\forall h \in \mathbb{C}$, the following inequalities hold:

$$\Pr_{X\sim\mathcal{P}_\mathcal{X}}[h(X) \neq h^*(X)] \leq \max\{2\Pr_{X\sim\mathcal{D}}[h(X) \neq h^*(X)], U(n, \gamma)\},$$
$$\Pr_{X\sim\mathcal{D}}[h(X) \neq h^*(X)] \leq \max\{2\Pr_{X\sim\mathcal{P}_\mathcal{X}}[h(X) \neq h^*(X)], U(n, \gamma)\}.$$

Here $X \sim \mathcal{D}$ means X is uniformly sampled from finite set $\mathcal{D}$.

Proof of Theorem 6. For a labeled dataset $W = \{(x_i, \hat{y}_i)\}_{i=1}^n$, let $\mathsf{err}_W(h) = \frac{1}{n}\sum_{i=1}^n I(h(x_i) \neq \hat{y}_i)$ be the empirical risk of h on W for any $h \in \mathbb{C}$ (remind that $\hat{y}_i$ are predictions of ADGAC). For a clearer explanation, we formalize the A^2-ADGAC algorithm in Algorithm 3. We use induction to

Algorithm 3 A^2-ADGAC

```
Input: $n_i, \mathbb{C}, \varepsilon, \delta$, comparison oracle $f$.
1: Let $V \leftarrow \mathbb{C}$.
2: for $i = 1, 2, ..., \lceil \log(1/\varepsilon) \rceil$ do
3:     Sample dataset $\tilde{S}$ of size $n_i$.
4:     Let $S \leftarrow \{x \in \tilde{S} : x \in \mathsf{DIS}(V)\}$.
5:     Run ADGAC (Subroutine 2) with $S, \tilde{S}, \varepsilon_i = 2^{-(i+2)}, k_i$ and labeled dataset $W$.
6:     $V = V \setminus \{h : |W|\mathsf{err}_W(h) \geq n_i\varepsilon_i\}$.
Output: Any Classifier $\hat{h} \in V$.
```

prove that after iteration i we have $\Pr[h(X) \neq h^*(X)] \leq 4\varepsilon_i$ for all $h \in V$ after step 6 in Algorithm 3. This proposition holds for $i = 0$. Suppose it holds for $i - 1$. By Theorem 4 and a union bound, with probability $1 - \delta/4$, for every iteration i we have at most $n_i\varepsilon_i$ errors with respect to h^* after running ADGAC, i.e., $|W|\mathsf{err}_W(h^*) \leq n_i\varepsilon_i$. So h^* will not be eliminated from V in any iteration with probability $1 - \delta/4$. On the other hand, notice that by Step 6 in Algorithm 3 all functions $h \in V$ satisfies $|W|\mathsf{err}_W(h) \leq n_i\varepsilon_i$, so by triangle inequality we have (notice that W is just the set S with labels)

$$\begin{aligned}|S| \Pr_{X\sim S}[h(X) \neq h^*(X)] &= |\{x \in S : h(x) \neq h^*(x)\}| \\ &\leq |\{(x, \hat{y}) \in W : h(x) \neq \hat{y}\}| + |\{(x, \hat{y}) \in W : h^*(x) \neq \hat{y}\}| \\ &\leq 2\varepsilon_i n_i.\end{aligned}$$

Also note that functions in V agrees on $\tilde{S} \setminus S$; so $|\tilde{S}| \Pr_{X\sim\tilde{S}}[h(X) \neq h^*(X)] \leq 2\varepsilon_i n_i$, and since $|\tilde{S}| = n_i$ we have $\Pr_{X\sim\tilde{S}}[h(X) \neq h^*(X)] \leq 2\varepsilon_i$. Now using Lemma 13 with $n = n_i$, we have $\Pr_{X\sim\mathcal{P}_\mathcal{X}}[h(x) \neq h^*(x)] \leq 4\varepsilon_i$ for every $h \in V$ by choosing n_i such that $U\left(n_i, \frac{\delta}{4\log(1/\varepsilon)}\right) \leq \varepsilon_i$. So at the end of the algorithm it outputs a classifier with $\Pr[\hat{h} \neq h^*] \leq \varepsilon$.

Now we examine the number of queries. By definition of disagreement coefficient, at round i we have $\mathsf{DIS}(V) \leq \theta\varepsilon_i$; thus using a relative Chernoff bound we know that with probability $1 - \delta/4$ we have

$$m_i := |S| \leq \log(12/\delta) + 2n_i\theta\varepsilon_i = \mathcal{O}\left(\theta\left((d\log(1/\varepsilon)) + \left(\frac{1}{\varepsilon_i}\right)^{2\kappa-2}\log(1/\delta)\right)\right).$$

It takes $O(m_i \log m_i)$ comparisons in expectation to rank the set, and there are $\log(1/\varepsilon)$ iterations. So the total comparison complexity is

$$\mathbb{E}[\mathsf{SC}_{\text{comp}}]=\tilde{\mathcal{O}}\left(\theta\log\left(\frac{1}{\varepsilon}\right)\left(\log d\theta + (\kappa-1)\log\left(\frac{1}{\varepsilon}\right)\right)\left(\left(d\log\left(\frac{1}{\varepsilon}\right)\right) + \left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\log(1/\delta)\right)\right).$$

This obtained the stated comparison complexity. The label complexity follows by multiplying the label complexity of ADGAC by $\log(1/\varepsilon)$. Note that in every step we have $\frac{m_i}{\varepsilon_i n_i} = O\left(\min\left\{\theta, \frac{1}{\varepsilon}\right\}\right)$. □

Proof of Theorem 7. With the same proof, A^2-ADGAC outputs a classifier with $\Pr[\hat{h} \neq h^*] \leq \varepsilon$ upon finishing. We know examine the number of queries. By definition of disagreement coefficient, at round i we have $\mathsf{DIS}(V) \leq \theta\varepsilon_i$; thus using a Chernoff bound we know that with probability $1 - \delta/4$ we have

$$m_i := |S| \leq \log(12/\delta) + 2n_i\theta\varepsilon_i = \mathcal{O}\left(\theta d\log\left(\frac{1}{\varepsilon_i}\right)\log\left(\frac{1}{\delta}\right)\right).$$

It takes $O(m_i \log m_i)$ comparisons in expectation to rank the set, and there are $\log(1/\varepsilon)$ iterations. So the total comparison complexity is

$$\mathbb{E}[\mathsf{SC}_{\text{comp}}] = \tilde{\mathcal{O}}\left(\theta d\log(\theta d)\log\left(\frac{1}{\varepsilon_i}\right)\log\left(\frac{1}{\delta}\right)\right).$$

This obtained the stated comparison complexity. The label complexity follows by multiplying the label complexity of ADGAC by $\log(1/\varepsilon)$. Note that in every step we have $\frac{m_i}{\varepsilon_i n_i} = O\left(\min\left\{\theta, \frac{1}{\varepsilon}\right\}\right)$. □

G Proof for Margin-ADGAC

Algorithm 4 Margin-ADGAC: Efficiently learning halfspaces with comparison

```
Input: $\varepsilon, \delta$, target errors $\varepsilon_k$, sample sizes $n_k$, sequences $r_k, b_k, \tau_k$, precision value $\kappa$.
Draw $n_1$ unlabeled samples to $S$ and run ADGAC with $\left(S, n_1, \varepsilon_0, \frac{\delta}{8\log(1/\varepsilon)}, k_1\left(\varepsilon_0, \frac{\delta}{8\log(1/\varepsilon)}\right)\right)$, and obtain a labeled dataset $W$.
for $k = 1, 2, ..., s = \lceil \log(4/\varepsilon) \rceil$ do
    Find $v_k \in B(w_{k-1}, r_k)$ that approximately minimize training hinge loss over $W$, with $\|v_k\|_2 \le 1$:
        $$l_{\tau_k}(v_k, W) \le \min_{w \in B(w_{k-1}, r_k) \cap B(0,1)} l_{\tau_k}(w, W) + \kappa/8.$$
    $w_k \leftarrow \frac{v_k}{\|v_k\|_2}$.
    Sample another dataset $\tilde{S}$ of $n_k$ unlabeled samples.
    $S = \{x \in S : |w_k \cdot x| \le b_k\}$.
    Run ADGAC with $\left(S, n_k, \varepsilon_k, \frac{\delta}{8\log(1/\varepsilon)}, k^{(1)}\left(\varepsilon_k, \frac{\delta}{8\log(1/\varepsilon)}\right)\right)$ and obtain labeled dataset $W$.
Output: Return $w_s$.
```

We first prove Theorem 8, and Theorem 9 follows exactly the same proof with $\kappa = 1$ and using Theorem 5. For clearer explanation, we re-illustrate Margin-ADGAC in a form similar to that in [6] in Algorithm 4. The proof mostly follows that of [6]. We give a refined sample complexity via Rademacher complexity following the ideas in [32], and also change the proof according to the properties of ADGAC (note that we are not using independent samples by replace the sampling step with ADGAC).

To simplify notations, let $\mathsf{err}(w)$ be $\mathsf{err}(h_w(x)) = \mathsf{err}(\mathrm{sign}(w \cdot x))$. Define $\Delta_D(w, w') = \Pr_{X \sim D}[\mathrm{sign}(w \cdot X) \ne \mathrm{sign}(w' \cdot X)]$. Also, let $\theta(w_1, w_2)$ be the angle between two vectors w_1, w_2. Let $D_{w,\gamma} = \{x : |w \cdot x| \le \gamma\}$.

The key step is to prove the following theorem:

Theorem 14. *For $k \le \log(1/\varepsilon)$, if $\Delta_{\mathcal{P}_{\mathcal{X}}}(w_{k-1}, w^*) \le M^{-(k-1)}$, with probability $1 - \frac{\delta}{k+k^2}$, after round k of Margin-ADGAC we have $\Delta_{D_{w_{k-1}, b_{k-1}}}(w_k, w^*) \le \kappa$.*

To prove the theorem, we first list useful properties of isotropic log-concave distributions and fix the parameters we use for the algorithm. We use exactly the same parameters for r_i, τ_i, b_i, z_i as in [6], and we restate them here for completeness.

Lemma 15 ([6, 10, 25]). *Suppose $X \sim \mathcal{P}_{\mathcal{X}}$ is a isotropic log-concave distribution in $\mathbb{R}^d$ with probability density function f. Then*

(a) There is an absolute constant c_1 such that, if $d = 1$, $f(x) > c_1$ for all $x \in [-1/9, 1/9]$.

(b) There is an absolute constant c_2 such that for any two unit vectors u and v in R^d we have $c_2\theta(u, v) \le \Delta_{\mathcal{P}_{\mathcal{X}}}(u, v)$.

(c) There exists constant c_3 such that for any unit vector w and $\gamma > 0$, $\Pr[|w \cdot X| \le \gamma] \le c_3\gamma$.

(d) There is a constant c_4 such that for any unit vector u, all $0 < \gamma < 1$, for all a such that $\|u - a\|_2 \le \gamma$ and $\|a\|_2 \le 1$, $\mathbb{E}_{X \sim D_{u,\gamma}}[(a \cdot X)^2] \le c_4(r^2 + \gamma^2)$.

(e) For any $c_5 > 0$, there is a constant $c_6 > 0$ such that the following holds: let u and v be two unit vectors in $\mathbb{R}^d$, and assume that $\theta(u, v) = \alpha < \pi/2$. Then $\Pr_{X \sim \mathcal{P}_{\mathcal{X}}}[sign(u \cdot X) \ne sign(v \cdot X)$ and $|v \cdot X| \le c_6\alpha] \le c_5\alpha$.

Now we give the settings of parameters. Let $M = \max\{\frac{2}{c_2\pi}, 2\}$. Let c_1' be the value of c_6 in Lemma 15 corresponding to the case where c_5 is $\frac{c_2}{4M}$; let $b_k = c_1' M^{-k}$. Let $r_k = \min\{M^{-(k-1)}/c_2, \pi/2\}$ and $\kappa = \frac{1}{4c_1' M}$. Let $\tau_k = \frac{c_1 \min\{b_{k-1}, 1/9\}\kappa}{6}$, and $z_k^2 = r_k^2 + b_{k-1}^2$. Let $\varepsilon_k = \frac{c_3\tau_k^2 b_k \kappa^2}{256 c_4 z_k^2}$, and $n_k = O\left(\frac{1}{b_k} d \log^3\left(\frac{dk}{1/\delta}\right)\right)$. Also let $m_k = 2c_3 b_k n_k + \log(12k/\delta)$.

Then we prove the following lemma:

Lemma 16. *Suppose $|W| \geq m_k$. Let $c(W)$ be the set with truthful labels w.r.t. w^*, i.e., $c(W) = \{(x, sign(w^* \cdot x)) : x \in W\}$. For any $w \in B(w_{k-1}, r_k)$, with probability $1 - \frac{\delta}{3(k+k^2)}$ we have*

$$|l(w, W) - l(w, c(W))| \leq \kappa/8.$$

Proof. Let $N = \{(x, \hat{y}) \in W : \hat{y} \neq \text{sign}(w^* \cdot x)\}$ be the set where ADGAC has x's label different than sign$(w^* \cdot x)$ (remind that $\hat{y}$ is the prediction of ADGAC). We have

$$\begin{aligned} l(w, W) &= \frac{1}{|W|} \sum_{(x,\hat{y}) \in W} l_{\tau_k}(w, x, \hat{y}) \\ &= \frac{1}{|W|} \left(\sum_{(x,y) \notin N} l_{\tau_k}(w, x, \text{sign}(w^* \cdot x)) + \sum_{(x,y) \in N} l_{\tau_k}(w, x, -\text{sign}(w^* \cdot x)) \right). \end{aligned}$$

So

$$\begin{aligned} |l(w, W) - l(w, c(W))| &\leq \frac{1}{\tau_k |W|} \sum_{x \in N} 2(w \cdot x) \\ &\leq \frac{1}{\tau_k |W|} \sum_{x \in W} I(x \in N) 2(w \cdot x). \end{aligned} \tag{5}$$

We use the following lemma from [6]:

Lemma 17 (Lemma D.4, [6])**.** *For an absolute constant c, with probability* $1 - \frac{\delta}{6(k+k^2)}$,

$$\max_{x \in W} \|x\|_2 \leq c\sqrt{d} \log \left(\frac{|W|k}{\delta} \right).$$

Note that

$$|w \cdot x| \leq |w_{k-1} \cdot x| + |(w - w_{k-1}) \cdot x| \leq b_k + r_k \|x\|_2.$$

So with probability $1 - \frac{\delta}{6(k+k^2)}$, an event E_δ happens such that

$$\frac{|w \cdot x|}{\tau_k} \leq c' \sqrt{d} \log \left(\frac{|W|k}{\delta} \right)$$

for all $x \in W$, for some constant c'.

Notice that $\frac{|N|}{|W|} \leq \frac{\varepsilon_k n_k}{m_k}$. Let N' be a $\frac{\varepsilon_k n_k}{m_k}$ fraction of W with the largest values of $|w \cdot x|$. Let $\varphi(W) = \sum_{x \in N'} |w \cdot x|$. So by (5) we have $|l(w, W) - l(w, c(W))| \leq \frac{2}{\tau_k |W|} \varphi(W)$. Now we have

$$\begin{aligned} \mathbb{E}[\varphi(W)] &= \mathbb{E} \left[\sum_{x \in W} \delta(x \in N') |w \cdot x| \right] \\ &\leq \sqrt{\frac{|N'|}{|W|}} \mathbb{E} \left[\sqrt{\sum_{x \in W} (w \cdot x)^2} \right] \\ &\leq \sqrt{\frac{\varepsilon_k n_k}{m_k}} \sqrt{\mathbb{E} \left[\sum_{x \in W} (w \cdot x)^2 \right]} \\ &\leq \sqrt{\frac{\varepsilon_k n_k}{m_k}} \sqrt{c_4} z_k |W| \leq \kappa \tau_k |W| / 16. \end{aligned}$$

The first inequality is by Cauchy-Schwartz inequality; the second is by Jensen's inequality; the third inequality is by property (d) in Lemma 15; the last inequality is by the value of ε_i. If we condition $\mathcal{P}_\mathcal{X}$ on E_δ, the above expectation will be smaller since we bound $|w \cdot x|$ from above. Now by Mcdiarmid's inequality, $\frac{1}{|W|}\varphi(W)$ deviates by at most $\frac{c'\sqrt{d}\log(\frac{|W|k}{\delta})}{|W|}$ when we change a single value of $w \cdot x$ for some $x \in W$. So by McDiarmid's inequality, using $|W| \geq m_k = \Omega(d\log^2(d/\delta))$, with probability $1 - \frac{\delta}{3(k+k^2)}$ we have

$$|l(w, W) - l(w, c(W))| \leq \mathbb{E}[\varphi(W)|E_\delta] + \kappa/16 \leq \kappa/8.$$

□

The other lemma is about bounding the difference between $l(w, c(W))$ and $E_W[l(w, c(W))]$. We improve the results in [6] using Rademacher complexity as below.

Lemma 18. *With probability* $1 - \frac{\delta}{6(k+k^2)}$ *we have*

$$|\mathbb{E}_W[l(w, x, \textit{sign}(w^* \cdot x))] - l(w, W)| \leq \kappa/16.$$

Proof. Note that every $x \in W$ is sampled independently from $D_{w_k,b_{k-1}}$. Following the same proof as in Lemma 16, an Event E_δ happens with probability $1 - \frac{\delta}{6(k+k^2)}$ that

$$\left|\frac{w \cdot x}{\tau_k}\right| \leq c'\sqrt{d}\log\left(\frac{|W|k}{\delta}\right)$$

for all $x \in W$, for some constant c'. This means $l_{\tau_k}(w, x, \text{sign}(w^* \cdot x))$ are also bounded in the same range under E_δ.

Define the function class $\mathcal{F} = \{x \to l_{\tau_k}(w, x, \text{sign}(w^* \cdot x)), \|w - w_k\| \leq r_k\}$. On event E_δ, all functions in $\mathcal{F}$ are bounded. Now we bound the Rademacher complexity $R_n(\mathcal{F})$. Actually, define $\mathcal{F}' = \{x \to \frac{1}{\tau_k} w \cdot x \cdot \text{sign}(w^* \cdot x), \|w - w_k\| \leq r_k\}$, we have $R_n(\mathcal{F}) \leq R_n(\mathcal{F}')$ by contraction inequality of Rademacher complexity (since hinge loss is 1-Lipschitz). So

$$\begin{aligned}
R_n(\mathcal{F}) &\leq R_n(\mathcal{F}') \\
&= \frac{1}{\tau_k n} E_{x_1,\ldots,x_n \sim D_{w_k,b_{k-1}}} E_{\sigma_1,\ldots,\sigma_n} \sup_{w:\|w-w_k\|\leq r_k} \sum_{i=1}^n \sigma_i w \cdot x_i \cdot \text{sign}(w^* \cdot x_i) \\
&= \frac{1}{\tau_k n} E_{x_1,\ldots,x_n \sim D_{w_k,b_{k-1}}} E_{\sigma_1,\ldots,\sigma_n} \sup_{w:\|w-w_k\|\leq r_k} \sum_{i=1}^n \sigma_i (w \cdot x_i) \qquad (6) \\
&= \frac{1}{\tau_k n} E_{x_1,\ldots,x_n \sim D_{w_k,b_{k-1}}} E_{\sigma_1,\ldots,\sigma_n} \sum_{i=1}^n \sigma_i (w_k \cdot x_i) + \\
&\frac{1}{\tau_k n} E_{x_1,\ldots,x_n \sim D_{w_k,b_{k-1}}} E_{\sigma_1,\ldots,\sigma_n} \sup_{w:\|w-w_k\|\leq r_k} \sum_{i=1}^n \sigma_i (w - w_k) \cdot x_i \\
&= \frac{1}{\tau_k n} E_{x_1,\ldots,x_n \sim D_{w_k,b_{k-1}}} E_{\sigma_1,\ldots,\sigma_n} \sup_{w:\|w-w_k\|\leq r_k} \sum_{i=1}^n \sigma_i (w - w_k) \cdot x_i \\
&= \frac{1}{\tau_k n} E_{x_1,\ldots,x_n \sim D_{w_k,b_{k-1}}} E_{\sigma_1,\ldots,\sigma_n} \sup_{w:\|w-w_k\|\leq r_k} (w - w_k) \sum_{i=1}^n \sigma_i \cdot x_i \\
&\leq \frac{1}{\tau_k n} E_{x_1,\ldots,x_n \sim D_{w_k,b_{k-1}}} E_{\sigma_1,\ldots,\sigma_n} \sup_{w:\|w-w_k\|\leq r_k} \|w - w_k\|_2 \left\|\sum_{i=1}^n \sigma_i x_i\right\|_2 \\
&\leq \frac{2r_k}{\tau_k n} \sqrt{E_{x_1,\ldots,x_n \sim D_{w_k,b_{k-1}}} E_{\sigma_1,\ldots,\sigma_n} \left\|\sum_{i=1}^n \sigma_i x_i\right\|_2^2} \qquad (7) \\
&\leq \frac{2r_k}{\tau_k n} \sqrt{E_{x_1,\ldots,x_n \sim D_{w_k,b_{k-1}}} E_{\sigma_1,\ldots,\sigma_n} \left[\sum_{i=1}^n \|x_i\|_2^2 + \sum_{i,j} \sigma_i \sigma_j x_i \cdot x_j\right]} \qquad (8)
\end{aligned}$$

$$\leq \mathcal{O}\left(\frac{1}{n}\cdot\sqrt{nd\log^2\left(\frac{nk}{\delta}\right)}\right) \tag{9}$$
$$= \mathcal{O}\left(\sqrt{\frac{d\log^2\left(\frac{nk}{\delta}\right)}{n}}\right).$$

(6) is by the property that $\sigma_i \cdot \text{sign}(w^* \cdot x_i)$ has the same distribution as σ_i, and thus we can substitute $\sigma_i \cdot \text{sign}(w^* \cdot x_i)$ with a single variable; (7) is by Jensen's inequality, and (9) is by the boundary condition on $\|x\|_2$. So by Rademacher's inequality we have

$$\begin{aligned}|\mathbb{E}_W[l(w,x,\text{sign}(w^*\cdot x))] - l(w,W)| &\leq R_{|W|}(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{|W|}}C\sqrt{d}\log\left(\frac{|W|k}{\delta}\right)\\ &\leq \mathcal{O}\left(\sqrt{\frac{d\log^2\left(\frac{|W|k}{\delta}\right)}{|W|}}\right) + \sqrt{\frac{\log(k/\delta)}{|W|}}C\sqrt{d}\log\left(\frac{|W|k}{\delta}\right)\\ &= \mathcal{O}\left(\sqrt{\frac{d\log(k/\delta)}{|W|}}\log\left(\frac{|W|k}{\delta}\right)\right).\end{aligned}$$

The choice of $|W| = m_i = \Omega\left(d\log^3\left(\frac{dk}{1/\delta}\right)\right)$ makes the above quantity less than $\kappa/16$. □

Now we are ready to prove Theorem 14.

Proof of Theorem 14. With a probability of $1 - \frac{\delta}{k+k^2}$, suppose the conditions in Lemma 16 and 18 holds for $w = v_k$ and $w = w^*$. We have

$$\begin{aligned}&\Delta_{D_{w_{k-1},b_{k-1}}}(w_k, w^*)\\ &= \Delta_{D_{w_{k-1},b_{k-1}}}(v_k, w^*)\\ &\leq \mathbb{E}_{x\in D_{w_{k-1},b_{k-1}}}[l(v_k, x, \text{sign}(w^*\cdot x))] && \text{(Since hinge loss upper bounds 0-1 loss)}\\ &\leq l(v_k, c(W)) + \kappa/16 && \text{(Using Lemma 18)}\\ &\leq l(v_k, W) + \kappa/8 && \text{(Using Lemma 16)}\\ &\leq l(w^*, W) + \kappa/4 && \text{(By the process of selecting } v_k\text{)}\\ &\leq l(w^*, c(W)) + \kappa/4 + \kappa/16 && \text{(Using Lemma 16)}\\ &\leq L(w^*) + \kappa/4 + \kappa/8 && \text{(Using Lemma 18)}\\ &\leq \kappa. && \text{(Using Lemma 3.7 in [6])}\end{aligned}$$

□

Now we can prove Theorem 8.

Proof of Theorem 8. By relative Chernoff bound and property (c) in Lemma 15, with probability $1 - \frac{\delta}{6(k+k^2)}$ we have $|W| \geq m_k = 2c_3 b_k n_k + \log(12k/\delta)$ in every iteration. Then the correctness of Margin-ADGAC follows the same way as in [6]. Now examine the number of queries: In each step we need to compare m_i instances, as well as fitting the minimum requirement of ADGAC. So the comparison complexity is

$$\mathbb{E}[\mathsf{SC}_{\text{comp}}] = \tilde{\mathcal{O}}\left(\log^2(1/\varepsilon)\left(d\log^4(d/\delta) + \left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\log(1/\delta)\right)\right).$$

The label complexity is again obtained by multiplying the label complexity in each iteration by $\log(1/\varepsilon)$. Note that $\frac{\varepsilon_k n_k}{m_k}$ is constant in each iteration. Therefore,

$$\mathsf{SC}_{\text{label}} = \tilde{\mathcal{O}}\left(\log(1/\varepsilon)\log(1/\delta)\left(\frac{1}{\varepsilon}\right)^{2\kappa-2}\right).$$

□

Proof of Theorem 9. The proof follows exactly the same process as that of Theorem 8 using $\kappa = 1$, and Theorem 5. □

H Proof of Lower Bounds

H.1 Proof of Theorem 10

Proof. Suppose $g(x_1) = a$ and $g(x_0) = b$ for $x_1, x_2 \in \mathcal{X}, a < b$. Let $h_1(x) = \text{sign}(g(x) - a)$ and $h_2(x) = \text{sign}(g(x) - b)$. Note that using $Z(x_1, x_2) = 0$ incurs $\nu' = 0$ for both $h^* = h_1$ and $h^* = h_2$, and thus comparison cannot distinguish between h_1 and h_2. Suppose $\mathbb{C} = \{h_1, h_2\}$. Thus, any algorithm $\mathcal{A}$ using both comparison and labeling oracles can be transformed into an algorithm $\mathcal{A}'$ that uses labeling oracle only, by making the comparison oracle always return 0. Note that $\mathsf{SC}_{\text{label}}(\mathcal{A}) = \mathsf{SC}_{\text{label}}(\mathcal{A}')$, so we only need to lower bound $\mathsf{SC}_{\text{label}}(\mathcal{A}')$. In the following, we adapt the proof in [19] to give a lower bound. The main difference is that our goal is to reach a small $\Pr[h(X) \neq h^*(X)]$, whereas in [19] the goal is a small $\mathsf{err}(h) - \mathsf{err}(h^*)$.

Let $P(x_1) = 24\varepsilon, P(x_0) = 1 - 24\varepsilon$. Consider two distributions $\mathcal{P}_1, \mathcal{P}_2$ over $\mathcal{X} \times \mathcal{Y}$ with two different Bayes function $\eta_1(), \eta_2()$. Let $\gamma = \varepsilon^{\kappa-1}$ if $\kappa > 1$, or $\gamma = \frac{1}{48}$ if $\kappa = 1$. Let $\eta_1(x_0) = \eta_2(x_0) = 1$, $\eta_1(x_1) = \frac{1}{2} + \gamma, \eta_2(x_1) = \frac{1}{2} - \gamma$. It is easy to verify both $\mathcal{P}_1$ and $\mathcal{P}_2$ satisfy Tsybakov noise condition.

Choose the groundtruth distribution to be $\mathcal{P}_1$ or $\mathcal{P}_2$ both with probability $1/2$. By the same proof as Theorem 4.3 in [19], an event happens with probability at least δ that $\hat{h}(x_1) \neq h^*(x_1)$, and thus $\Pr[\hat{h}(X) \neq h^*(X)] \geq \varepsilon$, if at most $2\lfloor \frac{1-\gamma^2}{\gamma^2} \log\left(\frac{1}{8\delta(1-2\delta)}\right) \rfloor$ labels are queried. So we prove the theorem for TNC.

The proof for adversarial noise is the same as the above proof using $\kappa = 1$. □

H.2 Proof of Theorem 11

Proof of Theorem 11. The first term in (2) follows directly from Theorem 10. For the second term, we consider the case where both labeling and comparison oracles are perfect with $\nu = \nu' = 0$. This is a special case for all Conditions 1, 2 and 3. Notice that in this case, a perfect comparison oracle can be constructed from a labeling oracle by $Z(x, x') = \text{sign}(Y(x) - Y(x')) = \text{sign}(h^*(x) - h^*(x'))$; thus, any algorithm $\mathcal{A}$ with access to both labeling and comparison oracles can be transformed into another algorithm $\mathcal{A}'$ that uses labeling oracle (by replacing the comparison oracle with one that queries labeling oracle instead). So we have

$$2\mathsf{SC}_{\text{comp}}(\mathcal{A}) + \mathsf{SC}_{\text{label}}(\mathcal{A}) = \mathsf{SC}_{\text{label}}(\mathcal{A}') = \Omega(d \log(1/\varepsilon)),$$

where $\Omega(d \log(1/\varepsilon))$ is the standard lower bound for realizable active learning (see e.g., [19]). □

H.3 Proof of Theorem 12

Define $R^B(\hat{g})$ to be the error of comparison oracle induced by $\hat{g}$, and also $\mathbb{C}_{\hat{g}} = \{h : h(x) = \text{sign}(\hat{g}(x) - t), t \in \mathbb{R}\}$. To prove Theorem 12, we first give a lower bound on the left hand side (Theorem 19) by giving a $\hat{g}$ that every $h \in \mathbb{C}_{\hat{g}}$ will have every at least $\sqrt{\nu'}$. Then we give an upper bound on it (Theorem 20) by finding a good estimator t. We find t by reducing $\Pr[\text{sign}(\hat{g}(X) - t) \neq h^*(X)]$ to the case when for every x, x' such that $\hat{g}(x) = \hat{g}(x')$ we also have $h^*(x) = h^*(x')$. We find such a good function f in this case by fixing the amount of error at each value of $\hat{g}(x)$, and carefully adjusting the noise levels.

Theorem 19. *Suppose* $\min\{\Pr[h^*(X) = 1], \Pr[h^*(X) = -1]\} \geq \sqrt{\nu'}$. *For any* g^* *such that* $g^*(X)$ *has a density function, there exists* $\hat{g}$ *which induces a comparison oracle with error* ν'*, such that for every* $h \in \mathbb{C}_{\hat{g}}$*, we have* $\Pr[h(X) \neq h^*(X)] \geq \sqrt{\nu'}$.

Proof. Consider the distribution of $g^*(X)$. Pick a consecutive interval $I = [a, b]$ with $a < 0 < b$ such that $\Pr(g^*(X) \in [0, b]) = \Pr(g^*(X) \in [a, 0]) = \sqrt{\nu'}$. Pick some integer $n \in \mathbb{N}$. Suppose the cdf and pdf of random variable $T = g^*(X)$ is $F(t)$ and $p(t)$ respectively. Define

$$\hat{g}(x) = \begin{cases} a + (b-a)\frac{F(g^*(x)) - F(a)}{\sqrt{v'}}, & \text{if } x \in [a, 0], \\ a + (b-a)\frac{F(g^*(x)) - F(0)}{\sqrt{v'}}, & \text{if } x \in (0, b], \\ g^*(x), & \text{otherwise.} \end{cases}$$

The error of the comparison oracle induced by $\hat{g}$ can be represented as

$$R^B(\hat{g}) = 2\int_{g^*(x)\in(0,b]} p(g^*(x)) \int_{g^*(x')\in[a,0)} p(g^*(x')) \cdot \delta(\hat{g}(x') > \hat{g}(x)) \, \mathrm{d}g^*(x)\mathrm{d}g^*(x')$$

Let $t = g^*(x)$ and $t' = g^*(x')$. Then $\hat{g}(x') > \hat{g}(x)$ if and only if

$$\begin{aligned} &F(t') - F(a) > F(t) - F(0), \\ \Leftrightarrow &F(t) - F(t') < \sqrt{\nu'}. \end{aligned}$$

For every $t \in [0, b]$, let $G(t)$ satisfy $F(t) - F(G(t)) = \sqrt{\nu'}$. Then

$$\begin{aligned} R^B(\hat{g}) &= 2\int_{t=0}^{b} p(t) \int_{t'=a}^{0} p(t') \cdot \delta\left(F(t) - F(t') < \sqrt{\nu'}\right) \, \mathrm{d}t\mathrm{d}t' \\ &= 2\int_{t=0}^{b} p(t) \int_{t'=a}^{G(t)} p(t') \, \mathrm{d}t\mathrm{d}t' \\ &= 2\int_{t=0}^{b} p(t)(F(G(t)) - F(a))\mathrm{d}t \\ &= 2\int_{t=0}^{b} p(t)(F(t) - F(0))\mathrm{d}t \\ &= 2\int_{t=0}^{b} p(t) \int_{t'=0}^{t} p(t')\mathrm{d}t\mathrm{d}t' \\ &= 2\int_{t=0}^{b}\int_{t'=0}^{b} p(t)p(t')\delta(t' < t)\mathrm{d}t\mathrm{d}t' \\ &= \nu'. \end{aligned}$$

Now examine any function in $\mathbb{C}_{\hat{g}}$. If we pick a threshold $t \notin [a, b]$, the error is at least $\sqrt{\nu'}$ since we incur error on either $\{x : g^*(x) \in [a, 0]\}$ or $\{x : g^*(x) \in [0, b]\}$. If we pick threshold $a + (b-a)t$ for $t \in [0, 1]$, we induce an error for any $g^*(x) \in [a, 0]$ with $\frac{F(g^*(x)) - F(a)}{\sqrt{v}} > t$, and any $g^*(x) \in (0, b]$ with $\frac{F(g^*(x)) - F(0)}{\sqrt{v}} < t$. A routine calculation shows the error is always $\sqrt{\nu'}$.

□

Theorem 20. *Suppose that $\hat{g}$ induces a comparison oracle with error ν', and also distributions of $\hat{g}(X)$ and $g^*(X)$ are smooth in the sense that they both have a density function. There exists $h_t(x) := \mathit{sign}(\hat{g}(x) - t) \in \mathbb{C}_{\hat{g}}$ such that the error of $h_t(x)$ with respect to $h^*(x)$ is at most $\sqrt{\nu'}$, i.e.,*

$$\Pr[h_t(X) \neq h^*(X)] = \Pr[(\hat{g}(X) - t)g^*(X) < 0] \leq \sqrt{\nu'}.$$

We first prove the inequality:

Lemma 21. *Suppose $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$ satisfies $x_i, y_i \in \mathbb{R}, x_i, y_i \geq 0$. If $\sum_{i=1}^n \sum_{j=i}^n x_i y_j \leq t$, we have*

$$\min_{k=0,1,\ldots,n} \{x_1 + \cdots + x_k + y_{k+1} + \cdots + y_n\} \leq \sqrt{\frac{2nt}{n+1}},$$

the equality holds when $x_1 = x_2 = \cdots = x_n = y_1 = \cdots = y_n = \sqrt{\frac{2t}{n(n+1)}}$.

Proof of Lemma 21. Let $f(k) = x_1 + \cdots + x_k + y_{k+1} + \cdots + y_n$. We first prove that when the maximum of $\min_{k=0,1,\ldots,n} f(k)$ is achieved, we must have $x_i = y_i$ for all i. If not, not losing generality suppose $x_l > y_l$. Now consider $x'_i = x_i$ for all $i \neq l, l+1$, and $x'_l = y_l, x'_{l+1} = x_{l+1} + x_l - y_l$ (omit the latter step if $l = n$). Let $f'(k)$ be the function of k computed based on x' and y. By $x_l > y_l$ we have $f(l) > f(l-1)$. Notice that only $f'(l) = f(l-1) < f(l)$ is reduced and for all other $k \neq l$ we have $f(k) = f'(k)$, so the minimum remains the same. Now we have

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=i}^{n} x'_i y_j &= \sum_{j=1}^{n}\sum_{i=1}^{j} x'_i y_j = \sum_{j=1}^{n} y_j \sum_{i=1}^{j} x'_i \\ &= \sum_{j=1}^{l-1} y_j \sum_{i=1}^{j} x_i + y_l \sum_{i=1}^{l} x'_i + \sum_{j=l+1}^{n} y_j \sum_{i=1}^{j} x_i \\ &\leq \sum_{j=1}^{l-1} y_j \sum_{i=1}^{j} x_i + y_l \sum_{i=1}^{l} x_i + \sum_{j=l+1}^{n} y_j \sum_{i=1}^{j} x_i \\ &\leq t.\end{aligned}$$

So there exists a configuration that maximizes $\min_k f(k)$ with $x_i = y_i$ for all i. Now suppose $x_i = y_i$ for all i. The constraint becomes

$$\sum_{i=1}^{n}\sum_{j=i}^{n} x_i x_j \leq \varepsilon,$$

which is equivalent to

$$\left(\sum_{i=1}^{n} x_i\right)^2 + \sum_{i=1}^{n} x_i^2 \leq 2\varepsilon.$$

By Cauchy-Schwarz inequality we have

$$\sum_{i=1}^{n} x_i^2 \geq \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}.$$

So

$$x_1 + \cdots + x_k + y_{k+1} + \cdots + y_n = \sum_{i=1}^{n} x_i \leq \sqrt{\frac{2nt}{n+1}}.$$

It is easy to verify the equality condition. □

Proof of Theorem 20. Not losing generality, suppose $\hat{g}(x) \in [0, 1]$; such a assumption is justifiable since any increasing transformation of $\hat{g}$ does not change $R^B(\hat{g})$. So we only need to consider $\mathbb{C}_{\hat{g}} = \{h : h(x) = h_t(x) = \text{sign}(\hat{g}(x) - t), t \in [0, 1]\}$. Let $q(u)$ denote the distribution of $\hat{g}(X)$. Let $\xi(u) = q(u)\Pr(h^*(X) = 1|\hat{g}(x) = u)$. So we have

$$\int_0^t \xi(u)du = \Pr(h^*(X) = 1, \hat{g}(X) < t).$$

So the error of h_t with respect to h^* can be expressed as

$$\begin{aligned}\Pr((\hat{g}(X) - t)g^*(X) < 0) &= \Pr(\hat{g}(X) > t, g^*(X) < 0) + \Pr(\hat{g}(X) < t, g^*(X) > 0) \\ &= \int_0^t \xi(u)du + \int_t^1 (q(u) - \xi(u))du.\end{aligned}$$

On the other hand, the comparison error can be expressed as

$$\begin{aligned}R^B(\hat{g}) &= 2\Pr(\hat{g}(X) > \hat{g}(X'), h^*(X) = -1, h^*(X) = 1) \\ &= \int_0^1 \xi(u) \int_u^1 (q(v) - \xi(v))dudv.\end{aligned}$$

Now consider we do this on the grid with step size $1/n$ and let $n \to \infty$; the integral will be the limit value. So, let

$$S_n = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=i}^{n} \xi(i/n)(q(j/n) - \xi(j/n)).$$

So

$$\Pr(\hat{g}(X) > g(X'), h^*(X) = -1, h^*(X) = 1) = \lim_{n \to \infty} S_n.$$

Also, let

$$T_n^t = \frac{1}{n} \left(\sum_{i: i/n < t} \xi(i/n) + \sum_{i: i/n >= t} (q(i/n) - \xi(i/n)) \right),$$

so

$$\Pr((\hat{g}(X) - t)g(X) < 0) = \lim_{n \to \infty} T_n^t.$$

Now let $x_i = \frac{1}{n}\xi(i/n), y_i = \frac{1}{n}(q(i/n) - \xi(i/n))$ in Lemma 21, and we have

$$\min_t T_n^t \leq \sqrt{\frac{2nS_n}{n+1}}.$$

Note that $\lim_{n \to \infty} 2S_n = R^B(\hat{g}) \leq \nu'$ and let $n \to \infty$ on both side, we have

$$\min_t \Pr[(\hat{g}(X) - t)g(X) < 0] \leq \sqrt{\nu'}.$$

□