
Improved Sample Complexity for Multiclass PAC Learning

Steve Hanneke
Purdue University
steve.hanneke@gmail.com

Shay Moran
Technion and Google Research
smoran@technion.ac.il

Qian Zhang
Purdue University
zhan3761@purdue.edu

Abstract

We aim to understand the optimal PAC sample complexity in multiclass learning. While finiteness of the Daniely-Shalev-Shwartz (DS) dimension has been shown to characterize the PAC learnability of a concept class [Brokheim, Carmon, Dinur, Moran, and Yehudayoff, 2022], there exist polylog factor gaps in the leading term of the sample complexity. In this paper, we reduce the gap in terms of the dependence on the error parameter to a single log factor and also propose two possible routes towards completely resolving the optimal sample complexity, each based on a key open question we formulate: one concerning list learning with bounded list size, the other concerning a new type of shifting for multiclass concept classes. We prove that a positive answer to either of the two questions would completely resolve the optimal sample complexity up to log factors of the DS dimension.

1 Introduction

Multiclass learning refers to the problem of classifying an input feature from a set (feature space) $\mathcal{X}$ to a label in a set (label space) $\mathcal{Y}$ with $|\mathcal{Y}| > 2$ ($\mathcal{Y}$ can be infinite) [Natarajan, 1989, Ben-David et al., 1995, Daniely and Shalev-Shwartz, 2014, Brokheim et al., 2022]. When $|\mathcal{Y}| = 2$, the problem is known as binary classification. Multiclass learning has wide applications to various tasks in machine learning including image classification [Rawat and Wang, 2017], natural language processing [Kowsari et al., 2019], tissue classification [Li et al., 2004], etc. For theoretical analysis of multiclass learning, a probabilistic setting is typically assumed, where all the feature-label pairs in the training sequence are assumed to be independent and identically distributed (iid) samples from some distribution P over $\mathcal{X} \times \mathcal{Y}$. Then, the objective of the learner is to minimize the error rate of the output classifier under the distribution P . A basic framework in the probabilistic setting is Probably Approximately Correct (PAC) learning [Valiant, 1984]. Though the characterization of PAC learnability of a binary concept class with the finiteness of its Vapnik-Chervonenkis (VC) dimension has been proved by Vapnik and Chervonenkis [1968], the characterization of the multiclass PAC learnability remained open until Brokheim et al. [2022] showed the equivalence between the PAC learnability of a concept class and the finiteness of its Daniely-Shalev-Shwartz (DS) dimension (dim, see Definition 1.4) [Daniely and Shalev-Shwartz, 2014] instead of Natarajan dimension or graph dimension.

However, the problem of establishing the optimal sample complexity or error rate (see Section 1.1 for formal definitions) for multiclass learning remains unsolved. For binary concept classes, Hanneke [2016] showed that the sample complexity is in $\Theta((d + \log(1/\delta))/\epsilon)$ where d is the VC dimension of the concept class. Since DS dimension and VC dimension coincide for binary concept classes, it is natural to ask if the sample complexity of a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ for $|\mathcal{Y}| > 2$ is also in

$\Theta((d + \log(1/\delta))/\varepsilon)$ where $d = \dim(\mathcal{H})$ is the DS dimension of $\mathcal{H}$. In terms of the upper bound, it asks if there exists a multiclass learner whose worst case error rate is in $O((d + \log(1/\delta))/n)$ with probability at least $1 - \delta$, where n denotes the size of the training sequence. However, on the one hand, an explicit proof of the $\Omega((d + \log(1/\delta))/\varepsilon)$ lower bound on the sample complexity is still missing in the literature. On the other hand, the current best upper bound on worst case error rate to our knowledge is $O\left(\frac{(d^{3/2} \log(d) + d \log(\log(n))) \log^2(n) + \log(1/\delta)}{n}\right)$ [Bruckhim et al., 2022], which differs from the conjectured rate by the factor $(\log(\log(n)) + \sqrt{d} \log(d)) \log^2(n)$.

In this paper, we step forward towards improved sample complexity and error rate in multiclass learning. As the concept class is fixed and the sample size increases during an online learning process, we mainly focus on improving the error rate in terms of the sample size n . Specifically, for a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim(\mathcal{H}) = d$, we prove an $\Omega((d + \log(1/\delta))/\varepsilon)$ lower bound on its sample complexity and construct a multiclass learner whose worst case error rate is $O((d^{3/2} \log(d) \log(n) + \log(1/\delta))/n)$ with probability at least $1 - \delta$, which implies that the sample complexity of $\mathcal{H}$ is $O\left(\frac{d^{3/2} \log(d) \log(d/\varepsilon) + \log(1/\delta)}{\varepsilon}\right)$. Our results greatly narrow the gap between the upper and lower bounds of the sample complexity and the error rate. The dependence of the upper bound of the error rate on the sample size has also been improved from $O(\log(\log(n)) \log^2(n)/n)$ to $O(\log(n)/n)$ (treating d as a constant). The multiclass learner we construct builds upon a list learner which predicts a list of labels for the test point (see Section 2.1 for a detailed introduction to list learning). Actually, we prove a reduction from multiclass learning to list learning and upper bound the error rate of the constructed multiclass learner with some function of the list size and the expected error rate of the list learner (the probability of excluding the true label in the predicted list). Moreover, the upper bound indicates that a list learner with size independent of n and expected error rate scaling linearly in $1/n$ in terms of the sample size n would imply an $O(1/n)$ error rate (treating d as a constant). We leave the construction of such list learners an open question.

Furthermore, we also explore an alternative combinatorial approach towards improved sample complexity in multiclass learning. For a concept class, we can define a hypergraph called the one-inclusion graph [Haussler et al., 1994] on its projection to a finite sequence of features (see Section 1.1 for definitions). Then, informally speaking, the “density” (defined through the average degree of the one-inclusion graph) of a concept class can be used to upper bound the error rate of multiclass learning [Daniely and Shalev-Shwartz, 2014, Aden-Ali et al., 2023]. Specifically, if we can upper bound the density of any concept class $\mathcal{H}$ by a multiple of its DS dimension, then the sample complexity is in $O((\dim(\mathcal{H}) + \log(1/\delta))/\varepsilon)$, which matches the lower bound we prove. Thus, a proof of the above upper bound directly leads to a $\Theta((\dim(\mathcal{H}) + \log(1/\delta))/\varepsilon)$ sample complexity for multiclass learning. When $\dim(\mathcal{H}) = 1$, we successfully prove the $\Theta(\log(1/\delta)/\varepsilon)$ sample complexity in Theorem 3.2. For general concept classes, we develop a technique named “pivot shifting” similar to the shifting operator [Haussler, 1995] on concept classes. We show that if a pivot shifting does not increase the DS dimension of a concept class, then its density is upper bounded by twice the DS dimension. We leave the impact of pivot shifting on DS dimension as another open question.

Throughout the paper, we use $\mathbb{N}$ to denote the set of positive integers. For any $n \in \mathbb{N}$, we define $[n] := \{1, \dots, n\}$. For any sets $\mathcal{X}, \mathcal{Y}$, sequence $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{X}^n$, and function $f \in \mathcal{Y}^{\mathcal{X}}$, we define the subsequence $\mathbf{x}_{-i} := (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ for $i \in [n]$ and $f|_{\mathbf{x}} := (f(x_1), \dots, f(x_n))$. The projection of a set $F \subseteq \mathcal{Y}^{\mathcal{X}}$ to $\mathbf{x}$ is defined as $F|_{\mathbf{x}} := \{f|_{\mathbf{x}} : f \in F\} \subseteq \mathcal{Y}^n$.

Outline In Section 1.1, we introduce the problem of multiclass learning and review some existing results. In Section 1.2, we summarize the key points of our theoretical results. In Section 2, we introduce list learning, present the reduction from multiclass learning to list learning, and improve the sample complexity upper bound of multiclass learning via this reduction together with a boosting technique for list learners. In Section 3, we prove the optimal sample complexity for classes of DS dimension 1, introduce the intuition and the definition of “pivot shifting”, and demonstrate its potential application to the proof of the optimal sample complexity of multiclass learning.

1.1 Multiclass learning

In this section, we formally introduce the problem of multiclass learning [Valiant, 1984]. For any distribution P over $\mathcal{X} \times \mathcal{Y}$, the error rate of a classifier $h \in \mathcal{Y}^{\mathcal{X}}$ under P is defined as

$$\text{er}_P(h) := P(\{(x, y) \in \mathcal{X} \times \mathcal{Y} : y \neq h(x)\}).$$

In this paper, we focus on **realizable** distributions: for a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, a distribution P over $\mathcal{X} \times \mathcal{Y}$ is $\mathcal{H}$ -realizable if $\inf_{h \in \mathcal{H}} \text{er}_P(h) = 0$. Let $\text{RE}(\mathcal{H})$ denote the set of $\mathcal{H}$ -realizable distributions. Besides, $((x_i, y_i))_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ is $\mathcal{H}$ -realizable if $\exists h \in \mathcal{H}$ such that $y_i = h(x_i), \forall i \in [n]$.

Definition 1.1 (Multiclass learner). A **multiclass learner** (or a learner) $\mathcal{A}$ is an algorithm which given a sequence $\mathbf{s} \in \cup_{n=0}^{\infty} (\mathcal{X} \times \mathcal{Y})^n$ and a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, outputs a classifier $\mathcal{A}(\mathbf{s}, \mathcal{H}) \in \mathcal{Y}^{\mathcal{X}}$.

Then, we can define multiclass PAC learning as follows.

Definition 1.2 (Multiclass PAC learning). For any concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, the **(PAC) sample complexity** $\mathcal{M}_{\mathcal{A}, \mathcal{H}} : (0, 1)^2 \rightarrow \mathbb{N}$ of a multiclass learner $\mathcal{A}$ is a mapping from $(\varepsilon, \delta) \in (0, 1)^2$ to the smallest positive integer such that for any $m \geq \mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta)$ and any distribution $P \in \text{RE}(\mathcal{H})$, $\mathbb{P}_{S \sim P^m}(\text{er}_P(\mathcal{A}(S, \mathcal{H})) > \varepsilon) \leq \delta$, and we define $\mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta) = \infty$ if no such integer exists. We say $\mathcal{H}$ is **PAC learnable** by $\mathcal{A}$ if $\mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta) < \infty$ for all $(\varepsilon, \delta) \in (0, 1)^2$. The **(PAC) sample complexity** of $\mathcal{H}$ is defined as $\mathcal{M}_{\mathcal{H}}(\varepsilon, \delta) := \inf_{\mathcal{A}} \mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta)$ for any $(\varepsilon, \delta) \in (0, 1)^2$.

Sometimes it is easier to analyze the expected error rate

$$\varepsilon_{\mathcal{A}, \mathcal{H}, P} : \mathbb{N} \rightarrow [0, 1], \quad n \mapsto \mathbb{E}_{S \sim P^n}[\text{er}_P(\mathcal{A}(S, \mathcal{H}))] = \mathbb{P}_{(S, (X, Y)) \sim P^{n+1}}(Y \neq \mathcal{A}(S, \mathcal{H})(X))$$

for a learner $\mathcal{A}$ and distribution P over $\mathcal{X} \times \mathcal{Y}$, or transductive error rate

$$\varepsilon_{\mathcal{A}, \mathcal{H}, \text{trans}} : \mathbb{N} \rightarrow [0, 1], \quad n \mapsto \sup_{\mathbf{s} = ((x_1, h(x_1)), \dots, (x_n, h(x_n))) \in (\mathcal{X} \times \mathcal{Y})^n : h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{h(x_i) \neq \mathcal{A}(\mathbf{s}_{-i}, \mathcal{H})(x_i)}.$$

We further define $\varepsilon_{\mathcal{A}, \mathcal{H}} := \sup_{P \in \text{RE}(\mathcal{H})} \varepsilon_{\mathcal{A}, \mathcal{H}, P}$, $\varepsilon_{\mathcal{H}} := \inf_{\mathcal{A}} \varepsilon_{\mathcal{A}, \mathcal{H}}$, and $\varepsilon_{\mathcal{H}, \text{trans}} := \inf_{\mathcal{A}} \varepsilon_{\mathcal{A}, \mathcal{H}, \text{trans}}$. By a leave-one-out argument [Bruckhim et al., 2022, Fact 14], we observe that $\varepsilon_{\mathcal{A}, \mathcal{H}} \leq \varepsilon_{\mathcal{A}, \mathcal{H}, \text{trans}}$. Aden-Ali et al. [2023, Theorem 2.1] upper bounded the high probability error rate using the transductive error rate, which leads to a guarantee on PAC sample complexity. Based on their result, we prove the same upper bound up to a multiplicative constant on the high probability error rate using the expected error rate in Theorem 2.6.

Next, we define pseudo-cubes and DS dimensions of concept classes. Here, we also present their extensions to the setting of k -list learning for future reference in Section 2.1.

Definition 1.3 (Pseudo-cube and k -pseudo-cube). For any $d, k \in \mathbb{N}$, a class $H \subseteq \mathcal{Y}^d$ is called a **k -pseudo-cube** of dimension d if it is non-empty, finite, and for every $h \in H$ and $i \in [d]$, there exist at least k i -neighbors of h in H , where g is an i -neighbor of h if $g(i) \neq h(i)$ and $g(j) = h(j)$ for all $j \in [d] \setminus \{i\}$. A **pseudo-cube** of dimension d is a 1-pseudo-cube of dimension d .

Definition 1.4 (DS dimension and k -DS dimension, Charikar and Pabbaraju 2023). For any $d, k \in \mathbb{N}$, we say $\mathbf{x} \in \mathcal{X}^d$ is **k -DS shattered** by $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ if $\mathcal{H}|_{\mathbf{x}}$ contains a d -dimensional k -pseudo-cube. The **k -DS dimension** $\text{dim}_k(\mathcal{H})$ of $\mathcal{H}$ is the maximum size of a k -DS shattered sequence. We say $\mathbf{x}$ is **DS shattered** by $\mathcal{H}$ if it is 1-DS shattered by $\mathcal{H}$. The **DS dimension** $\text{dim}(\mathcal{H})$ of $\mathcal{H}$ is defined as $\text{dim}_1(\mathcal{H})$.

Now, we introduce some existing results in multiclass learning. Bruckhim et al. [2022] proved that a class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is PAC learnable if and only if $d := \text{dim}(\mathcal{H}) < \infty$, and there exists a multiclass learner $\mathcal{A}$ which for any $P \in \text{RE}(\mathcal{H})$, $\delta \in (0, 1)$, $n \in \mathbb{N}$, and $S \sim P^n$, satisfies that with probability at least $1 - \delta$,

$$\text{er}_P(\mathcal{A}(S, \mathcal{H})) = O\left(\frac{(d^{3/2} \log(d) + d \log(\log(n))) \log^2(n) + \log(1/\delta)}{n}\right), \quad (1)$$

which is also the best upper bound before this paper to our knowledge. In terms of lower bound, it follows from Charikar and Pabbaraju [2023, Theorem 6] that $\varepsilon_{\mathcal{H}}(n) = \Omega(d/n)$. Thus, the current upper and lower bounds of the expected error rate does not match. Moreover, a potentially sharp lower bound on the sample complexity $\mathcal{M}_{\mathcal{H}}$ is still missing.

The learner $\mathcal{A}$ in Bruckhim et al. [2022] relies on orienting the one-inclusion graphs defined below as a building block.

Definition 1.5 (One-inclusion graph, Haussler et al. 1994). The **one-inclusion graph (OIG)** of $H \subseteq \mathcal{Y}^n$ for $n \in \mathbb{N}$ is a hypergraph $\mathcal{G}(H) = (H, E)$ where H is the vertex-set and E denotes the edge-set defined as follows. For any $i \in [n]$ and $f : [n] \setminus \{i\} \rightarrow \mathcal{Y}$, we define the set $e_{i,f} := \{h \in H : h(j) = f(j), \forall j \in [n] \setminus \{i\}\}$. Then, the edge-set is defined as

$$E := \{(e_{i,f}, i) : i \in [n], f : [n] \setminus \{i\} \rightarrow \mathcal{Y}, e_{i,f} \neq \emptyset\}.$$

For any $(e_{i,f}, i) \in E$ and $h \in H$, we say $h \in (e_{i,f}, i)$ if $h \in e_{i,f}$ and the size of the edge is $|(e_{i,f}, i)| := |e_{i,f}|$.

Typically, we consider the one-inclusion graph of the projection of a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ to a sequence $\mathbf{x} \in \mathcal{X}^n$ with $n \in \mathbb{N}$, i.e., $\mathcal{G}(\mathcal{H}|_{\mathbf{x}})$. The “density” of $\mathcal{H}$ discussed in Section 3 is defined via the “maximal average degree” (defined below) of the hypergraph $\mathcal{G}(\mathcal{H}|_{\mathbf{x}})$.

Definition 1.6 (Degree and average degree). *For any hypergraph $G = (V, E)$ and $v \in V$, we define the **degree** of v in G to be $\deg(v; G) := |\{e \in E : v \in e, |e| \geq 2\}|$. When the underlying graph is clear in the context, we simply write $\deg(v)$ in abbreviation. If $|V| < \infty$, we can define the **average degree** and **average out-degree** of G to be*

$$\text{avgdeg}(G) := \frac{1}{|V|} \sum_{v \in V} \deg(v; G) = \frac{1}{|V|} \sum_{e \in E: |e| \geq 2} |e| \text{ and } \text{avgoutdeg}(G) := \frac{1}{|V|} \sum_{e \in E} (|e| - 1).$$

For general V , we can define the **maximal average degree** of G to be $\text{md}(G) := \sup_{U \subseteq V: |U| < \infty} \text{avgdeg}(G[U])$, where $G[U] = (U, E[U])$ denotes the induced hypergraph of G on $U \subseteq V$ with $E[U] := \{e \cap U : e \in E, e \cap U \neq \emptyset\}$.

Note that for finite graphs, the average out-degree does not depend on the choice of orientation on G . Moreover, since $|e| \geq 2$ for all $e \in E_n$, we have

$$\text{avgdeg}(G) = \frac{1}{|V|} \sum_{e \in E: |e| \geq 2} |e| \leq \frac{1}{|V|} \sum_{e \in E} 2(|e| - 1) = 2\text{avgoutdeg}(G). \quad (2)$$

Now, we can define the density of a concept class as follows.

Definition 1.7. The **density** of $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is defined as $\mu_{\mathcal{H}}(m) := \sup_{\mathbf{x} \in \mathcal{X}^m} \text{md}(\mathcal{G}(\mathcal{H}|_{\mathbf{x}}))$, $\forall m \in \mathbb{N}$.

1.2 Main results

In this section, we summarize the key points of our theoretical results. The full versions of some results are stated in Section 2 and 3. We first need the following definition to rule out trivial concept classes for which one training point suffices to achieve zero error rate under any realizable distribution.

Definition 1.8 (Nondegenerate concept class, Hanneke et al. 2023). A concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is called **nondegenerate** if there exist $h_1, h_2 \in \mathcal{H}$ and $x_0, x_1 \in \mathcal{X}$ such that $h_1(x_0) = h_2(x_0)$ and $h_1(x_1) \neq h_2(x_1)$. $\mathcal{H}$ is called **degenerate** if it is not nondegenerate.

Our main result on the multiclass PAC sample complexity is as follows.

Theorem 1.9 (Partial summary of Theorem 2.5 and 2.11). *For any nondegenerate concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim(\mathcal{H}) = d$ and any $(\varepsilon, \delta) \in (0, 1)^2$, we have*

$$\Omega((d + \log(1/\delta))/\varepsilon) \leq \mathcal{M}_{\mathcal{H}}(\varepsilon, \delta) \leq O((d^{3/2} \log(d) \log(d/\varepsilon) + \log(1/\delta))/\varepsilon). \quad (3)$$

Our upper bound follows from a reduction to list learning and an improved sample complexity for list learning summarized below.

Theorem 1.10 (Informal summary of Theorem 2.7 and 2.10). *Assume that there exists a list learner which given a concept class $\mathcal{H}$ with $\dim(\mathcal{H}) = d$ and training sequence of size $n \in \mathbb{N}$ outputs a menu of size $p(\mathcal{H}, n)$ with expected error rate upper bounded by $\beta(\mathcal{H}, n)/n$ for some functions p and β nondecreasing in n . Then, there exists a multiclass learner whose error rate is*

$$O((\beta(\mathcal{H}, n) + d \log(p(\mathcal{H}, n)) + \log(1/\delta))/n) \text{ with probability at least } 1 - \delta.$$

Moreover, there exists a list learner satisfying $p(\mathcal{H}, n) = O((e\sqrt{d})^{\sqrt{d}} \log(n))$ and $\beta(\mathcal{H}, n) = O(d^{3/2} \log(d) \log(n))$.

We refer readers to Section 2.1 for detailed definitions regarding list learning. Note that if $p(\mathcal{H}, n)$ and $\beta(\mathcal{H}, n)$ of some list learner is independent of n , there exists a multiclass learner with error rate linear in $1/n$. We leave the establishment of such list learners as Open Question 1.

In addition to the above approach, we propose an alternative route toward obtaining the conjectured $\Theta((d + \log(1/\delta))/\varepsilon)$ sample complexity, by directly bounding the average degrees of one-inclusion graphs. In particular, we show in Theorem 3.2 that any $\mathcal{H}$ with $\dim(\mathcal{H}) = 1$ has $\mathcal{M}_{\mathcal{H}}(\varepsilon, \delta) = \Theta(\log(1/\delta)/\varepsilon)$, which was not previously known. Moreover, we approach the general case via a new technique we call “pivot shifting”. Specifically, we obtain the following result, which relies on an assumption on such pivot shifting. The verification of this assumption is left as Open Question 2.

Proposition 1.11 (Informal summary of the results in Section 3). *Assume that for any finite concept class, there exists a pivot shifting such that the DS dimension of the concept class does not increase after the pivot shifting, then we have $\mathcal{M}_{\mathcal{H}}(\varepsilon, \delta) = \Theta((\dim(\mathcal{H}) + \log(1/\delta))/\varepsilon)$ for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$.*

2 Multiclass learning via list learning

In this section, we prove a reduction from multiclass learning to list learning in Section 2.2. We improve the existing list learners using boosting in Section 2.3. Then, using a boosted list learner and the reduction, we improve the multiclass learning sample complexity upper bound in Section 2.4. We first present some definitions and results of list learning in Section 2.1.

2.1 List learning

In list learning, the menus defined below serve as classifiers in multiclass learning.

Definition 2.1 (k -menu, [Bruckhim et al. 2022](#)). A **menu** of size $k \in \mathbb{N}$ is a function $\mu : \mathcal{X} \rightarrow \{Y \subseteq \mathcal{Y} : |Y| \leq k\}$. A 1-menu can be viewed as a classifier in $\mathcal{Y}^{\mathcal{X}}$, and vice versa.

For any distribution P over $\mathcal{X} \times \mathcal{Y}$, the error rate of a k -menu μ under P is defined as $\text{er}_P(\mu) := P(\{(x, y) \in \mathcal{X} \times \mathcal{Y} : y \notin \mu(x)\})$ which agrees with the definition of the error rate of classifiers when the size of the menu is 1.

Definition 2.2 (k -list learner). A **list learner** $\mathcal{A}$ of size $k \in \mathbb{N}$ is an algorithm which given a sequence $\mathbf{s} \in \cup_{n=0}^{\infty} (\mathcal{X} \times \mathcal{Y})^n$ and a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, outputs a k -menu $\mathcal{A}(\mathbf{s}, \mathcal{H})$. A 1-list learner can be viewed as a multiclass learner, and vice versa.

Similar to multiclass learners, the expected error rate of a list learner $\mathcal{A}$ is defined as

$$\varepsilon_{\mathcal{A}, \mathcal{H}, P} : \mathbb{N} \rightarrow [0, 1], \quad n \mapsto \mathbb{E}_{S \sim P^n} [\text{er}_P(\mathcal{A}(S, \mathcal{H}))] = \mathbb{P}_{(S, (X, Y)) \sim P^{n+1}} (Y \notin \mathcal{A}(S, \mathcal{H})(X))$$

for any concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and distribution P over $\mathcal{X} \times \mathcal{Y}$. Restricting to realizable distributions, we can define

$$\varepsilon_{\mathcal{A}, \mathcal{H}} := \sup_{P \in \text{RE}(\mathcal{H})} \varepsilon_{\mathcal{A}, \mathcal{H}, P} \quad \text{and} \quad \varepsilon_{\mathcal{H}}^k := \inf_{k\text{-list learners } \mathcal{A}} \varepsilon_{\mathcal{A}, \mathcal{H}}.$$

Next, we define list PAC learning.

Definition 2.3 (List PAC learning, [Charikar and Pabbaraju 2023](#)). For any concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and $k \in \mathbb{N}$, the **(PAC) sample complexity** $\mathcal{M}_{\mathcal{A}, \mathcal{H}} : (0, 1)^2 \rightarrow \mathbb{N}$ of a k -list learner $\mathcal{A}$ is a mapping from $(\varepsilon, \delta) \in (0, 1)^2$ to the smallest positive integer such that for every $m \geq \mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta)$ and every distribution $P \in \text{RE}(\mathcal{H})$, $\mathbb{P}_{S \sim P^m} (\text{er}_P(\mathcal{A}(S, \mathcal{H})) > \varepsilon) \leq \delta$, and we define $\mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta) = \infty$ if no such integer exists. We say $\mathcal{H}$ is **k -list PAC learnable** by $\mathcal{A}$ if $\mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta) < \infty$ for all $(\varepsilon, \delta) \in (0, 1)^2$. The **k -list (PAC) sample complexity** of $\mathcal{H}$ is defined as $\mathcal{M}_{\mathcal{H}}^k(\varepsilon, \delta) := \inf_{k\text{-list learner } \mathcal{A}} \mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta)$ for any $(\varepsilon, \delta) \in (0, 1)^2$.

Note that $\mathcal{H}$ is PAC learnable by a learner $\mathcal{A}$ if and only if $\mathcal{H}$ is 1-list learnable by $\mathcal{A}$, and the PAC sample complexity of $\mathcal{H}$ is $\mathcal{M}_{\mathcal{H}} = \mathcal{M}_{\mathcal{H}}^1$. For list PAC learning, it was proved by [Charikar and Pabbaraju 2023](#) that a concept class $\mathcal{H}$ is k -list learnable if and only if $d_k := \dim_k(\mathcal{H}) < \infty$, and there exists a k -list learner $\mathcal{A}^k$ which for any $P \in \text{RE}(\mathcal{H})$, $\delta \in (0, 1)$, $n \in \mathbb{N}$, and $S \sim P^n$, satisfies that with probability at least $1 - \delta$,

$$\text{er}_P(\mathcal{A}^k(S, \mathcal{H})) = O\left(\frac{k^6 d_k (\sqrt{d_k} \log(d_k) + \log(k \log(n))) \log^2(n) + \log(1/\delta)}{n}\right). \quad (4)$$

For the expected error rate, the lower bound $\varepsilon_{\mathcal{H}}^k(n) = \Omega(d_k/(kn))$ has been proved in [Charikar and Pabbaraju 2023](#), Theorem 6]. However, a lower bound of the same order on the k -list PAC sample complexity is still missing in the literature. To establish a lower bound, we also need to rule out trivial classes in list learning. In analogy to Definition 1.8, we define k -nondegeneracy as follows.

Definition 2.4 (k -nondegenerate concept class). A concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is called **k -nondegenerate** for $k \in \mathbb{N}$ if there exist $h_1, \dots, h_{k+1} \in \mathcal{H}$ and $x_0, x_1 \in \mathcal{X}$ such that $|\{h_j(x_0) : j \in [k+1]\}| = 1$ and $|\{h_j(x_1) : j \in [k+1]\}| = k+1$. $\mathcal{H}$ is called **k -degenerate** if it is not k -nondegenerate.

We claim that only one training point is sufficient for the k -list learning of a k -degenerate concept class $\mathcal{H}$. Indeed, $\mathcal{H}$ is k -list learnable if $|\mathcal{H}| \leq k$. Now, suppose that $|\mathcal{H}| \geq k+1$ and is k -degenerate. Upon observing any point $(x', y') \in \mathcal{X} \times \mathcal{Y}$ realizable by $\mathcal{H}$, if $|\{h \in \mathcal{H} : h(x') = y'\}| \leq k$, then $x \mapsto \{h(x) : h \in \mathcal{H}, h(x') = y'\}$ is a k -menu which always contains the correct label. If $|\{h \in \mathcal{H} : h(x') = y'\}| \geq k+1$, then, for any $x \in \mathcal{X} \setminus \{x'\}$, we must have $|\{h(x) : h \in \mathcal{H}, h(x') = y'\}| \leq k$

because otherwise $\mathcal{H}$ is k -nondegenerate. Then, $x \mapsto \{h(x) : h \in \mathcal{H}, h(x') = y'\}$ is a k -list which always contains the correct label.

Now, we are ready to present the following lower bound on the k -list PAC sample complexity.

Theorem 2.5. *For any $k \in \mathbb{N}$, k -nondegenerate concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim_k(\mathcal{H}) = d_k \in \mathbb{N}$, $\varepsilon \in \left(0, \frac{1}{8(k+1)}\right)$, and $\delta \in \left(0, \frac{1}{4(k+1)}\right)$, we have $\mathcal{M}_{\mathcal{H}}^k(\varepsilon, \delta) \geq \frac{(d_k-1)\log(2)+4\log(1/\delta)}{16(k+1)\varepsilon}$. In particular, when $k = 1$, for any $\varepsilon \in (0, 1/16)$ and $\delta \in (0, 1/8)$, we have*

$$\mathcal{M}_{\mathcal{H}}(\varepsilon, \delta) \geq \frac{(\dim(\mathcal{H})-1)\log(2)+4\log(1/\delta)}{32\varepsilon}. \quad (5)$$

The proof of Theorem 2.5 is presented in Appendix A where we construct hard distributions based on properties of k -pseudo-cubes.

2.2 Reduction from multiclass learning to list learning

We first introduce the theorem that provides a guarantee on PAC sample complexity based on expected error rate, which will be used frequently in our analysis.

Theorem 2.6. *Fix a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and consider a learner $\mathcal{A}$ which satisfies $\varepsilon_{\mathcal{A}, \mathcal{H}, P}(n) \leq M_n/n$ for any $n \in \mathbb{N}$ and $P \in \text{RE}(\mathcal{H})$ with M_n nondecreasing in the sample size n . Then, there exists a learner $\mathcal{A}'$ such that for any $P \in \text{RE}(\mathcal{H})$, $\delta \in (0, 1)$, $n \geq 4$, and the training sequence $S \sim P^n$, with probability at least $1 - \delta$, we have*

$$\text{er}_P(\mathcal{A}'(S, \mathcal{H})) \leq 4.82 \cdot (8.34M_{\lfloor n/2 \rfloor} + \log(2/\delta))/n.$$

The proof of Theorem 2.6 is provided in Appendix B. Now, we consider general list learners whose sizes may depend on the sample size. The theorem below states our reduction to list learning.

Theorem 2.7. *Assume that there exists a list learner $\mathcal{A}_{\text{list}}$ which for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, $\mathcal{D} \in \text{RE}(\mathcal{H})$, $n \in \mathbb{N}$, and $S \sim \mathcal{D}^n$, outputs a menu $\mathcal{A}_{\text{list}}(S, \mathcal{H})$ of size $p(\mathcal{H}, n)$ satisfying $\varepsilon_{\mathcal{A}_{\text{list}}, \mathcal{H}, \mathcal{D}} \leq \beta(\mathcal{H}, n)/n$ for some function $\beta : \mathcal{Y}^{\mathcal{X}} \times \mathbb{N} \rightarrow [0, \infty)$. Without loss of generality, we assume that $p(\mathcal{H}, n)$ and $\beta(\mathcal{H}, n)$ are nondecreasing in n . Then, there exist multiclass learners $\mathcal{A}_{\text{red}}$ (see Algorithm 1) and $\mathcal{A}'_{\text{red}}$ which for any concept class $\mathcal{H}$ of DS dimension d , $\mathcal{D} \in \text{RE}(\mathcal{H})$, $\delta \in (0, 1)$, and $n \geq 4$, satisfy*

$$\varepsilon_{\mathcal{A}_{\text{red}}, \mathcal{H}, \mathcal{D}}(n) = O((\beta(\mathcal{H}, n_1) + d \log p(\mathcal{H}, n_1))/n)$$

where $n_1 := n - 2\lfloor n/3 \rfloor$, and for $S \sim \mathcal{D}^n$, with probability at least $1 - \delta$,

$$\text{er}_{\mathcal{D}}(\mathcal{A}'_{\text{red}}(S, \mathcal{H})) = O((\beta(\mathcal{H}, n_1) + d \log p(\mathcal{H}, n_1) + \log(1/\delta))/n). \quad (6)$$

The proof of Theorem 2.7 is presented in Appendix C. Note that the order of the error rate upper bound of the constructed multiclass learner is not smaller than that of the original list learner in the above theorem. Thus, the list learner $\mathcal{A}^k$ of size $k \in \mathbb{N}$ developed in Charikar and Pabbaraju [2023] cannot lead to an improved error rate of multiclass learning using our current result. The construction of $\mathcal{A}_{\text{red}}$ from $\mathcal{A}_{\text{list}}$ is shown in Algorithm 1.

Algorithm 1: Multiclass learner $\mathcal{A}_{\text{red}}$ using a list learner $\mathcal{A}_{\text{list}}$

Input: List learner $\mathcal{A}_{\text{list}}$, concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, training sequence

$S = ((x_1, y_1), \dots, (x_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n$ for $n \geq 3$, test feature $x_{n+1} \in \mathcal{X}$.

Output: A label $y \in \mathcal{Y}$ for the feature x_{n+1} .

- 1 $n_1 \leftarrow n - 2\lfloor n/3 \rfloor$, $n_2 \leftarrow \lfloor n/3 \rfloor$;
 - 2 $S^1 \leftarrow ((x_i, y_i))_{i \in [n_1]}$, $S^2 \leftarrow ((x_i, y_i))_{i=n_1+1}^n$, $\mathbf{x}' \leftarrow (x_{n_1+1}, \dots, x_n, x_{n+1})$;
 - 3 $\hat{\mu} \leftarrow \mathcal{A}_{\text{list}}(S^1, \mathcal{H})$, $N \leftarrow \sum_{(x, y) \in S^2} \mathbb{1}_{y \notin \hat{\mu}(x)}$;
 - 4 $\mathcal{H}_{\mathbf{x}'} \leftarrow \{h|_{\mathbf{x}'} : h \in \mathcal{H}, |\{i \in [n+1] \setminus [n_1] : h(x_i) \notin \hat{\mu}(x_i)\}| \leq N+1\}$;
 - 5 Sample $(I_1, \dots, I_{n_2}) \sim \text{Unif}([2n_2])^{n_2}$;
 - 6 $\hat{h} \leftarrow A_G(T, \mathcal{H}_{\mathbf{x}'})$ where $T \leftarrow ((I_j, y_{I_j+n_1}))_{j \in [n_2]}$;
 - 7 **return** the label $\hat{h}(2n_2 + 1)$.
-

In step 6 of Algorithm 1, A_G is a multiclass learner defined in Proposition H.5 in the appendix. Moreover, we prove in Proposition H.5 that for any $\mathcal{D} \in \text{RE}(\mathcal{H})$, $n \in \mathbb{N}$, $\delta \in (0, 1)$, and $S \sim \mathcal{D}^n$, with probability at least $1 - \delta$, we have

$$\text{er}_{\mathcal{D}}(A_G(S, \mathcal{H})) = O((\dim_G(\mathcal{H}) + \log(1/\delta))/n)$$

where $\dim_G(\mathcal{H})$ is the graph dimension [Natarajan and Tadepalli, 1988] of $\mathcal{H}$ (see Definition H.1 in the appendix). The above bound for classes of finite graph dimensions is also novel in the literature.

We briefly comment on the analysis of Algorithm 1. We first apply the list learner to the first third of the training samples to obtain the menu $\hat{\mu}$. Then, we count the number of errors (N) made by $\hat{\mu}$ in the last two thirds samples. Then, we consider $\mathcal{H}_{\mathbf{x}'}$ which is a subset of $\mathcal{H}|_{\mathbf{x}'}$ such that the number of errors on $\mathbf{x}'$ is bounded by $N + 1$. We show in Lemma H.9 that $\dim_G(\mathcal{H}_{\mathbf{x}'})$ is well controlled. However, as we do not observe the label of the test point, we can only consider resampling from elements in S^2 as the new training sequence fed to A_G together with the concept class $\mathcal{H}_{\mathbf{x}'}$. Thus, there still exist great challenges of upper bounding the error probability for the test point that will never be sampled. We need to emphasize that the standard leave-one-out argument [Bruckhim et al., 2022, Fact 14] cannot be directly applied as the definition of N that determines $\mathcal{H}_{\mathbf{x}'}$ only depends on S^2 but not the test point (X_{n+1}, Y_{n+1}) . We tackle this challenge by proving that some permutation of the error event together with the constraint on correctness of $\hat{\mu}$ on the last two points in $\mathbf{x}'$ when leaving the last element (i.e., the test point) out is a subset of the error event when leaving the previous element (i.e., the point in S^2) out. The details of the proof is presented in Appendix C.

2.3 Sampled boosting of list learners

We now build a list learner whose invocation to Theorem 2.7 yields the upper bound in Theorem 1.9. Bruckhim et al. [2022, Lemma 39] proposed a list sample compression scheme of size $r = O(d^{3/2} \log(n))$ for concept classes of DS dimension d and sample size n . One can show that its error rate is $O((r \log(n/r) + \log(1/\delta))/n)$ using standard techniques for sample compression schemes [David et al., 2016], which however brings the extra log factor $\log(n/r)$. Recently, da Cunha et al. [2024] proposed stable randomized sample compression schemes for binary classification whose generalization does not induce the extra log factor in n and used this framework to analyze a subsampling-based boosting algorithm for weak learners. Motivated by its success, we extend their boosting algorithm [da Cunha et al., 2024, Algorithm 1] for multiclass list learners in Algorithm 2. Before presenting the algorithm, we first need to define the majority vote of menus. For $K \in \mathbb{N}$ menus $\mu_1, \dots, \mu_K$ each of size p , we define their majority vote to be $\mu = \text{Maj}(\mu_1, \dots, \mu_K)$ with

$$\mu(x) = \text{Maj}(\mu_1, \dots, \mu_K)(x) := \{y \in \mathcal{Y} : |\{k \in [K] : y \in \mu_k(x)\}| > K/2\}, \forall x \in \mathcal{X}.$$

Note that μ has size $2p - 1$. For $p = 1$, the above definition recovers the majority vote of classifiers.

Algorithm 2: Sampled boosting $\mathcal{A}_{\text{boost}}$ of a list learner $\mathcal{A}_{\text{list}}$

Input: List learner $\mathcal{A}_{\text{list}}$, concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, training sequence

$$S = \{(x_1, y_1), \dots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n, \gamma \in (0, 1/2), \nu \in (0, \gamma/18], \delta \in (0, 1).$$

Output: Menu μ .

```

1 for  $i = 1, \dots, n$  do
2    $\mathcal{D}_1(\{(x_i, y_i)\}) \leftarrow 1/n$ ;
3  $\alpha \leftarrow \frac{1}{2} \log((1 + \gamma)/(1 - \gamma))$ ,  $m \leftarrow \mathcal{M}_{\mathcal{A}_{\text{list}}, \mathcal{H}}(1/2 - \gamma, \nu)$ ,  $K \leftarrow \lceil 4 \log(n/\delta)/\gamma \rceil$ ;
4 for  $k = 1, \dots, K$  do
5   Draw  $m$  samples  $S^k \sim \mathcal{D}_k^n$ ;
6    $\mu_k \leftarrow \mathcal{A}_{\text{list}}(S^k, \mathcal{H})$ ;
7   for  $i = 1, \dots, n$  do
8      $\mathcal{D}_{k+1}(\{(x_i, y_i)\}) \leftarrow \mathcal{D}_k(\{(x_i, y_i)\}) \exp(-\alpha(2\mathbb{1}_{y_i \in \mu_k(x_i)} - 1))$ ;
9    $\mathcal{D}_{k+1} \leftarrow \mathcal{D}_{k+1} / (\sum_{i=1}^n \mathcal{D}_k(\{(x_i, y_i)\}) \exp(-\alpha(2\mathbb{1}_{y_i \in \mu_k(x_i)} - 1)))$ ;
10 return  $\mu \leftarrow \text{Maj}((\mu_k)_{k \in [K]})$ .
```

Here, γ and ν are fixed constants, enabling us to invoke weak list learners (of constant error and confidence levels) to Algorithm 2. Next, we upper bound the error rate of the boosted list learner.

Theorem 2.8. Assume that $\mathcal{A}_{\text{list}}$ is a list learner with $\mathcal{M}_{\mathcal{A}_{\text{list}}, \mathcal{H}}(1/2 - \gamma, \nu) < \infty$ for some $\gamma \in (0, 1/2)$ and $\nu \in (0, \gamma/18]$. Then, for any $\mathcal{D} \in \text{RE}(\mathcal{H})$, $n \in \mathbb{N}$, and $\delta > 0$, sampling $S \sim \mathcal{D}^n$, with probability at least $1 - \delta$, the menu μ produced by $\mathcal{A}_{\text{boost}}$ using $\mathcal{A}_{\text{list}}$ in Algorithm 2 satisfies that

$$\text{er}_{\mathcal{D}}(\mu) = O\left(\frac{\mathcal{M}_{\mathcal{A}_{\text{list}}, \mathcal{H}}(1/2 - \gamma, \nu) \log(n/\delta)}{\gamma n}\right).$$

The proof of Theorem 2.8 is a generalization of the proof of da Cunha et al. [2024, Theorem 1.1] and is presented in Appendix D together with the proofs of other results in this section. Since multiclass learners are list learners of size 1, we can also boost multiclass learners using Algorithm 2. For instance, invoking the multiclass learner in Brukhim et al. [2022, Theorem 1] to Algorithm 2 and applying Theorem 2.6, we achieve the following sample complexity in multiclass learning.

Corollary 2.9. *There exists a multiclass learner $\mathcal{A}$ with $\varepsilon_{\mathcal{A}, \mathcal{H}}(n) = O\left(\frac{d^{3/2} \log^2(d) \log(n)}{n}\right)$ and $\mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta) = O\left(\frac{d^{3/2} \log^2(d) \log(d/\varepsilon) + \log(1/\delta)}{\varepsilon}\right)$ for any $n \in \mathbb{N}$, $(\varepsilon, \delta) \in (0, 1)^2$, and $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim(\mathcal{H}) = d$.*

There is an extra $\log(d)$ factor in the above upper bound compared to that in Theorem 1.9, which explains the reason of routing through list learning with our reduction in Theorem 2.7: the list sample compression scheme in Brukhim et al. [2022] saves a $\log(d)$ factor compared to their sample compression scheme. Therefore, we invoke their list sample compression scheme as $\mathcal{A}_{\text{list}}$ to Algorithm 2 and build a list learner whose error rate depends on only one log factor in both n and d .

Theorem 2.10. *There exists a list learner $\mathcal{A}_L$ which for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim(\mathcal{H}) = d$ and sample size $n \in \mathbb{N}$ outputs a menu of size $O((e\sqrt{d})^{\sqrt{d}} \log(n))$ with $\varepsilon_{\mathcal{A}_L, \mathcal{H}}(n) = O\left(\frac{d^{3/2} \log(d) \log(n)}{n}\right)$.*

2.4 Improved upper bounds on sample complexity

Applying the list learner $\mathcal{A}_L$ in Theorem 2.10 to our reduction, Theorem 2.7 immediately implies the following result.

Theorem 2.11. *There exists a multiclass learner $\mathcal{A}_{\text{multi}}$ such that for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ of DS dimension d , $\mathcal{D} \in \text{RE}(\mathcal{H})$, $\delta \in (0, 1)$, $n \geq d + 1$, and $S \sim \mathcal{D}^n$, with probability at least $1 - \delta$, we have*

$$\text{er}_{\mathcal{D}}(\mathcal{A}_{\text{multi}}(S, \mathcal{H})) = O\left(\frac{d^{3/2} \log(d) \log(n) + \log(1/\delta)}{n}\right), \quad (7)$$

which implies that

$$\mathcal{M}_{\mathcal{A}_{\text{multi}}, \mathcal{H}}(\varepsilon, \delta) = O\left(\frac{d^{3/2} \log(d) \log(d/\varepsilon) + \log(1/\delta)}{\varepsilon}\right), \quad \forall \varepsilon, \delta \in (0, 1). \quad (8)$$

Furthermore, if there exists a list learner $\mathcal{A}_{\text{goodlist}}$ of size $f_1(d)$ and expected error rate $\varepsilon_{\mathcal{A}_{\text{goodlist}}, \mathcal{H}}(n) \leq f_2(d)/n$ for some functions $f_1 : \mathbb{N} \rightarrow \mathbb{N}$ and $f_2 : \mathbb{N} \rightarrow [0, \infty)$, then, there exists a multiclass learner $\mathcal{A}_{\text{lin}}$ such that

$$\mathcal{M}_{\mathcal{A}_{\text{lin}}, \mathcal{H}}(\varepsilon, \delta) = O((d \log(f_1(d)) + f_2(d) + \log(1/\delta))/\varepsilon), \quad \forall \varepsilon, \delta \in (0, 1). \quad (9)$$

The proof of Theorem 2.11 follows directly from Theorem 2.7 and Theorem 2.10 and is provided in Appendix E. Moreover, observing that $\dim_k(\mathcal{H}) \geq \dim_{k'}(\mathcal{H})$ for $k < k'$, our requirement on $\mathcal{A}_{\text{goodlist}}$ does not violate the lower bound in Charikar and Pabbaraju [2023, Theorem 6].

Compared to the upper bound (1) by Brukhim et al. [2022], (7) improves the dependence of the error rate on the sample size n from $O(\log(\log(n)) \log^2(n)/n)$ to $O(\log(n)/n)$, which steps further towards the goal of $O(1/n)$ expected error rate (treating the DS dimension as a constant). Combining (5) and (8), we arrive at (3) where the gap has been improved to the factor $\sqrt{d} \log(d) \log(d/\varepsilon)$. However, we are not aware of any existing list learner satisfying the requirements of $\mathcal{A}_{\text{goodlist}}$ in Theorem 2.11. Thus, we leave the construction of $\mathcal{A}_{\text{goodlist}}$ as an open question.

Open Question 1. *Does there exist a list learner such that given a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, its size is $f_1(\dim(\mathcal{H}))$ and its expected error rate is $\varepsilon_{\mathcal{A}_{\text{list}}, \mathcal{H}}(n) = f_2(\dim(\mathcal{H}))/n$ for some functions $f_1 : \mathbb{N} \rightarrow \mathbb{N}$ and $f_2 : \mathbb{N} \rightarrow [0, \infty)$?*

Ideally, we would expect a list learner with size $O(\dim(\mathcal{H}))$ and expected error rate $O(\dim(\mathcal{H})/n)$ as it immediately implies an upper bound $\mathcal{M}_{\mathcal{H}}(\varepsilon, \delta) = O((\dim(\mathcal{H}) \log(\dim(\mathcal{H})) + \log(1/\delta))/\varepsilon)$ which matches the lower bound in (5) up to the factor $\log(\dim(\mathcal{H}))$.

3 Density, DS dimension, and pivot shifting

We now introduce an alternative route toward proving the optimal sample complexity of multiclass PAC learning: bounding the density $\mu_{\mathcal{H}} : \mathbb{N} \rightarrow [0, \infty)$ (Definition 1.7) of concept classes $\mathcal{H}$. The

following proposition summarizes existing results that illustrate the role of density in multiclass learning.

Proposition 3.1 (Daniely and Shalev-Shwartz 2014, Charikar and Pabbaraju 2023, Aden-Ali et al. 2023). *For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and $n \in \mathbb{N}$, we have*

$$\mu_{\mathcal{H}}(n)/(2en) \leq \varepsilon_{\mathcal{H}} \leq \varepsilon_{\mathcal{H},\text{trans}} \leq \mu_{\mathcal{H}}(n)/n. \quad (10)$$

Assume that $\mu_{\mathcal{H}}(n) \leq f(\dim(\mathcal{H}))$ for some function $f : \mathbb{N} \rightarrow [0, \infty)$ and all $n \in \mathbb{N}$. Then, there exists a learner $\mathcal{A}$ based on orienting the one-clusion graph (Definition 1.5) of the projected concept class (see Aden-Ali et al. [2023, Appendix A] for the formal definition of the algorithm) with sample complexity $\mathcal{M}_{\mathcal{A},\mathcal{H}}(\varepsilon, \delta) = O\left(\frac{f(\dim(\mathcal{H})) + \log(1/\delta)}{\varepsilon}\right)$ for all $\varepsilon, \delta \in (0, 1)$.¹

In Proposition 3.1, the first inequality of (10) follows from Charikar and Pabbaraju [2023, Theorem 6], the last inequality of (10) follows from Daniely and Shalev-Shwartz [2014, Theorem 2], and the last paragraph follows from Aden-Ali et al. [2023, Theorem 2.2]. Thus, for sharper multiclass sample complexity, it suffices to bound the density of a concept class with some functions of its DS dimension. Furthermore, by Definition 1.7 and (2), it suffices to bound the average out-degree (Definition 1.6) of finite one-inclusion graphs. In fact, it has been conjectured that $\mu_{\mathcal{H}}(n) \leq c \cdot \dim(\mathcal{H})$ for some constant $c > 0$ [Daniely and Shalev-Shwartz, 2014] and the question remained open since then. A positive resolution of this conjecture would immediately imply that the $\mathcal{M}_{\mathcal{H}}(\varepsilon, \delta) = \Theta((\dim(\mathcal{H}) + \log(1/\delta))/n)$ by Proposition 3.1 and Theorem 2.5. It is worth mentioning that for $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$, Haussler et al. [1994] proved that $\mu_{\mathcal{H}} \leq 2\dim(\mathcal{H})$ (for binary classes, the DS dimension is the VC dimension), which also motivates the above conjecture. In this paper, we confirm the above conjecture for concept classes of DS dimension 1.

Theorem 3.2. *For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim(\mathcal{H}) = 1$, we have $\mu_{\mathcal{H}}(n) \leq 2$, $\forall n \in \mathbb{N}$. Thus, $\mathcal{M}_{\mathcal{H}}(\varepsilon, \delta) = \Theta(\log(1/\delta)/\varepsilon)$ for any positive $\varepsilon, \delta \in O(1)$ and any $\mathcal{H}$ with $\dim(\mathcal{H}) = 1$.*

The above theorem follows from the following fact we prove for one-inclusion graphs of DS dimension 1 concept classes. The proofs of Theorem 3.2 and Proposition 3.3 are presented in Appendix F.

Proposition 3.3. *For any $n \in \mathbb{N}$ and $V_n \subseteq \mathcal{Y}^n$ with $|V_n| < \infty$ and $\dim(V_n) = 1$, there exists no cycle (see Definition 3.4) in the one-inclusion graph $\mathcal{G}(V_n)$ (see Definition 1.5).*

Definition 3.4 (Cycle in finite hypergraph). A **cycle** of length $m \in \mathbb{N} \setminus \{1\}$ in a finite hypergraph $G = (V, E)$ consists of pairwise different vertices $v^0, \dots, v^{m-1} \in V$ and pairwise different edges $e^0, \dots, e^{m-1} \in E$ such that $v^j, v^{(j+1) \bmod m} \in e^j$ for all $0 \leq j \leq m-1$.

We prove Proposition 3.3 by contradiction and analyzing different cases of the cycle. However, it is hard to extend such result to classes of higher DS dimensions. For general concept classes, motivated by the proof for binary classes [Haussler et al., 1994, Lemma 2.4], we also consider proving by induction on the size of the sequence the class projects to. Though the analysis for binary classes does not apply to general concept classes, we discover that the analysis in the induction step proceeds seamlessly for some special concept classes where a common label which we call a “pivot” exists for each edge in the last dimension of size greater than 1 in its one-inclusion graph. Before summarizing this result in Lemma 3.6 below, we first introduce the definition of a “pivot” formally.

Definition 3.5 (Pivot of finite concept class). *For any $n \in \mathbb{N} \setminus \{1\}$ and $V_n \subseteq \mathcal{Y}^n$, we define*

$$\mathfrak{P}(V_n) := \cup_{y \in \mathcal{Y}} \cup_{y' \in \mathcal{Y} \setminus \{y\}} \{(y_1, \dots, y_{n-1}) \in \mathcal{Y}^{n-1} : (y_1, \dots, y_{n-1}, y), (y_1, \dots, y_{n-1}, y') \in V_n\}.$$

*Then, $a \in \mathcal{Y}$ is said to be a **pivot** of V_n if $(y_1, \dots, y_{n-1}, a) \in V_n$ for any $(y_1, \dots, y_{n-1}) \in \mathfrak{P}(V_n)$. We emphasize that when $\mathfrak{P}(V_n) = \emptyset$, every $a \in \mathcal{Y}$ is a pivot of V_n .*

Then, we can present Lemma 3.6 whose proof is provided in Appendix G.

Lemma 3.6. *Assume that for some $n \in \mathbb{N} \setminus \{1\}$, any $d \in \mathbb{N}$, any $m \in [n-1]$, and any $H \subseteq \mathcal{Y}^m$ with $\dim(H) \leq d$ and $|H| < \infty$, we have $\text{avgoutdeg}(\mathcal{G}(H)) \leq d$. Consider an arbitrary set $V_n \subseteq \mathcal{Y}^n$ such that $|V_n| < \infty$ and $\dim(V_n) \leq d$. If V_n has a pivot, then we have $\text{avgoutdeg}(\mathcal{G}(V_n)) \leq d$.*

¹In Aden-Ali et al. [2023, Section 2.4], the label space considered is finite. However, extending the compactness argument in Brukhim et al. [2022, Appendix B], we can prove that there exists an orientation of the hypergraph with its maximum out-degree upper bounded by the ceiling of the density even when the graph is infinite, which implies that the above sample complexity of the learner $\mathcal{A}$ still holds for infinite label spaces.

Though it only works for special classes, Lemma 3.6 can serve as a building block in the induction step for the proof of $\text{avgoutdeg}(\mathcal{G}(\mathcal{H})) \leq \dim(\mathcal{H})$ for finite $\mathcal{H} \subseteq \cup_{n \in \mathbb{N}} \mathcal{Y}^n$. Moreover, the base case $n = d + 1$ has been verified in Brukhim et al. [2022, Lemma 13]. Consequently, it suffices to extend the induction step for concept classes without a pivot. With Lemma 3.6, it is natural to consider modifying the concept class to create a pivot for it while at the same time preserving the DS dimension of the modified class nonincreasing. The technique used here is similar to shifting [Haussler, 1995, Brukhim et al., 2022], though we do not shift the whole edge “downwards” but only shift the last label in some vertex of the edge to a candidate pivot. The difference is necessary, as it has already been shown that the DS dimension of a concept class can increase after the standard shifting [Brukhim et al., 2022, Example 19]. Thus, we name the technique used here “pivot shifting”.

Definition 3.7 (Pivot shifting). *For any $n \in \mathbb{N} \setminus \{1\}$, $a \in \mathcal{Y}$, and $V_n \subseteq \mathcal{Y}^n$ with $|V_n| < \infty$, we define*

$$\mathfrak{P}_a(V_n) := \cup_{y \in \mathcal{Y}} \{(y_1, \dots, y_{n-1}) \in \mathcal{Y}^{n-1} : (y_1, \dots, y_{n-1}, y) \in V_n, (y_1, \dots, y_{n-1}, a) \notin V_n\}.$$

For any $\mathbf{y} = (y_1, \dots, y_{n-1}) \in \mathfrak{P}_a(V_n)$ and the edge $(e_{n,\mathbf{y}}, n)$ in $\mathcal{G}(V_n)$, we define the set

$$L_{\mathbf{y}} := \{y \in \mathcal{Y} : (y_1, \dots, y_{n-1}, y) \in (e_{n,\mathbf{y}}, n)\}.$$

*A mapping $\gamma : \mathfrak{P}_a(V_n) \rightarrow \mathcal{Y}$ is called a **pivot shifting** on V_n to a if $\gamma(\mathbf{y}) \in L_{\mathbf{y}}$ for all $\mathbf{y} \in \mathfrak{P}_a(V_n)$. Let Γ_{a,V_n} denote the set of all pivot shifting on V_n to a . For any $\gamma \in \Gamma_{a,V_n}$, we define*

$$V_n^\gamma := (V_n \setminus \{(\mathbf{y}, \gamma(\mathbf{y})) : \mathbf{y} \in \mathfrak{P}_a(V_n)\}) \cup \{(\mathbf{y}, a) : \mathbf{y} \in \mathfrak{P}_a(V_n)\};$$

i.e., $V_{n,\gamma}$ is obtained by replacing the label $\gamma(\mathbf{y})$ in $(\mathbf{y}, \gamma(\mathbf{y}))$ with a for all $\mathbf{y} \in \mathfrak{P}_a(V_n)$.

We prove that the average out-degree does not decrease after pivot shiftings in the following lemma.

Lemma 3.8. *For any $a \in \mathcal{Y}$, $V \subseteq \cup_{n=2}^\infty \mathcal{Y}^n$ with $|V| < \infty$, and $\gamma \in \Gamma_{a,V}$, we have*

$$\text{avgoutdeg}(\mathcal{G}(V^\gamma)) \geq \text{avgoutdeg}(\mathcal{G}(V)).$$

The proof is presented in Appendix G. A key observation for the proof is that by definition, only edges of sizes greater than one contribute to the average out-degree. However, we are not able to show that the DS dimension does not increase after some pivot shifting, which we leave as an open question. Thus, whether pivot shifting is applicable to upper bounding density with DS dimension remains open.

Open Question 2. *For any $d \in \mathbb{N}$ and any $V \subseteq \cup_{n=d+2}^\infty \mathcal{Y}^n$ with $|V| < \infty$ and $\dim(V) = d$, are there some $a \in \mathcal{Y}$ and $\gamma \in \Gamma_{a,V}$ such that $\dim(V^\gamma) \leq d$?*

Nevertheless, we have taken a further and specific step toward the verification of the conjecture that $\mu_{\mathcal{H}} \leq 2\dim(\mathcal{H})$: a positive resolution of the above question would lead to the conclusion that $\mu_{\mathcal{H}} \leq 2\dim(\mathcal{H})$ by Lemma 3.6, Lemma 3.8, and Brukhim et al. [2022, Lemma 13].

Acknowledgments and Disclosure of Funding

Shay Moran is a Robert J. Shillman Fellow; he acknowledges support by ISF grant 1225/20, by BSF grant 2018385, by Israel PBC-VATAT, by the Technion Center for Machine Learning and Intelligent Systems (MLIS), and by the the European Union (ERC, GENERALIZATION, 101039692). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

- I. Aden-Ali, Y. Cherapanamjeri, A. Shetty, and N. Zhivotovskiy. Optimal pac bounds without uniform convergence. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1203–1223, Los Alamitos, CA, USA, nov 2023. IEEE Computer Society. doi: 10.1109/FOCS57990.2023.00071. URL <https://doi.ieeecomputersociety.org/10.1109/FOCS57990.2023.00071>.
- S. Ben-David, N. Cesa-Bianchi, D. Haussler, and P. Long. Characterizations of learnability for classes of $\{0, \dots, n\}$ -valued functions. *Journal of Computer and System Sciences*, 50:74–86, 1995.

- S. Bendavid, N. Cesabianchi, D. Haussler, and P.M. Long. Characterizations of learnability for classes of $\{0, \dots, n\}$ -valued functions. *Journal of Computer and System Sciences*, 50(1):74–86, 1995. ISSN 0022-0000. doi: <https://doi.org/10.1006/jcss.1995.1008>. URL <https://www.sciencedirect.com/science/article/pii/S0022000085710082>.
- Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 943–955. IEEE, 2022.
- Moses Charikar and Chirag Pabbaraju. A characterization of list learnability. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*, page 1713–1726, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399135. doi: 10.1145/3564246.3585190. URL <https://doi.org/10.1145/3564246.3585190>.
- Arthur da Cunha, Kasper Green Larsen, and Martin Ritzert. Boosting, voting classifiers and randomized sample compression schemes. *arXiv preprint arXiv:2402.02976*, 2024.
- Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *Conference on Learning Theory*, pages 287–316. PMLR, 2014.
- Ofir David, Shay Moran, and Amir Yehudayoff. Supervised learning through the lens of compression. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/59f51fd6937412b7e56ded1ea2470c25-Paper.pdf.
- S. Hanneke. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016.
- Steve Hanneke, Shay Moran, and Qian Zhang. Universal rates for multiclass learning. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 5615–5681. PMLR, 2023.
- D. Haussler. Sphere packing numbers for subsets of the Boolean n -cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory A*, 69(2):217–232, 1995.
- D. Haussler, N. Littlestone, and M. Warmuth. Predicting $\{0, 1\}$ -functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.
- Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.
- Tao Li, Chengliang Zhang, and Mitsunori Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437, 2004.
- B. K. Natarajan. On learning sets and functions. *Machine Learning*, 4:67–97, 1989.
- Balas K Natarajan and Prasad Tadepalli. Two new frameworks for learning. In *Machine Learning Proceedings 1988*, pages 402–415. Elsevier, 1988.
- Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017. doi: 10.1162/neco_a_00990.
- L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.
- V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Proc. USSR Acad. Sci.*, 181(4):781–783, 1968.

A Lower bound

Before proving Theorem 2.5, we first introduce two lemmas regarding k -pseudo-cubes that will be used in the proof.

Lemma A.1. *For any positive integers k, d , any label class $\mathcal{Y}$, any k -pseudo-cube $H \subseteq \mathcal{Y}^d$ of dimension d , any $j \in [d]$, and any label $y \in \mathcal{Y}$, define $H_y^j := \{h \in H : h(j) = y\}$. Then, we have*

$$|H_y^j| \leq \frac{|H|}{k+1}.$$

Proof. When $d = 1$, the result follows trivially from the definition of k -pseudo-cubes. We prove the result for $d \geq 2$ by contradiction. Suppose on the contrary that there exist some $j \in [d]$ and $y \in \mathcal{Y}$ such that $|H_y^j| > \frac{|H|}{k+1}$. The definition of pseudo-cubes implies that $|H| \geq k+1$. Then, there exist $h, h' \in H_y^j$ with $h \neq h'$. Let $\{f_1, \dots, f_k\}$ and $\{f'_1, \dots, f'_k\}$ denote the set of j -neighbors of h and h' in H respectively. Since $h \neq h'$ and $h(j) = y = h'(j)$, there exists some $j' \in [d] \setminus \{j\}$ such that $h(j') \neq h'(j')$. It follows that $f_i(j') = h(j') \neq h'(j') = f'_i(j')$ and thus $f_i \neq f'_i$ for any $i, l \in [k]$. Then, we have

$$|\{h \in H : h(j) \neq y\}| \geq k|H_y^j| > \frac{k|H|}{k+1}$$

and

$$|H| = |\{h \in H : h(j) \neq y\}| + |H_y^j| > |H|,$$

which is a contradiction. Thus, we must have $|H_y^j| \leq \frac{|H|}{k+1}$. $\square$

Lemma A.2. *For any integer $k \geq 1$, $d \geq 2$, $n \in [d-1]$, and $1 \leq j_1 < \dots < j_n \leq d$, any label class $\mathcal{Y}$, any k -pseudo-cube $H \subseteq \mathcal{Y}^d$ of dimension d , and any hypothesis $g \in H$, define $\mathbf{J} := (j_1, \dots, j_n)$ and $\mathbf{K} = (k_1, \dots, k_{d-n})$ such that $1 \leq k_1 < \dots < k_{d-n} \leq d$ and $\{j_1, \dots, j_n, k_1, \dots, k_{d-n}\} = [d]$. Then, $H_{g,\mathbf{J}} := \{h|_{\mathbf{K}} : h \in H, h(j_i) = g(j_i), \forall i \in [n]\}$ is a k -pseudo-cube of dimension $d-n$.*

Proof. For any $f \in H_{g,\mathbf{J}}$, there exists some $h \in H$ such that $f = h|_{\mathbf{K}}$. Then, for any $i \in [d-n]$, there exists k distinct $h_1, \dots, h_k \in H$ such that $h_m(k_i) \neq h(k_i)$ and $h_m(l) = h(l)$ for all $l \in [d] \setminus \{k_i\}$ and $m \in [k]$. Since $k_i \notin \{j_1, \dots, j_n\}$, we have $h|_{\mathbf{J}} = h_m|_{\mathbf{J}} = g|_{\mathbf{J}}$ and thus $f_m := h_m|_{\mathbf{K}} \in H_{g,\mathbf{J}}$ for all $m \in [k]$. Then, we have $f(i) = h(k_i) \neq h_m(k_i) = f_m(i)$ and $f(l) = h(k_l) = h_m(k_l) = f_m(l)$ for any $l \in [d-n] \setminus \{i\}$ and $m \in [k]$, which implies that $H_{g,\mathbf{J}}$ is a k -pseudo-cube of dimension $d-n$. $\square$

Proof of Theorem 2.5. Consider an arbitrary k -nondegenerate concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ for some $k \in \mathbb{N}$. Let $x_0, x_1 \in \mathcal{X}$ and $h_1, \dots, h_{k+1} \in \mathcal{H}$ witness the k -nondegeneracy as specified in Definition 2.4. For any $\varepsilon \in (0, 1/2(k+1))$ and $\delta \in (0, 1)$, let $I \sim \text{Bernoulli}((k+1)\varepsilon)$ and $J \sim \text{Unif}([k+1])$. Then, for any $j \in [k+1]$, $(x_I, h_j(x_I))$ follows the $\mathcal{H}$ -realizable distribution $P_{\varepsilon,j}$ over $\mathcal{X} \times \mathcal{Y}$ defined by $P_{\varepsilon,j}(\{(x_0, h_j(x_0))\}) = 1 - (k+1)\varepsilon$ and $P_{\varepsilon,j}(\{(x_1, h_j(x_1))\}) = (k+1)\varepsilon$. Sample $(I_1, \dots, I_n) \sim \text{Bernoulli}(\varepsilon)^n$ independent of (I, J) and define $S = ((x_{I_1}, h_J(x_{I_1})), \dots, (x_{I_n}, h_J(x_{I_n})))$ and $S' = ((x_0, h_J(x_0)), \dots, (x_0, h_J(x_0)))$. Then, for any k -list learner $\mathcal{A}$, we have

$$\begin{aligned} & \mathbb{P}(h_J(x_I) \notin \mathcal{A}(S, \mathcal{H})(x_I) | I_1 = 0, \dots, I_n = 0) \\ & \geq \mathbb{P}(h_J(x_1) \notin \mathcal{A}(S', \mathcal{H})(x_1), I = 1) \\ & = \mathbb{P}(I = 1) \mathbb{P}(h_J(x_1) \notin \mathcal{A}(S', \mathcal{H})(x_1)) \\ & = (k+1)\varepsilon \mathbb{E}[\mathbb{P}(h_J(x_1) \notin \mathcal{A}(S', \mathcal{H})(x_1) | \mathcal{A}(S', \mathcal{H}))] \\ & \geq \varepsilon, \end{aligned}$$

where the last inequality follows from the facts that $h_J(x_0) = h_1(x_0)$, $|\{h_1(x_1), \dots, h_{k+1}(x_1)\}| = k+1$, $h_J(x_1) \sim \text{Unif}(\{h_1(x_1), \dots, h_{k+1}(x_1)\})$. Since $J \sim \text{Unif}([k+1])$, there exists $j \in [k+1]$ such that

$$\begin{aligned} & \mathbb{P}(h_j(x_I) \notin \mathcal{A}(S^j, \mathcal{H})(x_I) | I_1 = 0, \dots, I_n = 0) \\ & = \mathbb{P}(h_J(x_I) \notin \mathcal{A}(S, \mathcal{H})(x_I) | J = j, I_1 = 0, \dots, I_n = 0) \geq \varepsilon, \end{aligned}$$

where $S^j := ((x_{I_1}, h_j(x_{I_1})), \dots, (x_{I_n}, h_j(x_{I_n}))) \sim P_{\varepsilon, j}^n$ is independent of $(x_I, h_j(x_I)) \sim P_{\varepsilon, j}$. Since $\frac{1}{2(k+1)\varepsilon} \leq \frac{1-(k+1)\varepsilon}{(k+1)\varepsilon} < \frac{1}{-\log(1-(k+1)\varepsilon)}$ for $\varepsilon \in (0, 1/2(k+1))$, if $n \leq \frac{\log(1/\delta)}{2(k+1)\varepsilon} < \frac{\log(\delta)}{\log(1-(k+1)\varepsilon)}$, we have

$$\mathbb{P}(I_1 = 0, \dots, I_n = 0) = (1 - (k+1)\varepsilon)^n > \delta.$$

Then, with probability greater than δ , we have $\text{er}_{P_{\varepsilon, j}}(\mathcal{A}(S^j, \mathcal{H})) \geq \varepsilon$, which implies that

$$\mathcal{M}_{\mathcal{H}}^k(\varepsilon, \delta) \geq \frac{\log(1/\delta)}{2(k+1)\varepsilon}.$$

Next, consider an arbitrary concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that $\dim_k(\mathcal{H}) = d_k \in \mathbb{N} \setminus \{1\}$ for some $k \in \mathbb{N}$. Then, there exist a sequence $\mathbf{x} := (x_1, \dots, x_{d_k}) \in \mathcal{X}^{d_k}$ and a k -pseudo-cube $H \subseteq \mathcal{H}|_{\mathbf{x}}$ of dimension d_k . Denote the elements in H with $\mathbf{y}^1 = (y_1^1, \dots, y_{d_k}^1), \dots, \mathbf{y}^{|H|} = (y_1^{|H|}, \dots, y_{d_k}^{|H|})$. For any $\varepsilon \in (0, 1/8(k+1))$, consider the categorical distribution Q_{ε} over $[d_k]$ defined by $Q_{\varepsilon}(\{1\}) = 1 - 4(k+1)\varepsilon$ and $Q_{\varepsilon}(\{i\}) = \frac{4(k+1)\varepsilon}{d_k - 1}$ for $i \in [d_k] \setminus \{1\}$. Let $J \sim \text{Unif}([|H|])$. For any $i \in [d_k] \setminus \{1\}$ and $\mathcal{I} = (i_1, \dots, i_m) \in ([d_k] \setminus \{i\})^m$ with $m \in [d_k - 1]$ and $i_1 < \dots < i_m$, define $\mathcal{I}' := (i'_1, \dots, i'_{d_k - m})$ such that $i'_1 < \dots < i'_{d_k - m}$ and $\{i_1, \dots, i_m, i'_1, \dots, i'_{d_k - m}\} = [d_k]$. Then, we have that conditional on $(i_1, y_{i_1}^J), \dots, (i_m, y_{i_m}^J)$, $\mathbf{y}^J|_{\mathcal{I}'}$ follows the uniform distribution over the set $H_{\mathbf{y}^J, \mathcal{I}'}$ which is a k -pseudo-cube by Lemma A.2. Consequently, we can apply Lemma A.1 to conclude that

$$|(H_{\mathbf{y}^J, \mathcal{I}'}|_y)| \leq \frac{|H_{\mathbf{y}^J, \mathcal{I}'}|}{k+1}$$

for any $y \in \mathcal{Y}$ and $i' \in [d_k - m]$, which immediately implies that

$$\mathbb{P}(y_i^J \in \{v_1, \dots, v_k\} | (i_1, y_{i_1}^J), \dots, (i_m, y_{i_m}^J)) \leq \frac{k}{k+1}. \quad (11)$$

for any distinct $v_1, \dots, v_k \in \mathcal{Y}$.

Let $(I, I_1, \dots, I_n) \sim Q_{\varepsilon}^{n+1}$ for $n \in \mathbb{N}$ be independent of J . Define $S := ((x_{I_1}, y_{I_1}^J), \dots, (x_{I_n}, y_{I_n}^J))$. For any k -list learner $\mathcal{A}$, by (11), we have

$$\begin{aligned} & \mathbb{P}(y_I^J \notin \mathcal{A}(S, \mathcal{H})(x_I), I \neq 1) \\ & \geq \mathbb{P}(y_I^J \notin \mathcal{A}(S, \mathcal{H})(x_I), I \neq 1, I \neq I_1, \dots, I \neq I_n) \\ & \geq \sum_{i=2}^{d_k} \mathbb{P}(y_I^J \notin \mathcal{A}(S, \mathcal{H})(x_I), I = i, I_1 \neq i, \dots, I_n \neq i) \\ & = \sum_{i=2}^{d_k} \mathbb{E}[\mathbb{1}_{I_1 \neq i, \dots, I_n \neq i} \mathbb{P}(y_i^J \notin \mathcal{A}(S, \mathcal{H})(x_i), I = i | \mathcal{A}(S, \mathcal{H}), (I_1, y_{I_1}^J), \dots, (I_n, y_{I_n}^J))] \\ & \geq \sum_{i=2}^{d_k} \frac{\mathbb{P}(I = i)}{k+1} \mathbb{P}(I_1 \neq i, \dots, I_n \neq i) \\ & = 4\varepsilon \left(1 - \frac{4(k+1)\varepsilon}{d_k - 1}\right)^n. \end{aligned}$$

Since $J \sim \text{Unif}([|H|])$, there exists $j = j(\mathcal{A}, \mathcal{H}) \in [|H|]$ such that

$$\mathbb{P}(y_I^j \notin \mathcal{A}(S^j, \mathcal{H})(x_I), I \neq 1) = \mathbb{P}(y_I^J \notin \mathcal{A}(S, \mathcal{H})(x_I), I \neq 1 | J = j) \geq 4\varepsilon \left(1 - \frac{4(k+1)\varepsilon}{d_k - 1}\right)^n,$$

where $S^j := ((x_{I_1}, y_{I_1}^j), \dots, (x_{I_n}, y_{I_n}^j))$. Note that if we define the distribution $P_{\varepsilon, j}$ over $\mathcal{X} \times \mathcal{Y}$ by

$$\begin{aligned} P_{\varepsilon, j}(\{x_1, y_1^j\}) &= 1 - 4(k+1)\varepsilon \quad \text{and} \\ P_{\varepsilon, j}(\{x_i, y_i^j\}) &= \frac{4(k+1)\varepsilon}{d_k - 1}, \quad \forall i \in [d_k] \setminus \{1\}, \end{aligned}$$

then, we have $(S, (x_I, y_I^j)) \sim P_{\varepsilon, j}^{n+1}$. For any $n \leq \frac{(d_k-1)\log(2)}{8(k+1)\varepsilon} < \frac{\log(1/2)}{\log(1-4(k+1)\varepsilon/(d_k-1))}$, we have $\left(1 - \frac{4(k+1)\varepsilon}{d_k-1}\right)^n > \frac{1}{2}$ and

$$\mathbb{P}(y_I^j \notin \mathcal{A}(S^j, \mathcal{H})(x_I), I \neq 1) > 2\varepsilon.$$

Now, we define the algorithm $\mathcal{A}'$ by

$$\mathcal{A}'(\mathbf{s}, \mathcal{H})(x_1) := \{y_1^j\} \text{ and } \mathcal{A}'(\mathbf{s}, \mathcal{H})(x) := \mathcal{A}(\mathbf{s}, \mathcal{H})(x), \forall x \in \mathcal{X} \setminus \{x_1\}$$

for any $\mathbf{s} \in \cup_{m=0}^{\infty} (\mathcal{X} \times \mathcal{Y})^m$. Then, we have

$$\begin{aligned} \mathbb{E}[\text{er}_{P_{\varepsilon, j}}(\mathcal{A}'(S^j, \mathcal{H}))] &\geq \mathbb{P}(y_I^j \notin \mathcal{A}'(S^j, \mathcal{H})(x_I), I \neq 1) \\ &= \mathbb{P}(y_I^j \notin \mathcal{A}(S^j, \mathcal{H})(x_I), I \neq 1) \\ &> 2\varepsilon. \end{aligned} \tag{12}$$

On the other hand, the definition of $\mathcal{A}'$ yields that $y_1^j \in \mathcal{A}'(S^j, (x_1))$ and hence

$$\text{er}_{P_{\varepsilon, j}}(\mathcal{A}'(S^j, \mathcal{H})) \leq P_{\varepsilon, j}(\mathcal{X} \setminus \{x_1\}) = 4(k+1)\varepsilon.$$

Suppose the following holds

$$\mathbb{P}(\text{er}_{P_{\varepsilon, j}}(\mathcal{A}(S^j, \mathcal{H})) > \varepsilon) \leq \frac{1}{4(k+1)}. \tag{13}$$

Since $\text{er}_{P_{\varepsilon, j}}(\mathcal{A}'(S^j, \mathcal{H})) \leq \text{er}_{P_{\varepsilon, j}}(\mathcal{A}(S^j, \mathcal{H}))$, the above inequality implies that

$$\mathbb{P}(\text{er}_{P_{\varepsilon, j}}(\mathcal{A}'(S^j, \mathcal{H})) > \varepsilon) \leq \frac{1}{4(k+1)}$$

and therefore

$$\begin{aligned} \mathbb{E}[\text{er}_{P_{\varepsilon, j}}(\mathcal{A}'(S^j, \mathcal{H}))] &\leq \varepsilon + \mathbb{E}[\text{er}_{P_{\varepsilon, j}}(\mathcal{A}'(S^j, \mathcal{H})) \mathbb{1}_{\text{er}_{P_{\varepsilon, j}}(\mathcal{A}'(S^j, \mathcal{H})) > \varepsilon}] \\ &\leq \varepsilon + 4(k+1)\varepsilon \mathbb{P}(\text{er}_{P_{\varepsilon, j}}(\mathcal{A}'(S^j, \mathcal{H})) > \varepsilon) \\ &\leq 2\varepsilon, \end{aligned}$$

which contradicts (12). Thus, we can conclude that (13) is false, i.e.,

$$\mathbb{P}(\text{er}_{P_{\varepsilon, j}}(\mathcal{A}(S^j, \mathcal{H})) > \varepsilon) > \frac{1}{4(k+1)}$$

for $n \leq \frac{(d_k-1)\log(2)}{8(k+1)\varepsilon}$. For $\delta \in (0, 1/4(k+1)]$, we have $\mathbb{P}(\text{er}_{P_{\varepsilon, j}}(\mathcal{A}(S^j, \mathcal{H})) > \varepsilon) > \delta$ and thus

$$\mathcal{M}_{\mathcal{H}}^k(\varepsilon, \delta) \geq \frac{(d_k-1)\log(2)}{8(k+1)\varepsilon}.$$

In conclusion, for $\varepsilon, \delta = O(1/k)$, we have

$$\mathcal{M}_{\mathcal{H}}^k(\varepsilon, \delta) = \Omega\left(\frac{d_k + \log(1/\delta)}{k\varepsilon}\right).$$

□

B Proof of Theorem 2.6

Proof. Consider an algorithm $\mathcal{A} : (\cup_{k=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^k) \times 2^{\mathcal{Y}^{\mathcal{X}}} \rightarrow \mathcal{Y}^{\mathcal{X}}$ which for any hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, $\mathcal{H}$ -realizable distribution $\mathcal{D}$, and $n \in \mathbb{N}$, satisfies that

$$\mathbb{P}_{(T, (X, Y)) \sim \mathcal{D}^{n+1}}(\mathcal{A}(T, \mathcal{H})(X) \neq Y) \leq \frac{M_n}{n} \tag{14}$$

where M_n is nondecreasing in the sample size n .

For an arbitrary sequence $S = ((x_1, y_1), \dots, (x_n, y_n), (x_{n+1}, y_{n+1})) \in (\mathcal{X} \times \mathcal{Y})^{n+1}$, define S_{-i} to be the subsequence of S excluding (x_i, y_i) for any $i \in [n+1]$. Let $(T_m, (X, Y)) \sim \text{Unif}(S)^{m+1}$ where $m \in \mathbb{N}$, $T_m \in (\mathcal{X} \times \mathcal{Y})^m$ and $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. Moreover, for any $i \in [n+1]$, let $T_m^i \sim \text{Unif}(S_{-i})^m$. Then, for any algorithm $\mathcal{A} : \cup_{k=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^k \times 2^{\mathcal{Y}^{\mathcal{X}}} \rightarrow \mathcal{Y}^{\mathcal{X}}$ and concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, we have

$$\begin{aligned} \mathbb{P}(\mathcal{A}(T_m, \mathcal{H})(X) \neq Y) &= \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{P}(\mathcal{A}(T_m, \mathcal{H})(x_i) \neq y_i) \\ &\geq \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{E} [\mathbb{1}_{(x_i, y_i) \notin T_m} \mathbb{1}_{\mathcal{A}(T_m, \mathcal{H})(x_i) \neq y_i}] \\ &= \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{P}((x_i, y_i) \notin T_m) \mathbb{P}(\mathcal{A}(T_m, \mathcal{H})(x_i) \neq y_i | (x_i, y_i) \notin T_m) \\ &\geq \frac{n+1-m}{n+1} \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{P}(\mathcal{A}(T_m, \mathcal{H})(x_i) \neq y_i | (x_i, y_i) \notin T_m) \\ &= \frac{n+1-m}{n+1} \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{P}(\mathcal{A}(T_m^i, \mathcal{H})(x_i) \neq y_i). \end{aligned} \quad (15)$$

Note that if S is consistent with $\mathcal{H}$, then $\text{Unif}(S)$ is $\mathcal{H}$ -realizable.

Next, define the algorithm $\mathcal{A}_{\text{Maj}, m} : (\cup_{k=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^k) \times 2^{\mathcal{Y}^{\mathcal{X}}} \rightarrow \mathcal{Y}^{\mathcal{X}}$ by

$$\mathcal{A}_{\text{Maj}, m}(R, \mathcal{H}) := \text{Majority}((\mathcal{A}(\mathbf{r}, \mathcal{H}))_{\mathbf{r} \in R^m}), \quad \forall R \in (\mathcal{X} \times \mathcal{Y})^n, \forall n \in \mathbb{N}.$$

By the definition above, for any permutation R_{π} of $R \in (\mathcal{X} \times \mathcal{Y})^n$, we have

$$\mathcal{A}_{\text{Maj}, m}(R_{\pi}, \mathcal{H}) = \mathcal{A}_{\text{Maj}, m}(R, \mathcal{H}). \quad (16)$$

Moreover, for any $i \in [n+1]$, the above definition yields that

$$\begin{aligned} \mathbb{1}_{\mathcal{A}_{\text{Maj}, m}(S_{-i}, \mathcal{H})(x_i) \neq y_i} &\leq \frac{1}{n+1-m} |\{s \in S_{-i}^m : \mathcal{A}(s, \mathcal{H})(x_i) \neq y_i\}| \geq \frac{1}{2} \\ &\leq \frac{2}{n+1-m} |\{s \in S_{-i}^m : \mathcal{A}(s, \mathcal{H})(x_i) \neq y_i\}| \\ &= 2\mathbb{P}(\mathcal{A}(T_m^i, \mathcal{H})(x_i) \neq y_i). \end{aligned}$$

Then, by (15), we have

$$\frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{1}_{\mathcal{A}_{\text{Maj}, m}(S_{-i}, \mathcal{H})(x_i) \neq y_i} \leq \frac{2(n+1)}{n+1-m} \mathbb{P}(\mathcal{A}(T_m, \mathcal{H})(X) \neq Y).$$

Choosing $m = \lfloor (n+1)/2 \rfloor$ and defining $\mathcal{A}_{\text{Maj}} := \mathcal{A}_{\text{Maj}, \lfloor (n+1)/2 \rfloor}$, for any $\mathcal{H}$ -realizable sequence S , by (14) and the above results, we have

$$\sum_{i=1}^{n+1} \mathbb{1}_{\mathcal{A}_{\text{Maj}}(S_{-i}, \mathcal{H})(x_i) \neq y_i} \leq \frac{2(n+1)^2 M_{\lfloor (n+1)/2 \rfloor}}{(n+1 - \lfloor (n+1)/2 \rfloor) \lfloor (n+1)/2 \rfloor} \leq 8.34 M_{\lfloor (n+1)/2 \rfloor} \quad (17)$$

for any $n \geq 4$. (16) and (17) imply that the algorithm $\mathcal{A}_{\text{Maj}}$ satisfies Assumptions 2.1 and 2.3 in Aden-Ali et al. [2023].

Now, we can define the randomized algorithm $\mathcal{A}_{\text{Ran}}$ which given a sample $S \in (\mathcal{X} \times \mathcal{Y})^n$ with $n \in \mathbb{N}$ and a concept class $\mathcal{H}$, outputs the classifier $\mathcal{A}_{\text{Maj}}(S, \mathcal{H})$ if $n \leq 3$ and outputs a random classifier following the uniform distribution over the sequence $(\mathcal{A}_{\text{Maj}}(S_{\leq t}, \mathcal{H}))_{\lfloor n/4 \rfloor \leq t \leq 4\lfloor n/4 \rfloor - 1}$ if $n \geq 4$, i.e.,

$$\mathbb{P}(\mathcal{A}_{\text{Ran}}(S, \mathcal{H}) = \mathcal{A}_{\text{Maj}}(S_{\leq t}, \mathcal{H})) = \frac{1}{3\lfloor n/4 \rfloor}, \quad \forall t \in [\lfloor n/4 \rfloor, 4\lfloor n/4 \rfloor - 1],$$

where $S_{\leq t}$ denotes the subsequence of S consisting of the first t elements in S . Then, by Aden-Ali et al. [2023, Theorem 2.1], for any $n \geq 4$, $\mathcal{H}$ -realizable distribution $\mathcal{D}$, and confidence parameter

$\delta \in (0, 1)$, given a training sample $S \sim \mathcal{D}^n$, we have

$$\begin{aligned} \text{er}_{\mathcal{D}}(\mathcal{A}_{\text{Ran}}(S, \mathcal{H})) &= \frac{1}{3^{\lfloor n/4 \rfloor}} \sum_{t=\lfloor n/4 \rfloor}^{4\lfloor n/4 \rfloor-1} \text{er}_{\mathcal{D}}(\mathcal{A}_{\text{Maj}}(S_{\leq t}, \mathcal{H})) \\ &\leq 4.82 \left(\frac{8.34M_{\lfloor n/2 \rfloor}}{n} + \frac{1}{n} \log \left(\frac{2}{\delta} \right) \right) \end{aligned}$$

with probability at least $1 - \delta$. $\square$

C Proof of Theorem 2.7

The following proof requires several lemmas from Appendix H.

Proof of Theorem 2.7. Assume that we have access to a list learning algorithm $\mathcal{A}_{\text{list}}$ which for any concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ of DS dimension d , any $\mathcal{H}$ -realizable distribution $\mathcal{D}$, any $n \in \mathbb{N}$, and $(S, (X, Y)) \sim \mathcal{D}^{n+1}$, outputs a menu $\mathcal{A}_{\text{list}}(S, \mathcal{H})$ of size $p(n) \in \mathbb{N}$ satisfying

$$\mathbb{P}(Y \notin \mathcal{A}_{\text{list}}(S, \mathcal{H})(X)) \leq \frac{\beta(\mathcal{H}, n)}{n}$$

for some function $\beta : 2^{\mathcal{Y}^{\mathcal{X}}} \times \mathbb{N} \rightarrow [0, \infty)$.

For $n_1, n_2 \in \mathbb{N}$, let $(S^1, S^2, (X, Y)) \sim \mathcal{D}^{n_1+2n_2+1}$ where $S^1 \in (\mathcal{X} \times \mathcal{Y})^{n_1}$, $S^2 \in (\mathcal{X} \times \mathcal{Y})^{2n_2}$, and $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. Let $\hat{\mu} = \mathcal{A}_{\text{list}}(S^1, \mathcal{H})$. According to the property of $\mathcal{A}_{\text{list}}$, the size of $\hat{\mu}$ is $p(n_1)$ and we have

$$\mathbb{P}(Y \notin \hat{\mu}(X)) \leq \frac{\beta(\mathcal{H}, n_1)}{n_1}.$$

For notational convenience, define $S' := (S^2, (X, Y))$, $S'' := (S^1, S^2, (X, Y))$, and enumerate the elements of S' as

$$((X_1, Y_1), \dots, (X_{2n_2+1}, Y_{2n_2+1}))$$

where (X_{2n_2+1}, Y_{2n_2+1}) denotes (X, Y) . We now define

$$N := \sum_{i \in [2n_2]} \mathbb{1}_{Y_i \notin \hat{\mu}(X_i)}$$

and

$$\begin{aligned} \mathcal{H}_{S'} &:= \mathcal{H}_{(X_1, \dots, X_{2n_2+1}), \hat{\mu}, N} \\ &= \left\{ h|_{(X_1, \dots, X_{2n_2+1})} : h \in \mathcal{H}, |\{i \in [2n_2+1] : h(X_i) \notin \hat{\mu}(X_i)\}| \leq N+1 \right\}. \end{aligned}$$

It follows from the property of $\hat{\mu}$ and Lemma H.9 that

$$\mathbb{E}[N] = \frac{2n_2\beta(\mathcal{H}, n_1)}{n_1} \quad (18)$$

and, conditional on S' ,

$$\dim_G(\mathcal{H}_{S'}) \leq (2\log_2 e + 4)(5d\log_2(p(n_1)) + 2N + 2).$$

Sample $(\mathbf{I}, I) \sim \text{Unif}([2n_2+1]^{n_2+1})$ independent of S'' where $\mathbf{I} = (I_1, \dots, I_{n_2}) \in [2n_2+1]^{n_2}$ and $I \in [2n_2+1]$. Then, we define the sequence $T(S', \mathbf{I}) := ((I_1, Y_{I_1}), \dots, (I_{n_2}, Y_{I_{n_2}}))$ and the classifier

$$\hat{h}_{S', \mathbf{I}} := A_G(T(S', \mathbf{I}), \mathcal{H}_{S'})$$

for any $i \in [2n_2+1]$, where A_G is the algorithm specified in Proposition H.5. Since conditional on S'' , the distribution of each element in $T(S', \mathbf{I})$ is $\mathcal{H}_{S'}$ -realizable, by Corollary H.6, there exists some constant $C' > 0$ such that

$$\mathbb{P}(\hat{h}_{S', \mathbf{I}}(I) \neq Y_I | S'') \leq \frac{C' \dim_G(\mathcal{H}_{S'})}{n_2} \leq \frac{C_1 d \log(p(n_1)) + C_2 N + C_3}{n_2}. \quad (19)$$

for some constant $C_1, C_2, C_3 > 0$.

For $\mathbf{I}' = (I'_1, \dots, I'_{n_2}) \sim \text{Unif}([2n_2])^{n_2}$ independent of all other random variables, define $T^i(S', \mathbf{I}') := ((\rho_i(I'_1), Y_{\rho_i(I'_1)}), \dots, (\rho_i(I'_{n_2}), Y_{\rho_i(I'_{n_2})}))$ where $\rho_i : [2n_2] \rightarrow [2n_2 + 1] \setminus \{i\}$, $k \mapsto k \mathbb{1}_{k < i} + (k + 1) \mathbb{1}_{k \geq i}$ for all $i \in [2n_2 + 1]$. Consider the classifier

$$\tilde{h}_{S', \mathbf{I}', i} := A_G(T^i(S', \mathbf{I}'), \mathcal{H}_{S'})$$

for each $i \in [2n_2 + 1]$. Since Y_i is not used in the construction of $\tilde{h}_{S', \mathbf{I}', i}$, we can also denote $\tilde{h}_{S', \mathbf{I}', i}$ as $\tilde{h}_{S'_{-i}, X_i, \mathbf{I}', i}$, where $S'_{-i} := ((X_1, Y_1), \dots, (X_{i-1}, Y_{i-1}), (X_{i+1}, Y_{i+1}), \dots, (X_{2n_2+1}, Y_{2n_2+1}))$. Thus, treating $S := (S_1, S_2) \sim \mathcal{D}^{n_1+2n_2}$ as the training sample, we can define the classifier $\hat{h}_S \in \mathcal{Y}^{\mathcal{X}}$ by

$$\hat{h}_S(x) := \tilde{h}_{S^2, x, \mathbf{I}', 2n_2+1}(2n_2 + 1),$$

where we emphasize that the RHS depends on S^1 through the construction of $\hat{\mu}$. Then, recalling that $(X, Y) \sim \mathcal{D}$ is independent of S , our task is to upper bound the expected error rate of $\hat{h}_S$:

$$\mathbb{P}(\hat{h}_S(X) \neq Y).$$

We first relate $\hat{h}_{S', \mathbf{I}}$ to $\tilde{h}_{S', \mathbf{I}', i}$ for $i \in [2n_2 + 1]$:

$$\begin{aligned} \mathbb{P}(\hat{h}_{S', \mathbf{I}}(I) \neq Y_I | S'') &\geq \frac{1}{2n_2 + 1} \sum_{i=1}^{2n_2+1} \mathbb{E} \left[\mathbb{1}_{i \notin \mathbf{I}} \mathbb{1}_{\hat{h}_{S', \mathbf{I}}(i) \neq Y_i} | S'' \right] \\ &= \frac{1}{2n_2 + 1} \sum_{i=1}^{2n_2+1} \mathbb{P}(i \notin \mathbf{I}) \mathbb{E} \left[\mathbb{1}_{\hat{h}_{S', \mathbf{I}}(i) \neq Y_i} | S'', i \notin \mathbf{I} \right] \\ &\geq \frac{1}{2(2n_2 + 1)} \sum_{i=1}^{2n_2+1} \mathbb{E} \left[\mathbb{1}_{\tilde{h}_{S', \mathbf{I}', i}(i) \neq Y_i} | S'' \right], \end{aligned}$$

which implies that

$$\begin{aligned} \mathbb{P}(\hat{h}_{S', \mathbf{I}}(I) \neq Y_I | S^1) &\geq \frac{1}{2(2n_2 + 1)} \sum_{i=1}^{2n_2+1} \mathbb{E} \left[\mathbb{1}_{\tilde{h}_{S', \mathbf{I}', i}(i) \neq Y_i} | S^1 \right] \\ &= \frac{1}{2(2n_2 + 1)} \sum_{i=1}^{2n_2+1} \mathbb{E} \left[\mathbb{E} \left[\mathbb{1}_{\tilde{h}_{S', \mathbf{I}', i}(i) \neq Y_i} | S^1, \mathbf{I}' \right] | S^1 \right]. \end{aligned}$$

Conditional on $\mathbf{I}'$ and S^1 , for any $i \in [2n_2 + 1]$ and any sequence $\mathbf{s} \in (\mathcal{X} \times \mathcal{Y})^{2n_2+1}$, we let $\tilde{h}_{\mathbf{s}, \mathbf{I}', i}$ denote the classifier when replacing S' with $\mathbf{s}$ in $\tilde{h}_{S', \mathbf{I}', i}$, i.e., $\tilde{h}_{\mathbf{s}, \mathbf{I}', i} = A_G(T^i(\mathbf{s}, \mathbf{I}'), \mathcal{H}_{\mathbf{s}})$. For any $i \in [2n_2 + 1]$, define the set

$$B_i := \left\{ \mathbf{s} = ((x_1, y_1), \dots, (x_{2n_2+1}, y_{2n_2+1})) \in (\mathcal{X} \times \mathcal{Y})^{2n_2+1} : \tilde{h}_{\mathbf{s}, \mathbf{I}', i}(i) \neq y_i \right\}$$

and the permutation

$$\pi^i : (\mathcal{X} \times \mathcal{Y})^{2n_2+1} \rightarrow (\mathcal{X} \times \mathcal{Y})^{2n_2+1}, (z_1, \dots, z_{2n_2+1}) \mapsto (z_1, \dots, z_{i-1}, z_{2n_2+1}, z_i, \dots, z_{2n_2}).$$

We also define the set

$$B := \left\{ \mathbf{s} = ((x_1, y_1), \dots, (x_{2n_2+1}, y_{2n_2+1})) \in (\mathcal{X} \times \mathcal{Y})^{2n_2+1} : y_{2n_2} \in \hat{\mu}(x_{2n_2}), y_{2n_2+1} \in \hat{\mu}(x_{2n_2+1}), \text{ and } \tilde{h}_{\mathbf{s}, \mathbf{I}', 2n_2+1}(2n_2 + 1) \neq y_{2n_2+1} \right\}.$$

We would like to show that $\pi^i(B) \subseteq B_i$ for all $i \in [2n_2]$. For any $\mathbf{s} = ((x_1, y_1), \dots, (x_{2n_2+1}, y_{2n_2+1})) \in B$, we let $(x_j^i, y_j^i) \in \mathcal{X} \times \mathcal{Y}$ denote the j -th element of $\pi^i(\mathbf{s})$ for each $j \in [2n_2 + 1]$. By the definition of π^i and T^i , we have

$$\begin{aligned} T^{2n_2+1}(\mathbf{s}, \mathbf{I}') &= ((I'_1, y_{I'_1}), \dots, (I'_{n_2}, y_{I'_{n_2}})) \text{ and} \\ T^i(\pi^i(\mathbf{s}), \mathbf{I}') &= ((\rho_i(I'_1), y_{I'_1}), \dots, (\rho_i(I'_{n_2}), y_{I'_{n_2}})). \end{aligned}$$

Define

$$n_{\mathbf{s}} := \sum_{j \in [2n_2]} \mathbb{1}_{y_j \notin \hat{\mu}(x_j)}.$$

Since $y_{2n_2} \in \hat{\mu}(x_{2n_2})$ and $y_{2n_2+1} \in \hat{\mu}(x_{2n_2+1})$, we have

$$n_{\pi^i(\mathbf{s})} := \sum_{j \in [2n_2]} \mathbb{1}_{y_j^i \notin \hat{\mu}(x_j^i)} = \sum_{j \in [2n_2+1] \setminus \{2n_2\}} \mathbb{1}_{y_j \notin \hat{\mu}(x_j)} = n_{\mathbf{s}}$$

and therefore,

$$\begin{aligned} \mathcal{H}_{\pi^i(\mathbf{s})} &= \mathcal{H}_{(x_1^i, \dots, x_{2n_2+1}^i), \hat{\mu}, n_{\pi^i(\mathbf{s})}} \\ &= \mathcal{H}_{(x_1^i, \dots, x_{2n_2+1}^i), \hat{\mu}, n_{\mathbf{s}}} \\ &= \left\{ h|_{(x_1, \dots, x_{i-1}, x_{2n_2+1}, x_i, \dots, x_{2n_2})} : h \in \mathcal{H}, |\{j \in [2n_2+1] : h(x_j) \notin \hat{\mu}(x_j)\}| \leq n_{\mathbf{s}} + 1 \right\} \\ &= \{\kappa_i(h) : h \in \mathcal{H}_{\mathbf{s}}\} \end{aligned}$$

where for any $h \in \mathcal{Y}^{2n_2+1}$, $\kappa_i(h) \in \mathcal{Y}^{2n_2+1}$ is defined by

$$\kappa_i(h)(j) := h(\rho_i^{-1}(j)) \mathbb{1}_{j \neq i} + h(2n_2+1) \mathbb{1}_{j=i}, \quad \forall j \in [2n_2+1],$$

i.e., $h(j) = \kappa_i(h)(\rho_i(i))$ for $j \in [2n_2]$ and $h(2n_2+1) = \kappa_i(h)(i)$. Note that κ_i is a bijection from $\mathcal{H}_{\mathbf{s}}$ to $\mathcal{H}_{\pi^i(\mathbf{s})}$. Thus, for any $h \in \mathcal{H}_{\pi^i(\mathbf{s})}$, we have

$$h(\rho_i(I'_j)) = \kappa_i^{-1}(h)(I'_j), \quad \forall j \in [n_2], \text{ and } h(i) = \kappa_i^{-1}(h)(2n_2+1).$$

Similarly, for any $h \in \mathcal{H}_{\mathbf{s}}$, we have

$$h(I'_j) = \kappa_i(h)(\rho_i(I'_j)), \quad \forall j \in [n_2], \text{ and } h(2n_2+1) = \kappa_i(h)(i).$$

Given the above analysis, we have

$$\begin{aligned} \tilde{h}_{\mathbf{s}, \mathbf{I}', 2n_2+1}(2n_2+1) &= A_G(T^{2n_2+1}(\mathbf{s}, \mathbf{I}'), \mathcal{H}_{\mathbf{s}})(2n_2+1) \\ &= A_G(T^i(\pi^i(\mathbf{s}), \mathbf{I}'), \mathcal{H}_{\pi^i(\mathbf{s})})(i) \\ &= \tilde{h}_{\pi^i(\mathbf{s}), \mathbf{I}', i}(i), \end{aligned}$$

which immediately implies that

$$\tilde{h}_{\pi^i(\mathbf{s}), \mathbf{I}', i}(i) = \tilde{h}_{\mathbf{s}, \mathbf{I}', 2n_2+1}(2n_2+1) \neq y_{2n_2+1} = y_i^i$$

and thus $\pi^i(\mathbf{s}) \in B_i$. Since $i \in [2n_2]$ and $\mathbf{s} \in B$ is arbitrary, we have $\pi^i(B) \subseteq B_i$ for any $i \in [2n_2]$. Then, conditional on $\mathbf{I}'$ and S^1 , we have

$$\mathcal{D}^{2n_2+1}(B_i) \supseteq \mathcal{D}^{2n_2+1}(\pi^i(B)) = \mathcal{D}^{2n_2+1}(B)$$

Since $B \subseteq B_{2n_2+1}$ holds trivially, we have

$$\begin{aligned} \mathbb{P}(\hat{h}_{S', \mathbf{I}}(I) \neq Y_I | S^1) &\geq \frac{1}{2(2n_2+1)} \sum_{i=1}^{2n_2+1} \mathbb{E} \left[\mathbb{E} \left[\mathbb{1}_{\tilde{h}_{S', \mathbf{I}', i}(i) \neq Y_i} | S^1, \mathbf{I}' \right] | S^1 \right] \\ &\geq \frac{1}{2} \mathbb{E} \left[\mathbb{1}_{Y_{2n_2} \in \hat{\mu}(X_{2n_2})} \mathbb{1}_{Y_{2n_2+1} \in \hat{\mu}(X_{2n_2+1})} \mathbb{1}_{\tilde{h}_{S', \mathbf{I}', 2n_2+1}(2n_2+1) \neq Y_{2n_2+1}} | S^1 \right]. \end{aligned}$$

Taking expectation on both sides and applying (18) and (19), we have

$$\begin{aligned} &\frac{C_1 d \log p(n_1) + 2C_2 \beta(\mathcal{H}, n_1) n_2 / n_1 + C_3}{n_2} \\ &= \frac{C_1 d \log p(n_1) + C_2 \mathbb{E}[N] + C_3}{n_2} \\ &\geq \mathbb{P}(\hat{h}_{S', \mathbf{I}}(I) \neq Y_I) \\ &\geq \frac{1}{2} \mathbb{E} \left[\mathbb{1}_{Y_{2n_2} \in \hat{\mu}(X_{2n_2})} \mathbb{1}_{Y_{2n_2+1} \in \hat{\mu}(X_{2n_2+1})} \mathbb{1}_{\tilde{h}_{S', \mathbf{I}', 2n_2+1}(2n_2+1) \neq Y_{2n_2+1}} \right] \end{aligned}$$

which leads to

$$\begin{aligned}
& \mathbb{P}(\hat{h}_S(X) \neq Y) \\
&= \mathbb{P}(\tilde{h}_{S', \mathbf{I}', 2n_2+1}(2n_2+1) \neq Y_{2n_2+1}) \\
&\leq \mathbb{E}\left[1 - \mathbb{1}_{Y_{2n_2} \in \hat{\mu}(X_{2n_2})} \mathbb{1}_{Y_{2n_2+1} \in \hat{\mu}(X_{2n_2+1})}\right] \\
&\quad + \mathbb{E}\left[\mathbb{1}_{Y_{2n_2} \in \hat{\mu}(X_{2n_2})} \mathbb{1}_{Y_{2n_2+1} \in \hat{\mu}(X_{2n_2+1})} \mathbb{1}_{\tilde{h}_{S', \mathbf{I}', 2n_2+1}(2n_2+1) \neq Y_{2n_2+1}}\right] \\
&\leq \mathbb{P}(Y_{2n_2} \notin \hat{\mu}(X_{2n_2})) + \mathbb{P}(Y_{2n_2+1} \notin \hat{\mu}(X_{2n_2+1})) \\
&\quad + \frac{2(C_1 d \log p(n_1) + 2C_2 \beta(\mathcal{H}, n_1) n_2/n_1 + C_3)}{n_2} \\
&\leq \frac{2(\beta(\mathcal{H}, n_1) n_2/n_1 + C_1 d \log p(n_1) + 2C_2 \beta(\mathcal{H}, n_1) n_2/n_1 + C_3)}{n_2}.
\end{aligned}$$

Now, for any $n \in \mathbb{N}$ such that $n \geq 3$, setting $n_1 = n - 2\lfloor n/3 \rfloor$, $n_2 = \lfloor n/3 \rfloor$, $(S, (X, Y)) \sim \mathcal{D}^{n+1}$ with $S \in (\mathcal{X} \times \mathcal{Y})^n$ and $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, we have

$$\mathbb{P}(\hat{h}_S(X) \neq Y) = O\left(\frac{\beta(\mathcal{H}, n_1) + d \log p(n_1)}{n}\right).$$

Then, when $\beta(\mathcal{H}, n)$ and $p(n)$ are nondecreasing in n , by Theorem 2.6, there exists a learner $\mathcal{A}'$ such that for any $n \geq 4$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over $S \sim \mathcal{D}^n$, we have

$$\text{er}_{\mathcal{D}}(\mathcal{A}'(S, \mathcal{H})) = O\left(\frac{\beta(\mathcal{H}, n_1) + d \log p(n_1) + \log(1/\delta)}{n}\right).$$

□

D Proofs of the results in Section 2.3

In this section, we provide the proofs of Theorem 2.8, Corollary 2.9, and Theorem 2.10 in Section 2.3.

Proof of Theorem 2.8. Assume that $\mathcal{A}_{\text{list}}$ satisfies that $\mathcal{M}_{\mathcal{A}_{\text{list}}, \mathcal{H}}(1/2 - \gamma, \nu) < \infty$ for some $\gamma \in (0, 1/2)$ and $\nu \in (0, \gamma/18]$. Define the random variable $J_k := \mathbb{1}_{\text{er}_{\mathcal{D}_k}(\mu_k) > 1/2 - \gamma}$ for any $k \in [K]$. Then, we have

$$\mathbb{E}[J_k] = \mathbb{P}(J_k = 1) \leq \nu.$$

Define the event $\mathcal{E} := \left\{\sum_{k=1}^K J_k \geq 2\nu K\right\}$. By the multiplicative Chernoff bound, we have

$$\mathbb{P}(\mathcal{E}) = \mathbb{P}\left(\sum_{k=1}^K J_k \geq 2\nu K\right) \leq e^{-\nu K/3} \leq \delta$$

as $K \geq \frac{3 \log(1/\delta)}{\nu}$. Define $Z_k := \sum_{i=1}^n \mathcal{D}_k(\{(x_i, y_i)\}) \exp(-\alpha(2\mathbb{1}_{y_i \in \mu_k(x_i)} - 1))$ for all $k \in [K]$. Since $\mathcal{D}_{K+1}$ is a probability distribution over S , we have

$$\begin{aligned}
1 &= \sum_{i=1}^n \mathcal{D}_{K+1}(\{(x_i, y_i)\}) \\
&= \sum_{i=1}^n \frac{\mathcal{D}_k(\{(x_i, y_i)\}) \exp(-\alpha(2\mathbb{1}_{y_i \in \mu_k(x_i)} - 1))}{Z_K} \\
&= \frac{1}{n} \sum_{i=1}^n \frac{\exp\left(-\alpha \sum_{k=1}^K (2\mathbb{1}_{y_i \in \mu_k(x_i)} - 1)\right)}{\prod_{k=1}^K Z_k},
\end{aligned}$$

which implies that

$$n \prod_{k=1}^K Z_k = \sum_{i=1}^n \exp\left(-\alpha \sum_{k=1}^K (2\mathbb{1}_{y_i \in \mu_k(x_i)} - 1)\right). \quad (20)$$

For any $k \in [K]$, we have

$$\begin{aligned} Z_k &= \sum_{i=1}^n \mathcal{D}_k(\{(x_i, y_i)\}) \exp(-\alpha(2\mathbb{1}_{y_i \in \mu_k(x_i)} - 1)) \\ &= \sum_{i \in [n]: y_i \in \mu_k(x_i)} \mathcal{D}_k(\{(x_i, y_i)\}) e^{-\alpha} + \sum_{i \in [n]: y_i \notin \mu_k(x_i)} \mathcal{D}_k(\{(x_i, y_i)\}) e^{\alpha} \\ &= (1 - \text{er}_{\mathcal{D}_k}(\mu_k)) e^{-\alpha} + \text{er}_{\mathcal{D}_k}(\mu_k) e^{\alpha} = \text{er}_{\mathcal{D}_k}(\mu_k) (e^{\alpha} - e^{-\alpha}) + e^{-\alpha}. \end{aligned}$$

If $J_k = 0$, we have

$$Z_k \leq \gamma e^{\alpha} + (1 - \gamma) e^{-\alpha} = \sqrt{1 - \gamma^2}.$$

If $J_k = 1$, we have

$$Z_k \leq e^{\alpha} = \sqrt{1 + \frac{\gamma}{1/2 - \gamma/2}} \leq \sqrt{1 + 4\gamma}.$$

Then, under the event $\mathcal{E}^c$, we can upper bound

$$\begin{aligned} \prod_{k=1}^K Z_k &\leq (1 - \gamma^2)^{(1/2 - \nu)K} (1 + 4\gamma)^{\nu K} \\ &\leq \exp\left(-\gamma K \left(\frac{\gamma}{2} - (\gamma + 4)\nu\right)\right) \\ &\leq \exp\left(-\frac{\gamma K}{4}\right) \leq \frac{\delta}{n}. \end{aligned}$$

where the second last inequality follows from $\nu \leq \frac{\gamma}{18} \leq \frac{\gamma}{4(\gamma+4)}$ and the last inequality follows from $K \geq \frac{4 \log(n/\delta)}{\gamma}$. By (20) and the above inequality, we have that on $\mathcal{E}^c$,

$$\sum_{i=1}^n \exp\left(-\alpha \sum_{k=1}^K (2\mathbb{1}_{y_i \in \mu_k(x_i)} - 1)\right) \leq \delta,$$

which implies that $\exp\left(-\alpha \sum_{k=1}^K (2\mathbb{1}_{y_i \in \mu_k(x_i)} - 1)\right) \leq \delta$ for all $i \in [n]$. It follows that

$$\exp(-f(x_i, y_i)) \leq \delta^{1/\alpha K}$$

for $f(x, y) := \frac{1}{K} \sum_{k=1}^K (2\mathbb{1}_{y \in \mu_k(x)} - 1)$, $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$. Thus, we have

$$f(x_i, y_i) \geq \frac{\log(1/\delta)}{\alpha K} = \frac{2 \log(1/\delta)}{\log\left(\frac{1/2 + \gamma}{1/2 - \gamma}\right) [4 \log(n/\delta)/\gamma]} > 0, \forall i \in [n].$$

Since by definition,

$$f(x, y) > 0 \Leftrightarrow y \in \mu(x), \forall (x, y) \in \mathcal{X} \times \mathcal{Y},$$

we can conclude that on $\mathcal{E}^c$,

$$y_i \in \mu(x_i), \forall i \in [n],$$

i.e., with probability at least $1 - \delta$, $y_i \in \mu(x_i)$ for all $i \in [n]$. Moreover, by da Cunha et al. [2024, Lemma 3.3], the randomized compression scheme $S \mapsto (S^1, \dots, S^K)$ is stable. Thus, we can apply da Cunha et al. [2024, Theorem 1.2] with compression size $s_n = mK = O\left(\frac{\mathcal{M}_{\text{list}, \mathcal{H}(1/2 - \gamma, \nu) \log(n/\delta)}{\gamma}\right)$ to conclude the proof. $\square$

For the proofs of Corollary 2.9 and Theorem 2.10, we will need the following Lemma.

Lemma D.1. *If $x > 0$ satisfies $x \leq a \log(x/a) + b$ for some $a, b > 0$, then, we have $x \leq 2a + 2b$.*

Proof. Define $f(x) := x - a \log(x/a) - b$ for $x > 0$. Then, we have $f'(x) = \frac{x-a}{x}$, which implies that f decreases with x for $x \in (0, a)$ and increases with x for $x > a$. Since $2a + 2b > a$, it suffices to prove that $f(2a + 2b) \geq 0$. Indeed,

$$f(2a + 2b) = (2 - \log 2)a + b - a \log((a + b)/a) > a((a + b)/a - \log((a + b)/a)) \geq 0.$$

□

Proof of Corollary 2.9. By [Brukhim et al. \[2022, Theorem 36\]](#) (choosing $t = \lceil \sqrt{d} \rceil$), there exists an $n \rightarrow r$ sample compression scheme $\mathcal{A}_M$ for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ of DS dimension $\dim(\mathcal{H}) = d < \infty$ where

$$r = O((d^{3/2} + d \log(p)) \log(n)) \text{ with } p = O((e\sqrt{d})^{\sqrt{d}} \log(n)). \quad (21)$$

Then, by [David et al. \[2016, Theorem 3.1\]](#), there exists a universal constant $C > 0$ such that for any $\mathcal{D} \in \text{RE}(\mathcal{H})$, $\delta \in (0, 1)$, $n \in \mathbb{N}$ large enough, and $S \sim \mathcal{D}^n$, letting $h_S := \mathcal{A}_M(S, \mathcal{H})$ denote the output classifier of the above $n \rightarrow r$ sample compression scheme $\mathcal{A}_M$, we have that

$$\text{er}_{\mathcal{D}}(h_S) \leq \frac{C(r \log(n/r) + \log(1/\delta))}{n}$$

with probability at least $1 - \delta$. Thus, for any $\varepsilon \in (0, 1)$, if

$$n \geq \frac{Cr}{\varepsilon} \log(n/r) + \frac{C \log(1/\delta)}{\varepsilon} = \frac{Cr}{\varepsilon} \log\left(\frac{n}{Cr/\varepsilon}\right) + \frac{Cr}{\varepsilon} \log\left(\frac{C}{\varepsilon}\right) + \frac{C \log(1/\delta)}{\varepsilon},$$

we have $\text{er}_{\mathcal{D}}(h_S) \leq \varepsilon$ with probability at least $1 - \delta$. By [Lemma D.1](#), it suffices to require

$$n \geq \frac{2Cr}{\varepsilon} \log\left(\frac{Ce}{\varepsilon}\right) + \frac{2C \log(1/\delta)}{\varepsilon} = \frac{2C}{\varepsilon} \left(r \log\left(\frac{Ce}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right) \right).$$

Applying the upper bound of r in (21), it suffices to require

$$n \geq C' \left(\frac{(d^{3/2} + d \log(p)) \log(1/\varepsilon)}{\varepsilon} \log(n) + \frac{\log(1/\delta)}{\varepsilon} \right)$$

for some universal constant $C' > 0$. By [Lemma D.1](#) again, it suffices to require

$$n \geq \frac{2C'(d^{3/2} + d \log(p)) \log(1/\varepsilon)}{\varepsilon} \log\left(\frac{eC'(d^{3/2} + d \log(p)) \log(1/\varepsilon)}{\varepsilon}\right) + \frac{2C' \log(1/\delta)}{\varepsilon}.$$

Since $\log(\log(1/\varepsilon)/\varepsilon) = \log(1/\varepsilon) + \log \log(1/\varepsilon) \leq 2 \log(1/\varepsilon)$, it suffices to require

$$n \geq C'' \left(\frac{(d^{3/2} + d \log(p)) \log(1/\varepsilon)}{\varepsilon} \log\left(\frac{d^{3/2} + d \log(p)}{\varepsilon}\right) + \frac{\log(1/\delta)}{\varepsilon} \right)$$

for some universal constant $C'' > 0$. Applying the upper bound of p in (21), it suffices to require that

$$n \geq C''' \left(\frac{(d^{3/2} \log(d) + d \log \log(n)) \log(1/\varepsilon)}{\varepsilon} \log\left(\frac{d^{3/2} \log(d) + d \log \log(n)}{\varepsilon}\right) + \frac{\log(1/\delta)}{\varepsilon} \right)$$

for some universal constant $C''' > 0$. For $\varepsilon = 1/6$ and $\delta = 1/54$, we require that

$$n \geq C_1(d^{3/2} \log(d) + d \log \log(n)) \log(d^{3/2} \log(d) + d \log \log(n))$$

for some universal constant $C_1 > 0$. If $C_1(\sqrt{2C_1} + 1)d^{3/2} \log(d) \log((\sqrt{2C_1} + 1)d^{3/2} \log(d)) \leq n \leq e^{d\sqrt{2C_1d}}$, we have $\sqrt{2C_1}d^{3/2} \log(d) \geq d \log \log(n)$ and

$$\begin{aligned} & C_1(d^{3/2} \log(d) + d \log \log(n)) \log(d^{3/2} \log(d) + d \log \log(n)) \\ & \leq C_1(\sqrt{2C_1} + 1)d^{3/2} \log(d) \log((\sqrt{2C_1} + 1)d^{3/2} \log(d)) \leq n. \end{aligned}$$

If $n > e^{d\sqrt{2C_1d}}$, we have $\log(\log(n)) > \sqrt{2C_1d} \log(d)$, $\log(\log(n))^2 > 2C_1d \log^2(d) \geq 2C_1(d \log(2d) + d)$, and

$$\begin{aligned} & C_1(d^{3/2} \log(d) + d \log \log(n)) \log(d^{3/2} \log(d) + d \log \log(n)) \\ & \leq 2C_1d \log(\log(n)) \log(2d \log \log(n)) \\ & = 2C_1d \log(2d) \log(\log(n)) + 2C_1d \log(\log(n)) \log(\log \log(n)) \\ & \leq \log(\log(n))^3 \log(\log \log(n)) \leq n. \end{aligned}$$

Therefore, we can conclude that if $n \geq C_1(\sqrt{2C_1} + 1)d^{3/2} \log(d) \log((\sqrt{2C_1} + 1)d^{3/2} \log(d)) = \Theta(d^{3/2} \log^2(d))$, we have $\mathbb{P}(\text{er}_{\mathcal{D}}(h_S) > 1/6) \leq 1/54$, i.e.,

$$\mathcal{M}_{\mathcal{A}_M, \mathcal{H}}(1/2 - 1/3, 1/54) = O(d^{3/2} \log^2(d)).$$

Let $\mathcal{A}_B$ denote the multiclass learner output by $\mathcal{A}_{\text{boost}}$ in Algorithm 2 using $\mathcal{A}_M$ as the weak list learner of size 1. Then, by Theorem 2.8, we have

$$\varepsilon_{\mathcal{A}_B, \mathcal{H}}(n) = O\left(\frac{d^{3/2} \log^2(d) \log(n)}{n}\right), \quad \forall n \in \mathbb{N}.$$

Next, by Theorem 2.6, there exists a multiclass learner $\mathcal{A}$ such that for any $\mathcal{D} \in \text{RE}(\mathcal{H})$, $\delta \in (0, 1)$, $n \in \mathbb{N}$, and $S \sim \mathcal{D}^n$, with probability at least $1 - \delta$, we have

$$\text{er}_{\mathcal{D}}(\mathcal{A}(S, \mathcal{H})) = O\left(\frac{d^{3/2} \log^2(d) \log(n) + \log(1/\delta)}{n}\right).$$

Setting $\delta = 1/n$ and observing that $\text{er}_{\mathcal{D}} \in [0, 1]$, it follows that $\varepsilon_{\mathcal{A}, \mathcal{H}}(n) = O\left(\frac{d^{3/2} \log^2(d) \log(n)}{n}\right)$.

Moreover, by Lemma D.1, for any $\varepsilon \in (0, 1)$, $n \geq O\left(\frac{d^{3/2} \log^2(d) \log(d/\varepsilon) + \log(1/\delta)}{\varepsilon}\right)$ implies that $\text{er}_{\mathcal{D}}(\mathcal{A}(S, \mathcal{H})) \leq \varepsilon$ with probability at least $1 - \delta$. Thus, we have

$$\mathcal{M}_{\mathcal{A}, \mathcal{H}}(\varepsilon, \delta) = O\left(\frac{d^{3/2} \log^2(d) \log(d/\varepsilon) + \log(1/\delta)}{\varepsilon}\right).$$

□

Proof of Theorem 2.10. By Brukhim et al. [2022, Theorem 39] (choosing $t = \lceil \sqrt{d} \rceil$), there exists an $n \rightarrow r$ list sample compression scheme $\mathcal{A}_{\text{list}}$ of size $p = O((e\sqrt{d})^{\sqrt{d}} \log(n))$ for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ of DS dimension $\dim(\mathcal{H}) = d < \infty$ where

$$r = O(d^{3/2} \log(n)). \quad (22)$$

Define the following loss function for menus μ ,

$$\ell(\mu, (x, y)) := \mathbb{1}_{y \notin \mu(x)}, \quad \forall (x, y) \in \mathcal{X} \times \mathcal{Y}.$$

Then, we can apply the proof of David et al. [2016, Theorem 3.1] with the loss function ℓ to show that there exists a universal constant $C > 0$ such that for any $n \rightarrow r$ list sample compression scheme $\mathcal{A}_{LSC}$ for $\mathcal{H}$, any $\mathcal{D} \in \text{RE}(\mathcal{H})$, $\delta \in (0, 1)$, $n \in \mathbb{N}$ large enough, and $S \sim \mathcal{D}^n$, letting $\mu_S := \mathcal{A}_{LSC}(S, \mathcal{H})$ denote the output menu, we have

$$\text{er}_{\mathcal{D}}(\mathcal{A}_{LSC}(S, \mathcal{H})) = \mathbb{E}_{(X, Y) \sim \mathcal{D}}[\ell(\mu_S, (X, Y)) \mid S] \leq \frac{C(r \log(n/r) + \log(1/\delta))}{n}$$

with probability at least $1 - \delta$. Thus, for any $\varepsilon \in (0, 1)$, if

$$n \geq \frac{Cr}{\varepsilon} \log(n/r) + \frac{C \log(1/\delta)}{\varepsilon} = \frac{Cr}{\varepsilon} \log\left(\frac{n}{Cr/\varepsilon}\right) + \frac{Cr}{\varepsilon} \log\left(\frac{C}{\varepsilon}\right) + \frac{C \log(1/\delta)}{\varepsilon},$$

we have $\text{er}_{\mathcal{D}}(\mathcal{A}_{LSC}(S, \mathcal{H})) \leq \varepsilon$ with probability at least $1 - \delta$. By Lemma D.1, it suffices to require

$$n \geq \frac{2Cr}{\varepsilon} \log\left(\frac{Ce}{\varepsilon}\right) + \frac{2C \log(1/\delta)}{\varepsilon} = \frac{2C}{\varepsilon} \left(r \log\left(\frac{Ce}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)\right).$$

Since $\mathcal{A}_{\text{list}}$ is an $n \rightarrow r$ list sample compression scheme for $\mathcal{H}$ with r bounded in (22), The above results imply that in order for $\mathbb{P}(\text{er}_{\mathcal{D}}(\mathcal{A}_{\text{list}}(S, \mathcal{H})) > \varepsilon) \leq \delta$, it suffices to require

$$n \geq C' \left(\frac{d^{3/2} \log(1/\varepsilon)}{\varepsilon} \log(n) + \frac{\log(1/\delta)}{\varepsilon}\right)$$

for some universal constant $C' > 0$. By Lemma D.1 again, it suffices to require

$$n \geq \frac{2C'd^{3/2} \log(1/\varepsilon)}{\varepsilon} \log\left(\frac{eC'd^{3/2} \log(1/\varepsilon)}{\varepsilon}\right) + \frac{2C' \log(1/\delta)}{\varepsilon}.$$

Since $\log(\log(1/\varepsilon)/\varepsilon) = \log(1/\varepsilon) + \log \log(1/\varepsilon) \leq 2 \log(1/\varepsilon)$, it suffices to require

$$n \geq C'' \left(\frac{d^{3/2} \log(d/\varepsilon) \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon} \right)$$

for some universal constant $C'' > 0$, which implies that

$$\mathcal{M}_{\mathcal{A}_{\text{list}}, \mathcal{H}}(1/2 - 1/3, 1/54) = O\left(d^{3/2} \log(d)\right).$$

Now, we can define $\mathcal{A}_L$ to be the list learner output by $\mathcal{A}_{\text{boost}}$ in Algorithm 2 using $\mathcal{A}_{\text{list}}$ as the weak list learner of size $O((e\sqrt{d})^{\sqrt{d}} \log(n))$ for training sample size $n \in \mathbb{N}$. Then, by Theorem 2.8, we have that for any training sample size $n \in \mathbb{N}$,

$$\varepsilon_{\mathcal{A}_L, \mathcal{H}}(n) = O\left(\frac{d^{3/2} \log(d) \log(n)}{n}\right)$$

and the size of $\mathcal{A}_L$ is also $O((e\sqrt{d})^{\sqrt{d}} \log(n))$. □

E Proof of Theorem 2.11

Proof. By Theorem 2.10, there exists a list learner $\mathcal{A}_L$ with $p(\mathcal{H}, n) = O((e\sqrt{d})^{\sqrt{d}} \log(n))$ and $\beta(\mathcal{H}, n) = O(d^{3/2} \log(d) \log(n))$ for any $n \in \mathbb{N}$ and concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim(\mathcal{H}) = d \in \mathbb{N}$ in the context of Theorem 2.7. Thus, by Theorem 2.7, there exists a multiclass learner $\mathcal{A}_{\text{multi}} = \mathcal{A}'_{\text{red}}$ such that for any $\mathcal{D} \in \text{RE}(\mathcal{H})$, any $\delta \in (0, 1)$, $S \sim \mathcal{D}^n$, and $n_1 = n - 2\lfloor n/3 \rfloor$, with probability at least $1 - \delta$,

$$\begin{aligned} \text{er}_{\mathcal{D}}(\mathcal{A}_{\text{multi}}(S, \mathcal{H})) &= O\left(\frac{\beta(\mathcal{H}, n_1) + d \log p(\mathcal{H}, n_1) + \log(1/\delta)}{n}\right) \\ &= O\left(\frac{d^{3/2} \log(d) \log(n) + d^{3/2} \log(e\sqrt{d} \log(n)) + \log(1/\delta)}{n}\right) \\ &= O\left(\frac{d^{3/2} \log(d) \log(n) + \log(1/\delta)}{n}\right). \end{aligned}$$

For any $\varepsilon \in (0, 1)$, by Lemma D.1, if

$$\begin{aligned} n &\geq \frac{2d^{3/2} \log(d)}{\varepsilon} \left(1 + \frac{5}{2} \log(d/\varepsilon)\right) + \frac{2 \log(1/\delta)}{\varepsilon} \\ &\geq \frac{2d^{3/2} \log(d)}{\varepsilon} \left(1 + \frac{3}{2} \log(d) + \log \log(d) + \log(1/\varepsilon)\right) + \frac{2 \log(1/\delta)}{\varepsilon} \\ &\geq \frac{2d^{3/2} \log(d)}{\varepsilon} \log\left(\frac{ed^{3/2} \log(d)}{\varepsilon}\right) + \frac{2 \log(1/\delta)}{\varepsilon}, \end{aligned}$$

then, we have

$$n \geq \frac{d^{3/2} \log(d)}{\varepsilon} \log(n) + \frac{\log(1/\delta)}{\varepsilon}$$

i.e.,

$$\frac{d^{3/2} \log(d) \log(n) + \log(1/\delta)}{n} \leq \varepsilon.$$

It follows that

$$\mathcal{M}_{\mathcal{A}_{\text{multi}}, \mathcal{H}}(\varepsilon, \delta) = O\left(\frac{d^{3/2} \log(d) \log(d/\varepsilon) + \log(1/\delta)}{\varepsilon}\right).$$

(9) follows directly from (6) by plugging in $p(\mathcal{H}, n) = f_1(d)$ and $\beta(\mathcal{H}, n) = f_2(d)$ for $d = \dim(\mathcal{H})$ and any $n \in \mathbb{N}$. □

F Classes of DS dimension 1

In this section, we present the proof of Proposition 3.3 and Theorem 3.2.

Proof of Proposition 3.3. Let E_n denote the edge set of the hypergraph $\mathcal{G}(V_n)$. Suppose on the contrary that there exists a cycle consisting of pairwise different vertices $\mathbf{y}^0, \dots, \mathbf{y}^{m-1} \in V_n$ and pairwise different edges $\mathbf{e}^0 = (e_{i_0, f_0}, i_0), \dots, \mathbf{e}^{m-1} = (e_{i_{m-1}, f_{m-1}}, i_{m-1}) \in E_n$ for some $m \in \{4, \dots, |V_n|\}$ such that $\mathbf{y}^j, \mathbf{y}^{(j+1) \bmod m} \in (e_{i_j, f_j}, i_j)$ for all $0 \leq j \leq m-1$. Since the edges are pairwise different, by the definition of E_n , we have $i_j \neq i_{(j-1) \bmod m}$ for all $0 \leq j \leq m-1$.

Define $a := i_{m-1} \in [n]$, $b := i_0 \in [n]$, $p_0 := y_a^0 = y_a^1$, $p_{-1} := y_a^{m-1} \neq p_0$, and $q_0 := y_b^0 = y_b^{m-1}$. Then, we have $a \neq b$.

For any $k \in \mathbb{N}$, we define

$$j_k := \max\{j \in \mathbb{N}_0 : j \leq m-1, y_a^j = p_{k-1}\},$$

$$p_k := y_a^{(j_k+1) \bmod m}, \text{ and } q_k := y_b^{j_k} = y_b^{(j_k+1) \bmod m} \text{ because } i_{j_k} = a. \text{ Define}$$

$$K := \min\{k \in \mathbb{N} : p_k = p_{-1}\}.$$

By definition, we have $p_0, \dots, p_K$ are pairwise different. There are two cases depending on the values of $q_1, \dots, q_K$ as follows.

1. Suppose that there exists some $k \in [K]$ such that $q_k \neq q_0$. Define $l_0 := 0$ and

$$l_w := \min\{k \in [K] : k > l_{w-1}, q_k \neq q_{l_{w-1}}\}$$

for all $w \in \mathbb{N}$ with the convention that $\inf \emptyset = +\infty$. Define

$$v := \max\{w \in \mathbb{N} : l_w \neq +\infty\}.$$

We have $v \in [K]$ and $q_{l_w} \neq q_{l_{w-1}}$ for all $w \in [v]$. Note that

$$p_{-1} = p_K = y_a^{m-1}, \quad q_0 = y_b^{m-1},$$

$$p_{l_0} = p_0 = y_a^0, \quad q_{l_0} = q_0 = y_b^0,$$

$$p_{l_w-1} = y_a^{j_{l_w-1}+1}, \quad q_{l_{w-1}} = y_b^{j_{l_{w-1}+1}},$$

$$p_{l_w-1} = y_a^{j_{l_w}}, \quad q_{l_w} = y_b^{j_{l_w}},$$

$$p_{l_w} = y_a^{j_{l_w}+1}, \quad q_{l_w} = y_b^{j_{l_w}+1}$$

for all $w \in [v]$, and

$$p_K = p_{-1} = y_a^{j_K+1}, \quad q_K = y_b^{j_K+1}.$$

Since it always holds that $q_K = q_{l_v}$, there are two cases depending on the value of q_{l_v} and q_0 as follows.

- 1.1 Suppose that $q_{l_v} \neq q_0$. Then, $V_n|_{(a,b)}$ contains the following pseudo-cube of dimension 2:

$$\begin{aligned} & (p_{l_1-1}, q_0), \\ & (p_{l_1-1}, q_{l_1}), \\ & (p_{l_2-1}, q_{l_1}), \\ & (p_{l_2-1}, q_{l_2}), \\ & \vdots \\ & (p_{l_v-1}, q_{l_{v-1}}), \\ & (p_{l_v-1}, q_{l_v}), \\ & (p_{-1}, q_{l_v}), \\ & (p_{-1}, q_0), \end{aligned}$$

which contradicts the assumption that $\dim(V_n) = 1$.

1.2 Suppose that $q_{l_v} = q_0$. Then, $V_n|_{(a,b)}$ contains the following pseudo-cube of dimension 2:

$$\begin{aligned} & (p_{l_1-1}, q_0), \\ & (p_{l_1-1}, q_{l_1}), \\ & (p_{l_2-1}, q_{l_1}), \\ & (p_{l_2-1}, q_{l_2}), \\ & \vdots \\ & (p_{l_v-1}, q_{l_{v-1}}), \\ & (p_{l_v-1}, q_{l_v} = q_0), \end{aligned}$$

which contradicts the assumption that $\dim(V_n) = 1$.

Thus, Case 1 does not exist.

2. Suppose that $q_k = q_0$ for all $k \in [K]$. Since $y_a^0 = p_0 = p_a^{j_1}$, $y_b^0 = q_0 = q_1 = p_b^{j_1}$, and $\mathbf{y}^0 \neq \mathbf{y}^{j_1}$, there must exist some $c \in [n] \setminus \{a, b\}$ such that $y_c^{m-1} = y_c^1 = y_c^0 \neq y_c^{j_1} = y_c^{j_1+1}$. We define $r_0 := y_c^0$ and $r_k := y_c^{j_k} = y_c^{j_k+1}$ for $k \in [K]$. There are two cases depending on the value of r_k for $k \in [K]$ as follows.

2.1 Suppose that there exists some $k \in [K]$ such that $r_k \neq r_1$. Similar to Case 1, we define $l_0 := 1$ and

$$l_w := \min\{k \in [K] : k > l_{w-1}, r_k \neq r_{l_{w-1}}\}$$

for $w \in \mathbb{N}$. Define $v := \max\{w \in \mathbb{N} : l_w \neq +\infty\}$. We have $v \in [K]$ and $r_{l_w} \neq r_{l_{w-1}}$ for all $w \in [v]$. Note that

$$\begin{aligned} p_{-1} &= p_K = y_a^{m-1}, & r_0 &= y_c^{m-1}, \\ p_0 &= y_a^0, & r_0 &= y_c^0, \\ p_0 &= y_a^{j_1}, & r_{l_0} &= r_1 = y_c^{j_1}, \\ p_{l_w-1} &= y_a^{j_{l_w-1}+1}, & r_{l_{w-1}} &= y_c^{j_{l_w-1}+1}, \\ p_{l_w-1} &= y_a^{j_{l_w}}, & r_{l_w} &= y_c^{j_{l_w}}, \end{aligned}$$

for all $w \in [v]$, and

$$p_K = p_{-1} = y_a^{j_K+1}, \quad r_K = y_c^{j_K+1}.$$

Since it always holds that $r_K = r_{l_v}$, there are two cases depending on the value of q_{l_v} and q_0 as follows.

2.1.1 Suppose that $r_{l_v} \neq r_0$. Then, $V_n|_{(a,c)}$ contains the following pseudo-cube of dimension 2:

$$\begin{aligned} & (p_0, r_0), \\ & (p_0, r_{l_0} = r_1), \\ & (p_{l_1-1}, r_{l_0}), \\ & (p_{l_1-1}, q_{l_1}), \\ & \vdots \\ & (p_{l_v-1}, r_{l_{v-1}}), \\ & (p_{l_v-1}, r_{l_v}), \\ & (p_{-1}, r_{l_v}), \\ & (p_{-1}, r_0), \end{aligned}$$

which contradicts the assumption that $\dim(V_n) = 1$.

2.1.2 Suppose that $r_{l_v} = r_0$. Then, $V_n|_{(a,c)}$ contains the following pseudo-cube of dimension 2:

$$\begin{aligned} & (p_0, r_0), \\ & (p_0, r_{l_0} = r_1), \\ & (p_{l_1-1}, r_{l_0}), \\ & (p_{l_1-1}, q_{l_1}), \\ & \vdots \\ & (p_{l_v-1}, r_{l_{v-1}}), \\ & (p_{l_v-1}, r_{l_v} = r_0), \end{aligned}$$

which contradicts the assumption that $\dim(V_n) = 1$.

2.2 Suppose that $r_k = r_1$ for all $k \in [K]$. Then, we have

$$\begin{aligned} p_{-1} &= y_a^{m-1}, & r_0 &= y_c^{m-1}, \\ p_0 &= y_a^0, & r_0 &= y_c^0, \\ p_0 &= y_a^{j_1}, & r_1 &= y_c^{j_1}, \\ p_{-1} &= y_a^{j_K+1}, & r_1 &= y_c^{j_K+1}, \end{aligned}$$

which implies that $V_n|_{(a,c)}$ contains a pseudo-cube of dimension 2.

Thus, Case 2 does not exist either.

In conclusion, there exists no cycle in the hypergraph $\mathcal{G}(V_n) = (V_n, E_n)$. $\square$

Proof of Theorem 3.2. It suffices to show that for any $n \in \mathbb{N}$ and $V_n \subseteq \mathcal{Y}^n$, the average degree of $G = \mathcal{G}(V_n) = (V_n, E_n)$ with $|V_n| < \infty$ and $\dim(V_n) = 1$ is at most 2.

We prove by induction on $|V_n|$. When $|V_n| = 1$, we have $E_n = \emptyset$ and $\text{avgdeg}(G) = 0 < 2$. When $|V_n| = 2$, we have $\sum_{e \in E_n: |e| \geq 2} |e| \leq 2$ and $\text{avgdeg}(G) \leq 1 < 2$. Suppose that $\text{avgdeg}(G) \leq 2$ for any V_n of size $|V_n| \leq m$ with some $m \in \mathbb{N}$. When $|V_n| = m + 1$, since there is no cycle in G according to Proposition 3.3, the set of vertices of degree 1 ($V_n^1 := \{\mathbf{y} \in V_n : \deg(\mathbf{y}) = 1\}$) is not empty. Define $V_n^2 := V_n \setminus V_n^1$ and E_n^2 to be the edge set such that (V_n^2, E_n^2) is the one-inclusion graph on V_n^2 . Then, we have $|V_n^2| = |V_n| - |V_n^1| \leq m$ and

$$\sum_{e \in E_n: |e| \geq 2} |e| \leq \sum_{e \in E_n^2: |e| \geq 2} |e| + 2|V_n^1|$$

because deleting a vertex of degree 1 decreases the total degree by at most 2. By the induction hypothesis, we have $\sum_{e \in E_n^2} |e| \leq 2|V_n^2|$. Thus,

$$\sum_{e \in E_n: |e| \geq 2} |e| \leq 2(|V_n^1| + |V_n^2|) = 2|V_n|$$

which implies that $\text{avgdeg}(G) \leq 2$. By induction, $\text{avgdeg}(G) \leq 2$ for any V_n with $|V_n| < \infty$ and $\dim(V_n) = 1$. $\square$

G Pivot shifting

In this section, we present the proofs of Lemma 3.6 and Lemma 3.8.

Proof of Lemma 3.6. For notational convenience, we let V'_{n-1} denote $\mathfrak{P}(V_n)$ which is defined in Definition 3.5. Define

$$V_{n-1} := \bigcup_{y \in \mathcal{Y}} \{(y_1, \dots, y_{n-1}) \in \mathcal{Y}^{n-1} : (y_1, \dots, y_{n-1}, y) \in V_n\}$$

and

$$V''_{n-1} := V_{n-1} \setminus V'_{n-1}.$$

Let E_n denote the edge set in $\mathcal{G}(V_n)$ and E_{n-1} denote the edge set in $\mathcal{G}(V_{n-1})$. For any $y \in \mathcal{Y}$, define

$$V_{n,y} := \{(y_1, \dots, y_n) \in V_n : y_n = y, (y_1, \dots, y_{n-1}) \in V'_{n-1}\}.$$

By the assumption on $a \in \mathcal{Y}$, we have $|V_n| = |V''_{n-1}| + \sum_{y \in \mathcal{Y}} |V_{n,y}|$, $|V'_{n-1}| = |V_{n,a}|$, and $|V_n| - |V_{n-1}| = \sum_{y \in \mathcal{Y} \setminus \{a\}} |V_{n,y}|$. Defining

$$E_n^n := \{(e_{i,f}, i) \in E_n : i = n\},$$

we have

$$\sum_{e \in E_n^n} (|e| - 1) = |V_n| - |V_{n-1}|.$$

For any $y \in \mathcal{Y}$, we define

$$E_{n,y} := \{(e_{i,f}, i) \in E_n : f(n) = y, i \in [n-1]\}$$

and let $E'_{n,y}$ denote the edge set in $\mathcal{G}(V_{n,y})$; for any $e = (e_{i,f}, i) \in E_{n,y}$, we define

$$s_1(e) := |\{\mathbf{y} \in e_{i,f} : \mathbf{y} \in V_{n,y}\}| \text{ and } s_2(e) := |\{\mathbf{y} \in e_{i,f} : \mathbf{y} \notin V_{n,y}\}|.$$

Then, we have $|e| = s_1(e) + s_2(e)$, $E_n = (\cup_{y \in \mathcal{Y}} E_{n,y}) \cup E_n^n$, and

$$\begin{aligned} & \sum_{e \in E_{n-1}} (|e| - 1) \\ & \geq \sum_{e \in E_{n,a}} (|e| - 1) + \sum_{y \in \mathcal{Y} \setminus \{a\}} \left(\sum_{e \in E_{n,y} : s_1(e)=0} (s_2(e) - 1) + \sum_{e \in E_{n,y} : s_1(e) \geq 1} s_2(e) \right). \end{aligned}$$

Note that by the induction hypothesis, we have

$$\sum_{e \in E_{n-1}} (|e| - 1) \leq d|V_{n-1}|$$

and by definition, we also have

$$\sum_{e \in E'_{n,y}} (|e| - 1) = \sum_{e \in E_{n,y} : s_1(e) \geq 2} (s_1(e) - 1).$$

We claim that $\dim(V_{n,y}) \leq d-1$ for all $y \in \mathcal{Y} \setminus \{a\}$. Suppose on the contrary that $\dim(V_{n,y}) > d-1$. Since $\dim(V_n) = d$, we have $\dim(V_{n,y}) = d$ and there exists a set $\mathbf{i} := \{i_1, \dots, i_d\} \subseteq [n]$ with $i_1 < i_2 < \dots < i_d$ such that $V_{n,y}[\mathbf{i}]$ contains a pseudo-cube H_y of dimension d . Since for any $(y_1, \dots, y_n) \in V_{n,y}$, we have $y_n = y$, it must hold that $i_d \leq n-1$. Now, we can define

$$\begin{aligned} H_{y,a} &:= \{(y_{i_1}, \dots, y_{i_d}, a) \in \mathcal{Y}^{d+1} : (y_{i_1}, \dots, y_{i_d}) \in H_y\} \\ &\quad \cup \{(y_{i_1}, \dots, y_{i_d}, y) \in \mathcal{Y}^{d+1} : (y_{i_1}, \dots, y_{i_d}) \in H_y\}. \end{aligned}$$

By the assumption that $|V'_{n-1}| = |V_{n,a}|$, we have $|H_{y,a}| = 2|H_y|$ and $H_{y,a} \subseteq V_n|_{(i_1, \dots, i_d, n)}$. For any $k \in [d]$ and any $(y_{i_1}, \dots, y_{i_d}, y), (y_{i_1}, \dots, y_{i_d}, a) \in H_{y,a}$, since there exists a k -neighbor of $(y_{i_1}, \dots, y_{i_d})$ denoted by $(y_{i_1}, \dots, y'_{i_k}, \dots, y_{i_d})$ in H_y , $(y_{i_1}, \dots, y'_{i_k}, \dots, y_{i_d}, y') \in H_{y,a}$ is a k -neighbor of $(y_{i_1}, \dots, y_{i_d}, y')$ in $H_{y,a}$ for $y' = y, a$. Moreover, $(y_{i_1}, \dots, y_{i_d}, y)$ is a $(d+1)$ -neighbor of $(y_{i_1}, \dots, y_{i_d}, a)$ in $H_{y,a}$ and vice-versa. Thus, $H_{y,a}$ is a pseudo-cube of dimension $d+1$, which contradicts the assumption that $\dim(V_n) \leq d$. Therefore, we must have $\dim(V_{n,y}) \leq d-1$. Then, by the induction hypothesis, we have for any $y \in \mathcal{Y} \setminus \{a\}$,

$$\sum_{e \in E'_{n,y}} (|e| - 1) \leq (d-1)|V_{n,y}|.$$

Summarizing the results above, we have

$$\begin{aligned}
& \sum_{e \in E_n} (|e| - 1) \\
&= \sum_{e \in E_n^n} (|e| - 1) + \sum_{y \in \mathcal{Y}} \sum_{e \in E_{n,y}} (|e| - 1) \\
&= \sum_{e \in E_n^n} (|e| - 1) + \sum_{e \in E_{n,a}} (|e| - 1) + \\
& \quad \sum_{y \in \mathcal{Y} \setminus \{a\}} \left(\sum_{e \in E_{n,y}: s_1(e)=0} (s_2(e) - 1) + \sum_{e \in E_{n,y}: s_1(e)=1} s_2(e) + \sum_{e \in E_{n,y}: s_1(e) \geq 2} (s_2(e) + (s_1(e) - 1)) \right) \\
&= \sum_{e \in E_n^n} (|e| - 1) + \sum_{e \in E_{n,a}} (|e| - 1) + \sum_{y \in \mathcal{Y} \setminus \{a\}} \left(\sum_{e \in E_{n,y}: s_1(e)=0} (s_2(e) - 1) + \sum_{e \in E_{n,y}: s_1(e) \geq 1} s_2(e) \right) \\
& \quad + \sum_{y \in \mathcal{Y} \setminus \{a\}} \sum_{e \in E_{n,y}: s_1(e) \geq 2} (s_1(e) - 1) \\
&\leq |V_n| - |V_{n-1}| + \sum_{e \in E_{n-1}} (|e| - 1) + \sum_{y \in \mathcal{Y} \setminus \{a\}} \sum_{e \in E'_{n,y}} (|e| - 1) \\
&\leq |V_n| - |V_{n-1}| + d|V_{n-1}| + \sum_{y \in \mathcal{Y} \setminus \{a\}} (d-1)|V_{n,y}| \\
&= |V_n| - |V_{n-1}| + d|V_{n-1}| + (d-1)(|V_n| - |V_{n-1}|) \\
&= d|V_n|.
\end{aligned}$$

□

Proof of Lemma 3.8. Consider arbitrary $n \in \mathbb{N} \setminus \{1\}$ and $V_n \subseteq \mathcal{Y}^n$. By the definition of $\mathfrak{P}_a(V_n)$, we have $|V_n^\gamma| = |V_n|$. Let E_n denote the edge set of $\mathcal{G}(V_n)$ and $E_{n,\gamma}$ denote the edge set of $\mathcal{G}(V_n^\gamma)$. It suffices to prove that $\sum_{e \in E_n} (|e| - 1) \leq \sum_{e \in E_{n,\gamma}} (|e| - 1)$. Define

$$E_n^i := \{(e_{k,f}, k) \in E_n : k = i\} \text{ and } E_{n,\gamma}^i := \{(e_{k,f}, k) \in E_{n,\gamma} : k = i\}$$

for all $i \in [n]$. By the definition of V_n^γ , we have

$$\sum_{e \in E_n^n} (|e| - 1) = \sum_{e \in E_{n,\gamma}^n} (|e| - 1). \quad (23)$$

For any $i \in [n]$ and $f : [n] \setminus \{i\} \rightarrow \mathcal{Y}$, we define

$$\begin{aligned}
e_{i,f} &:= \{(y_1, \dots, y_n) \in V_n : y_k = f(k) \ \forall k \in [n] \setminus \{i\}\} \text{ and} \\
e_{i,f}^\gamma &:= \{(y_1, \dots, y_n) \in V_n^\gamma : y_k = f(k) \ \forall k \in [n] \setminus \{i\}\}
\end{aligned}$$

to distinguish edges in E_n and $E_{n,\gamma}$. For any $y \in \mathcal{Y}$, $i \in [n-1]$, and $f : [n-1] \setminus \{i\} \rightarrow \mathcal{Y}$, define $f_y : [n] \setminus \{i\} \rightarrow \mathcal{Y}$ such that $f_y|_{[n-1] \setminus \{i\}} = f$ and $f_y(n) = y$. Then, we have

$$\begin{aligned}
& \sum_{y \in \mathcal{Y} \setminus \{a\}} (|e_{i,f_y}| - 1) - \sum_{y \in \mathcal{Y} \setminus \{a\}} (|e_{i,f_y}^\gamma| - 1) \\
&\leq \mathbb{1}_{(e_{i,f_a}^\gamma, i) \in E_{n,\gamma}} (|e_{i,f_a}^\gamma| - 1) - \mathbb{1}_{(e_{i,f_a}, i) \in E_n} (|e_{i,f_a}| - 1)
\end{aligned}$$

which implies that

$$\sum_{i=1}^{n-1} \sum_{e \in E_n^i} (|e| - 1) \leq \sum_{i=1}^{n-1} \sum_{e \in E_{n,\gamma}^i} (|e| - 1).$$

and from (23),

$$\sum_{e \in E_n} (|e| - 1) \leq \sum_{e \in E_{n,\gamma}} (|e| - 1).$$

Thus, we can conclude that $\text{avgoutdeg}(\mathcal{G}(V_n^\gamma)) \geq \text{avgoutdeg}(\mathcal{G}(V_n))$ for any $\gamma \in \Gamma_{a,V_n}$. □

H Lemmas regarding graph dimension

In this section, we provide the technical lemmas on learning with finite graph dimension and bounding the graph dimension of certain classes. Those lemmas are used in the proof of Theorem 2.7.

H.1 Learning algorithm for classes with finite graph dimension

We first provide the definition of graph dimension.

Definition H.1 (Graph dimension). For $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and $n \in \mathbb{N}$, $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{X}^n$ is said to be **G-shattered** by $\mathcal{H}$ if there exists $f : [n] \rightarrow \mathcal{Y}$ such that for any $\mathbf{i} \subseteq [n]$, there exists $g \in \mathcal{H}$ satisfying $g(x_i) = f(i)$ for all $i \in \mathbf{i}$ and $g(x_i) \neq f(i)$ for all $i \in [n] \setminus \mathbf{i}$. The **graph dimension** of $\mathcal{H}$, denoted as $\dim_G(\mathcal{H})$ is the maximum size of a G-shattered sequence.

Define $\text{Log} : [0, \infty) \rightarrow [1, \infty)$, $x \mapsto \log(x \vee e)$ where $x \vee e = \max\{x, e\}$. For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, $n \in [\dim_G(\mathcal{H})]$, and $\mathbf{x} = \{x_1, \dots, x_n\} \subseteq \mathcal{X}$ that is G-shattered by $\mathcal{H}$, there exists $f : [n] \rightarrow \mathcal{Y}$ such that for any $\mathbf{i} \subseteq [n]$, there exists $g \in \mathcal{H}$ satisfying $g(x_i) = f(i)$ for all $i \in \mathbf{i}$ and $g(x_i) \neq f(i)$ for all $i \in [n] \setminus \mathbf{i}$. Thus, we can define

$$\mathcal{H}(\mathbf{x}) := \left\{ h : [n] \rightarrow \{0, 1\}, i \mapsto \mathbb{1}_{\tilde{h}(x_i)=f(i)} \mid \tilde{h} \in \mathcal{H} \right\}.$$

For general $n \in \mathbb{N}$, $\mathbf{x} \subseteq \mathcal{X}$ with $|\mathbf{x}| = n$, and $f \in \mathcal{Y}^n$, we define

$$\begin{aligned} \mathcal{H}_f(\mathbf{x}) &:= \left\{ h : [n] \rightarrow \{0, 1\}, i \mapsto \mathbb{1}_{\tilde{h}(x_i)=f(i)} \mid \tilde{h} \in \mathcal{H} \right\} \text{ and} \\ \tau_{\mathcal{H}}(n) &:= \sup_{\mathbf{x} \subseteq \mathcal{X}: |\mathbf{x}|=n} \sup_{f \in \mathcal{Y}^n} |\mathcal{H}_f(\mathbf{x})|. \end{aligned}$$

Note that $\tau_{\mathcal{H}}(n) = |\mathcal{H}(\mathbf{x})| = 2^n$ for any $n \in [\dim_G(\mathcal{H})]$. We have the following lemma.

Lemma H.2. For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim_G(\mathcal{H}) = d$, we have $\tau_{\mathcal{H}}(n) \leq \sum_{i=0}^d \binom{n}{i}$. In particular, if $n \geq d$, then $\tau_{\mathcal{H}}(n) \leq (en/d)^d$.

Proof. We first prove by induction on n that for any $\mathbf{x} = \{x_1, \dots, x_n\} \subseteq \mathcal{X}$ and $f \in \mathcal{Y}^n$,

$$|\mathcal{H}_f(\mathbf{x})| \leq |\{\mathbf{x}' \subseteq \mathbf{x} : \mathcal{H} \text{ G-shatters } \mathbf{x}'\}|. \quad (24)$$

For $n = 1$, it is obviously that $|\mathcal{H}_f(\mathbf{x})| \leq 2$ and $|\{\mathbf{x}' \subseteq \mathbf{x} : \mathcal{H} \text{ G-shatters } \mathbf{x}'\}| \geq 1$. If $|\{\mathbf{x}' \subseteq \mathbf{x} : \mathcal{H} \text{ G-shatters } \mathbf{x}'\}| = 1$, then $\mathbf{x}$ is not G-shattered by $\mathcal{H}$, which implies that $|\mathcal{H}_f(\mathbf{x})| \leq 1$. Thus, (24) holds. Now, suppose that (24) holds for any $k < n$. Consider $\bar{\mathbf{x}} := \{x_2, \dots, x_n\}$,

$$\begin{aligned} Y^0 &:= \{(y_2, \dots, y_n) \in \{0, 1\}^{n-1} : (0, y_2, \dots, y_n) \in \mathcal{H}_f(\mathbf{x}) \text{ or } (1, y_2, \dots, y_n) \in \mathcal{H}_f(\mathbf{x})\}, \text{ and} \\ Y^1 &:= \{(y_2, \dots, y_n) \in \{0, 1\}^{n-1} : (0, y_2, \dots, y_n) \in \mathcal{H}_f(\mathbf{x}) \text{ and } (1, y_2, \dots, y_n) \in \mathcal{H}_f(\mathbf{x})\}. \end{aligned}$$

We have $|\mathcal{H}_f(\mathbf{x})| = |Y^0| + |Y^1|$ and $|Y^0| = |\mathcal{H}_f(\bar{\mathbf{x}})|$. Then, by the induction hypothesis, we have

$$|Y^0| \leq |\{\mathbf{x}' \subseteq \bar{\mathbf{x}} : \mathcal{H} \text{ G-shatters } \mathbf{x}'\}| = |\{\mathbf{x}' \subseteq \mathbf{x} : x_1 \notin \mathbf{x}' \text{ and } \mathcal{H} \text{ G-shatters } \mathbf{x}'\}|.$$

For any $y \in \mathcal{Y}$, define

$$\mathcal{H}^y := \{h \in \mathcal{H} : \exists h' \in \mathcal{H} \text{ s.t. } h|_{\bar{\mathbf{x}}} \text{ and } h'|_{\bar{\mathbf{x}}} \text{ differs only at } 1 \text{ and } y \in \{h(x_1), h'(x_1)\}\}.$$

Then, $\mathcal{H}^y$ G-shatters $\mathbf{x}' \subseteq \bar{\mathbf{x}}$ implies that $\mathcal{H}$ G-shatters $\mathbf{x}' \cup \{x_1\}$. We also have $Y^1 = \mathcal{H}_f^{f(1)}(\bar{\mathbf{x}})$. It follows from the induction hypothesis that

$$|Y^1| = |\mathcal{H}_f^{f(1)}(\bar{\mathbf{x}})| \leq |\{\mathbf{x}' \subseteq \bar{\mathbf{x}} : \mathcal{H}^{f(1)} \text{ G-shatters } \mathbf{x}'\}| \leq |\{\mathbf{x}' \subseteq \mathbf{x} : x_1 \in \mathbf{x}' \text{ and } \mathcal{H} \text{ G-shatters } \mathbf{x}'\}|.$$

In conclusion, we have

$$\begin{aligned} &|\mathcal{H}_f(\mathbf{x})| \\ &= |Y^0| + |Y^1| \\ &\leq |\{\mathbf{x}' \subseteq \mathbf{x} : x_1 \notin \mathbf{x}' \text{ and } \mathcal{H} \text{ G-shatters } \mathbf{x}'\}| + |\{\mathbf{x}' \subseteq \mathbf{x} : x_1 \in \mathbf{x}' \text{ and } \mathcal{H} \text{ G-shatters } \mathbf{x}'\}| \\ &= |\{\mathbf{x}' \subseteq \mathbf{x} : \mathcal{H} \text{ G-shatters } \mathbf{x}'\}|, \end{aligned}$$

which is exactly (24). By (24), we have

$$\tau_{\mathcal{H}}(n) \leq |\{\mathbf{x}' \subseteq \mathbf{x} : \mathcal{H} \text{ G-shatters } \mathbf{x}'\}| \leq \sum_{i=0}^d \binom{n}{i}.$$

□

For any (measurable) classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$, define

$$\text{ER}(h) := \{(x, y) \in \mathcal{X} \times \mathcal{Y} : h(x) \neq y\}.$$

Then, for any probability measure $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, we can define

$$\text{er}_{\mathcal{D}}(h) := \mathcal{D}(\text{ER}(h)) = \mathbb{P}_{(X,Y) \sim \mathcal{D}}(h(X) \neq Y).$$

Definition H.3. $S \subseteq \mathcal{X} \times \mathcal{Y}$ is said to be an ε -net ($\varepsilon \in (0, 1)$) for $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with respect to a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ if for any $h \in \mathcal{H}$,

$$\text{er}_{\mathcal{D}}(h) \geq \varepsilon \implies \text{ER}(h) \cap S = \{(x, y) \in S : h(x) \neq y\} \neq \emptyset.$$

For any integer $n \in \mathbb{N}$, set $\mathcal{Z}$, and $T = (z_1, \dots, z_n) \in \mathcal{Z}^n$, we say $z \in T$ if $z = z_i$ for some $i \in [n]$ and use $|T|$ to denote the length of the sequence T . For notational convenience, we use $\emptyset$ to also denote an empty sequence (a sequence of length 0). For any subset $E \subseteq \mathcal{Z}$, we use $T \cap E = E \cap T$ to denote the subsequence of T consisting of all elements in E , i.e., for $I := \{i \in [n] : z_i \in E\}$,

$$T \cap E = E \cap T = (z_i)_{i \in I}.$$

Then, we have $|T \cap E| = |E \cap T| = |I|$.

Proposition H.4. For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim_G(\mathcal{H}) = d$, any $\mathcal{H}$ -realizable distribution $\mathcal{D}$, any $\delta \in (0, 1]$, any $n \in \mathbb{N}$, and any ERM algorithm A , consider $S_n \sim \mathcal{D}^n$. With probability at least $1 - \delta$, we have

$$\text{er}_{\mathcal{D}}(A(S_n, \mathcal{H})) \leq \frac{2}{n} \left[d \vee \left(d \log_2 \left(\frac{2en}{d} \right) \right) + \log_2 \left(\frac{2}{\delta} \right) \right].$$

Proof. For any $n \geq 2$ and $\varepsilon \in [2/n, 1]$, define

$$B := \{S \in (\mathcal{X} \times \mathcal{Y})^n : \exists h \in \mathcal{H} \text{ s.t. } \text{er}_{\mathcal{D}}(h) \geq \varepsilon \text{ and } \text{ER}(h) \cap S = \emptyset\} \text{ and}$$

$$B' := \{(S, T) \in (\mathcal{X} \times \mathcal{Y})^{2n} : |S| = |T| = |n|, \exists h \in \mathcal{H} \text{ s.t.}$$

$$\text{er}_{\mathcal{D}}(h) \geq \varepsilon, \text{ER}(h) \cap S = \emptyset, \text{ and } |\text{ER}(h) \cap T| > \varepsilon n/2\}.$$

Let $(S, T) \sim \mathcal{D}^{2n}$ with $S, T \in (\mathcal{X} \times \mathcal{Y})^n$. Since $(S, T) \in B'$ implies that $S \in B$, we have

$$\mathbb{P}((S, T) \in B') = \mathbb{E}[\mathbb{1}_{(S,T) \in B'} \mathbb{1}_{S \in B}] = \mathbb{E}[\mathbb{1}_{S \in B} \mathbb{P}((S, T) \in B' | S)].$$

On $S \in B$, there exists $h \in \mathcal{H}$ such that $\text{er}_{\mathcal{D}}(h) \geq \varepsilon$ and $\text{ER}(h) \cap S = \emptyset$. Then, $|\text{ER}(h) \cap T| > \varepsilon n/2$ implies that $(S, T) \in B'$. It follows that

$$\mathbb{1}_{S \in B} \mathbb{P}((S, T) \in B' | S) \geq \mathbb{1}_{S \in B} \mathbb{P}(|\text{ER}(h) \cap T| > \varepsilon n/2 | S).$$

Since T is independent of S , h is determined by S , and $\mathcal{D}(\text{ER}(h)) = \text{er}_{\mathcal{D}}(h) \geq \varepsilon$ on $S \in B$, we know that on $S \in B$, $|\text{ER}(h) \cap T|$ follows the Binomial distribution $B(n, \text{er}_{\mathcal{D}}(h))$ conditional on S , and

$$\mathbb{1}_{S \in B} \mathbb{E}[|\text{ER}(h) \cap T| | S] = \text{er}_{\mathcal{D}}(h) n \mathbb{1}_{S \in B} \geq \varepsilon n \mathbb{1}_{S \in B}.$$

Thus, by Lemma H.8, since $n\varepsilon \geq 2$, we have

$$\mathbb{1}_{S \in B} \mathbb{P}(|\text{ER}(h) \cap T| \leq \varepsilon n/2 | S) \leq \mathbb{P}(|\text{ER}(h) \cap T| \leq \text{er}_{\mathcal{D}}(h) n/2 | S) \mathbb{1}_{S \in B} < \frac{1}{2} \mathbb{1}_{S \in B},$$

which implies that

$$\mathbb{P}((S, T) \in B') > \frac{1}{2} \mathbb{P}(S \in B).$$

By the definition of B' , we have

$$\begin{aligned}\mathbb{P}((S, T) \in B') &= \mathbb{E} \left[\sup_{h \in \mathcal{H}} \mathbb{1}_{\text{er}_{\mathcal{D}}(h) \geq \varepsilon} \mathbb{1}_{\text{ER}(h) \cap S = \emptyset} \mathbb{1}_{|\text{ER}(h) \cap T| > \varepsilon n/2} \right] \\ &= \mathbb{E} \left[\sup_{h \in \mathcal{H}} \mathbb{1}_{\text{er}_{\mathcal{D}}(h) \geq \varepsilon} \mathbb{1}_{\text{ER}(h) \cap S = \emptyset} \mathbb{1}_{|\text{ER}(h) \cap (S, T)| > \varepsilon n/2} \right] \\ &\leq \mathbb{E} \left[\sup_{h \in \mathcal{H}} \mathbb{1}_{\text{ER}(h) \cap S = \emptyset} \mathbb{1}_{|\text{ER}(h) \cap (S, T)| > \varepsilon n/2} \right].\end{aligned}$$

For any $m \in \mathbb{N}$, $\mathbf{r} = ((x_1, y_1), \dots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$, and $h \in \mathcal{Y}^{\mathcal{X}}$, we define

$$h_{\mathbf{r}} : [m] \rightarrow \{0, 1\}, \quad i \mapsto \mathbb{1}_{h(x_i) = y_i}$$

and $\mathcal{H}_{\mathbf{r}} := \{h_{\mathbf{r}} : h \in \mathcal{H}\}$. Note that $\mathcal{H}_{\mathbf{r}} = \mathcal{H}_{(y_1, \dots, y_m)}((x_1, \dots, x_m))$, which implies that

$$|\mathcal{H}_{\mathbf{r}}| \leq \tau_{\mathcal{H}}(m). \quad (25)$$

For any $k \in [m]$ and $1 \leq i_1 < \dots < i_k \leq m$, we use $\mathbf{r}_{(i_1, \dots, i_k)}$ to denote a permutation of $\mathbf{r}$ where (x_j, y_j) appears in the i_j -th position for all $j \in [k]$, specifically, $\mathbf{r}_{\{i_1, \dots, i_k\}} = (x_{\sigma(i)}, y_{\sigma(i)})_{i \in [n]}$ where $\sigma(i_j) := j$ for all $j \in [k]$ and $(\sigma(l))_{l \in [n] \setminus \{i_1, \dots, i_k\}} := (k+1, \dots, m)$. Then, for any $\mathbf{i} \subseteq [2n]$ with $|\mathbf{i}| = n$, we have

$$\begin{aligned}\sup_{h \in \mathcal{H}} \mathbb{1}_{\text{ER}(h) \cap S = \emptyset} \mathbb{1}_{|\text{ER}(h) \cap (S, T)| > \varepsilon n/2} &\leq \sum_{h \in \mathcal{H}_{(S, T)}} \mathbb{1}_{h(i)=1, \forall i \in [n]} \mathbb{1}_{\sum_{i \in [2n]} h(i) < (2-\varepsilon/2)n} \\ &= \sum_{h \in \mathcal{H}_{(S, T)_{\mathbf{i}}}} \mathbb{1}_{h(i)=1, \forall i \in \mathbf{i}} \mathbb{1}_{\sum_{i \in [2n]} h(i) < (2-\varepsilon/2)n}.\end{aligned}$$

Since $(S, T) \sim \mathcal{D}^{2n}$, we also have

$$\mathbb{E} \left[\sum_{h \in \mathcal{H}_{(S, T)_{\mathbf{i}}}} \mathbb{1}_{h(i)=1, \forall i \in \mathbf{i}} \mathbb{1}_{\sum_{i \in [2n]} h(i) < (2-\varepsilon/2)n} \right] = \mathbb{E} \left[\sum_{h \in \mathcal{H}_{(S, T)}} \mathbb{1}_{h(i)=1, \forall i \in \mathbf{i}} \mathbb{1}_{\sum_{i \in [2n]} h(i) < (2-\varepsilon/2)n} \right].$$

Thus,

$$\begin{aligned}&\mathbb{E} \left[\sup_{h \in \mathcal{H}} \mathbb{1}_{\text{ER}(h) \cap S = \emptyset} \mathbb{1}_{|\text{ER}(h) \cap (S, T)| > \varepsilon n/2} \right] \\ &\leq \frac{1}{\binom{2n}{n}} \sum_{\mathbf{i} \subseteq [2n]: |\mathbf{i}| = n} \mathbb{E} \left[\sum_{h \in \mathcal{H}_{(S, T)}} \mathbb{1}_{h(i)=1, \forall i \in \mathbf{i}} \mathbb{1}_{\sum_{i \in [2n]} h(i) < (2-\varepsilon/2)n} \right] \\ &= \mathbb{E} \left[\sum_{h \in \mathcal{H}_{(S, T)}} \mathbb{1}_{\sum_{i \in [2n]} h(i) < (2-\varepsilon/2)n} \frac{1}{\binom{2n}{n}} \sum_{\mathbf{i} \subseteq [2n]: |\mathbf{i}| = n} \mathbb{1}_{h(i)=1, \forall i \in \mathbf{i}} \right] \\ &\leq \mathbb{E} \left[\sum_{h \in \mathcal{H}_{(S, T)}} \mathbb{1}_{\sum_{i \in [2n]} h(i) < (2-\varepsilon/2)n} \frac{\binom{\lfloor (2-\varepsilon/2)n \rfloor}{n}}{\binom{2n}{n}} \right] \\ &\leq 2^{-\varepsilon n/2} \mathbb{E} \left[\sum_{h \in \mathcal{H}_{(S, T)}} \mathbb{1}_{\sum_{i \in [2n]} h(i) < (2-\varepsilon/2)n} \right] \quad (26) \\ &\leq 2^{-\varepsilon n/2} \mathbb{E} [|\mathcal{H}_{(S, T)}|] \\ &\leq 2^{-\varepsilon n/2} \tau_{\mathcal{H}}(2n), \quad (27)\end{aligned}$$

where (26) follows from Lemma H.7 and (27) follows from (25). Finally, we have proved that

$$\mathcal{D}^n(B) = \mathbb{P}(S \in B) < 2\mathbb{P}((S, T) \in B') \leq 2\tau_{\mathcal{H}}(2n)2^{-\varepsilon n/2}. \quad (28)$$

Since $\mathcal{D}$ is $\mathcal{H}$ -realizable and A is an ERM algorithm, we must have $A(S_n, \mathcal{H}) \in \mathcal{H}$ and

$$\text{ER}(A(S_n, \mathcal{H})) \cap S_n = \emptyset$$

almost surely. Moreover, by the definition of B , if $S_n \notin B$, then $\text{ER}(A(S_n, \mathcal{H})) \cap S_n = \emptyset$ implies that $\text{er}_{\mathcal{D}}(A(S_n, \mathcal{H})) < \varepsilon$. Thus, we have

$$\mathbb{P}(\text{er}_{\mathcal{D}}(A(S_n, \mathcal{H})) < \varepsilon) \geq \mathbb{P}(S_n \notin B) = 1 - \mathcal{D}^n(B).$$

Solving $2\tau_{\mathcal{H}}(2n)2^{-\varepsilon n/2} = \delta$, we get

$$\varepsilon = 1 \wedge \left\lceil \frac{2}{n} \left(\log_2(\tau_{\mathcal{H}}(2n)) + \log_2\left(\frac{2}{\delta}\right) \right) \right\rceil \leq \frac{2}{n} \left[d \vee \left(d \log_2\left(\frac{2en}{d}\right) \right) + \log_2\left(\frac{2}{\delta}\right) \right]$$

where the last inequality follows from Lemma H.2. Note that $1 \wedge \left\lceil \frac{2}{n} (\log_2(\tau_{\mathcal{H}}(2n)) + \log_2(\frac{2}{\delta})) \right\rceil \geq \frac{2}{n}$, which implies that the above choice of ε is legitimate. Applying (28), we can conclude that with probability at least $1 - \delta$,

$$\text{er}_{\mathcal{D}}(A(S_n, \mathcal{H})) < \frac{2}{n} \left[d \vee \left(d \log_2\left(\frac{2en}{d}\right) \right) + \log_2\left(\frac{2}{\delta}\right) \right].$$

□

Proposition H.5. *There exists a learning algorithm A_G such that for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim_G(\mathcal{H}) = d$, any $\mathcal{H}$ -realizable distribution $\mathcal{D}$, any $\delta \in (0, 1]$, and any $n \in \mathbb{N}$, given $S_n \sim \mathcal{D}^n$, it holds with probability at least $1 - \delta$ that*

$$\text{er}_{\mathcal{D}}(A_G(S_n, \mathcal{H})) = O\left(\frac{1}{n} \left(d + \text{Log}\left(\frac{1}{\delta}\right) \right)\right). \quad (29)$$

Proof. The algorithm A_G is the algorithm $S_n \mapsto \text{Majority}(\text{ERM}_{\mathcal{H}}(\mathbb{A}(S_n; \emptyset)))$ defined in Hanneke [2016], where $\text{ERM}_{\mathcal{H}}$ denotes an ERM algorithm on the concept class $\mathcal{H}$. Applying the error rate of ERM algorithms proved in Proposition H.4 in the proof of Hanneke [2016, Theorem 2], we establish (29). □

The above proposition immediately implies the following corollary on the expected error rate of the learning algorithm A_G .

Corollary H.6. *There exists a learning algorithm A_G such that for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\dim_G(\mathcal{H}) = d$, any $\mathcal{H}$ -realizable distribution $\mathcal{D}$, and any $n \in \mathbb{N}$ it holds that*

$$\mathbb{E}_{S_n \sim \mathcal{D}^n} [\text{er}_{\mathcal{D}}(A_G(S_n, \mathcal{H}))] = \mathbb{E}_{((S_n, (X, Y)) \sim \mathcal{D}^{n+1}} [A_G(S_n, \mathcal{H})(X) \neq Y] = O\left(\frac{d}{n}\right). \quad (30)$$

Proof. According to Proposition H.5, there exists some constant $C > 0$ such that for any $\delta \in (0, 1]$, it holds with probability at least $1 - \delta$ that

$$\text{er}_{\mathcal{D}}(A_G(S_n, \mathcal{H})) \leq \frac{C}{n} \left(d + \log\left(\frac{1}{\delta} \vee e\right) \right),$$

which implies that for any $t \geq \frac{C(d+1)}{n}$,

$$\mathbb{P}(\text{er}_{\mathcal{D}}(A_G(S_n, \mathcal{H})) > t) \leq e^{-\frac{nt}{C} + d}.$$

Since $\text{er}_{\mathcal{D}}(A_G(S_n, \mathcal{H}))$ is nonnegative, we have

$$\begin{aligned} \mathbb{E}[\text{er}_{\mathcal{D}}(A_G(S_n, \mathcal{H}))] &= \int_0^\infty \mathbb{P}(\text{er}_{\mathcal{D}}(A_G(S_n, \mathcal{H})) > t) dt \\ &\leq \frac{C(d+1)}{n} + \int_{\frac{C(d+1)}{n}}^\infty e^{-\frac{nt}{C} + d} dt \\ &= \frac{C(d+1 + e^{-1})}{n} \\ &= O\left(\frac{d}{n}\right). \end{aligned}$$

□

Lemma H.7. For any $n \in \mathbb{N}$ and $m \in \mathbb{N} \cap [n, 2n]$, we have

$$\frac{\binom{m}{n}}{\binom{2n}{n}} \leq 2^{m-2n}. \quad (31)$$

Proof. Note that

$$\frac{\binom{m}{n}}{\binom{2n}{n}} = \frac{m(m-1) \cdots (m-n+1)}{2n(2n-1) \cdots (n+1)}.$$

We prove by induction on m . When $m = 2n$, we have

$$\frac{\binom{m}{n}}{\binom{2n}{n}} = 1 = 2^{m-2n}.$$

Suppose that (31) holds for some $m \in \mathbb{N} \cap [n+1, 2n]$. Then, we have

$$\frac{\binom{m-1}{n}}{\binom{2n}{n}} = \frac{m-n}{m} \frac{\binom{m}{n}}{\binom{2n}{n}} \leq \frac{1}{2} \cdot 2^{m-2n} = 2^{m-1-2n}.$$

Thus, by induction, (31) holds for any $m \in \mathbb{N} \cap [n, 2n]$. $\square$

Lemma H.8. For $X \sim B(n, p)$, if $np \geq 2$, then

$$\mathbb{P}(X \leq np/2) < 1/2.$$

Proof. If $np > 8$, since $\mathbb{E}[X] = np$, by the multiplicative Chernoff bound, we have

$$\mathbb{P}(X \leq np/2) \leq e^{-np/8} < e^{-1} \leq 1/2.$$

For $2 \leq np \leq 8$, we have

$$\mathbb{P}(X \leq np/2) = \sum_{i=0}^{\lfloor np/2 \rfloor} \binom{n}{i} p^i (1-p)^{n-i}.$$

For $6 \leq pn \leq 8$, we have $6/n \leq p \leq 8/n$ and $n \geq 6$. Thus,

$$\mathbb{P}(X \leq np/2) = \sum_{i=0}^3 \binom{n}{i} p^i (1-p)^{n-i}.$$

Consider

$$f_3(x, p) := \log(x(x-1)(x-2)) + (x-3) \log(1-p), \quad x \geq 6, \quad 1 \geq p \geq 6/x.$$

Fixing $p \in (0, 1]$, we have $x \geq 6/p$ and for $x \geq 6/p$,

$$\begin{aligned} \frac{\partial}{\partial x} f_3(x, p) &= \frac{1}{x} + \frac{1}{x-1} + \frac{1}{x-2} + \log(1-p) \leq \frac{1}{x} + \frac{1}{x-1} + \frac{1}{x-2} - p \\ &\leq \frac{p}{6} + \frac{p}{6-p} + \frac{p}{6-2p} - p < 0. \end{aligned}$$

Thus, fixing $p \in (0, 1]$, $f_3(\cdot, p)$ is a decreasing function on $[6/p, \infty)$. Therefore, we have

$$f_3(x, p) \leq f_3(6/p, p),$$

which implies that

$$g_3(x, p) \leq g_3(6/p, p)$$

for

$$g_3(x, p) := \frac{x(x-1)(x-2)}{6} p^3 (1-p)^{x-3}, \quad p \in (0, 1], \quad x \geq 6/p.$$

Since $\log z - z + 1 \leq 0$ for $z \in (0, 1]$ and $g'(z) = \log z - z + 1$ for $g(z) := z \log(z) + 1 - z - \frac{1}{2}(1-z)^2$ defined on $z \in (0, 1]$, we have

$$0 = g(1) \leq g(z) \leq \lim_{z \rightarrow 0_+} g(z) = 1/2$$

for $z \in (0, 1]$. Thus, we have

$$\log(z) + \frac{(1-z)(1-\frac{1-z}{2})}{z} \geq 0.$$

Plugging in $z = 1 - 6/t$ for $t > 6$, we have

$$\log(1 - 6/t) + \frac{6/t(1 - 3/t)}{1 - 6/t} \geq 0.$$

Then, defining

$$f(t) := \log(t) + \log(t-1) + \log(t-2) - 3\log(t) + (t-3)\log(1-6/t), \quad t > 6,$$

we have

$$f'(t) = \frac{1}{t} + \frac{1}{t-1} + \frac{1}{t} - \frac{3}{t} + \log(1-6/t) + \frac{6/t(1-3/t)}{1-6/t} > 0,$$

which implies that $f(t)$ increases with t for $t > 6$. Since

$$g_3(6/p, p) = \frac{6/p(6/p-1)(6/p-2)}{6} p^3(1-p)^{6/p-3}$$

and

$$e^{f(t)} = t(t-1)(t-2)t^{-3}(1-6/t)^{t-3},$$

we know that $g_3(6/p, p)$ decreases with $p \in (0, 1]$. Thus,

$$f_3(6/p, p) \leq \lim_{p \rightarrow 0_+} g_3(6/p, p) = \lim_{t \rightarrow \infty} \frac{t(t-1)(t-2)}{6} (6/t)^3(1-6/t)^{t-3} = 36e^{-6}.$$

Following the above steps, for

$$g_2(x, p) := \frac{x(x-1)}{2} p^2(1-p)^{x-2},$$

$$g_1(x, p) := xp(1-p)^{x-1}, \text{ and}$$

$$g_0(x, p) := (1-p)^x,$$

$$\text{where } p \in (0, 1], \quad x \geq 6/p,$$

it is easy to prove that for any $i = 0, 1, 2$, and $p \in (0, 1]$,

$$g_i(x, p) \leq g_i(6/p, p) \leq \lim_{p \rightarrow 0_+} g_i(6/p, p).$$

Specifically, we have

$$g_2(x, p) \leq \lim_{t \rightarrow \infty} \frac{t(t-1)}{2} (6/t)^2(1-6/t)^{t-2} = 18e^{-6},$$

$$g_1(x, p) \leq \lim_{t \rightarrow \infty} t(6/t)(1-6/t)^{t-1} = 6e^{-6}, \text{ and}$$

$$g_0(x, p) \leq \lim_{t \rightarrow \infty} (1-6/t)^t = e^{-6}.$$

Then, we can conclude that

$$\mathbb{P}(X \leq np/2) \leq \sum_{i=0}^3 \sup_{p \in (0, 1], x \geq 6/p} g_i(x, p) = (36 + 18 + 6 + 1)e^{-6} < \frac{1}{2}.$$

Next, we consider the regime that $4 \leq np < 6$. Now, we have

$$\mathbb{P}(X \leq np/2) = \sum_{i=0}^2 \binom{n}{i} p^i (1-p)^{n-i} \leq \sum_{i=0}^2 \sup_{p \in (0,1], x \geq 4/n} g_i(x, p).$$

Following the procedures in the previous case, it is not hard to verify that for $i = 0, 1, 2$,

$$\sup_{p \in (0,1], x \geq 4/n} g_i(x, p) = \lim_{p \rightarrow 0} g_i(4/p, p) = \lim_{t \rightarrow \infty} \frac{4^i}{i!} (1 - 4/t)^{t-i} = \frac{4^i}{i!} e^{-4}.$$

It implies that

$$\mathbb{P}(X \leq np/2) = \sum_{i=0}^2 \binom{n}{i} p^i (1-p)^{n-i} \leq \sum_{i=0}^2 \frac{4^i}{i!} e^{-4} < \frac{1}{2}.$$

Finally, we consider the regime that $2 \leq np < 4$. Now, we have

$$\mathbb{P}(X \leq np/2) = (1-p)^n + np(1-p)^{n-1} \leq \sum_{i=0}^1 \sup_{p \in (0,1], x \geq 2/n} g_i(x, p).$$

Following the previous procedures, it is not hard to verify that for $i = 0, 1$,

$$\sup_{p \in (0,1], x \geq 2/n} g_i(x, p) = \lim_{p \rightarrow 0} g_i(2/p, p) = \lim_{t \rightarrow \infty} \frac{2^i}{i!} (1 - 2/t)^{t-i} = \frac{2^i}{i!} e^{-2}.$$

It implies that

$$\mathbb{P}(X \leq np/2) \leq \sum_{i=0}^1 \frac{2^i}{i!} e^{-2} < \frac{1}{2}.$$

In conclusion, if $np \geq 2$, then

$$\mathbb{P}(X \leq np/2) < \frac{1}{2}.$$

□

H.2 Bounding the graph dimension

For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ of DS dimension $\dim(\mathcal{H}) = d < \infty$, sequence $S = (x_1, \dots, x_n) \in \mathcal{X}^n$, menu $\mu : \mathcal{X} \rightarrow \{Y \in \mathcal{Y} : |Y| \leq p\}$ of size $p \in \mathbb{N}$ with $n \in \mathbb{N}$, and $d' \in [n]$, define

$$\begin{aligned} \mathcal{H}_{S,\mu,d'} &:= \{h|_S : h \in \mathcal{H}, |\{i \in [n] : h(x_i) \notin \mu(x_i)\}| \leq d'\}, \\ \mathcal{H}_{S,\mu,\mathbf{i}} &:= \{h|_S : h \in \mathcal{H}, \{i \in [n] : h(x_i) \notin \mu(x_i)\} \subseteq \mathbf{i}\}, \text{ and} \\ \mathcal{H}'_{S,\mu,\mathbf{i}} &:= \{h : [n] \setminus \mathbf{i} \rightarrow \mathcal{Y}, i \mapsto \tilde{h}(i) \mid \tilde{h} \in \mathcal{H}_{S,\mu,\mathbf{i}}\} \end{aligned}$$

for all $\mathbf{i} \in \mathcal{2}_{d'}^{[n]}$ which denotes the collection of all subsets of $[n]$ of size d' . We have the following lemma.

Lemma H.9. $\dim_G(\mathcal{H}_{S,\mu,d'}) \leq (2 \log_2(e) + 4)(5d \log_2(p) + 2d')$ for any $d' \in [n]$.

Proof. For any $\mathbf{i} \in \mathcal{2}_{d'}^{[n]}$, by [Bendavid et al. \[1995\]](#), [Daniely and Shalev-Shwartz \[2014\]](#), we have

$$\dim_G(\mathcal{H}'_{S,\mu,\mathbf{i}}) \leq 5 \log_2(p) \dim_N(\mathcal{H}'_{S,\mu,\mathbf{i}}) \leq 5 \log_2(p) \dim(\mathcal{H}) = 5d \log_2(p).$$

For any $\mathbf{j} \subseteq [n]$ that is G-shattered by $\mathcal{H}_{S,\mu,\mathbf{i}}$, define $\mathbf{j}' := \mathbf{j} \setminus \mathbf{i}$. We have that $|\mathbf{j}'| \geq |\mathbf{j}| - d'$ and $\mathbf{j}'$ is G-shattered by $\mathcal{H}'_{S,\mu,\mathbf{i}}$, which immediately implies that $|\mathbf{j}'| \leq \dim_G(\mathcal{H}'_{S,\mu,\mathbf{i}}) \leq 5d \log_2(p)$ and $|\mathbf{j}| \leq |\mathbf{j}'| + d' \leq 5d \log_2(p) + d'$. It follows that

$$\dim_G(\mathcal{H}_{S,\mu,\mathbf{i}}) \leq 5d \log_2(p) + d' =: d_1.$$

For any $m \in \mathbb{N}$ and $\mathbf{j} = \{j_1, \dots, j_m\} \subseteq [n]$ that is G -shattered by $\mathcal{H}_{S,\mu,d'}$, there exists $f : [m] \rightarrow \mathcal{Y}$ such that for any $K \subseteq [m]$, there exists $g \in \mathcal{H}_{S,\mu,d'}$ satisfying $g(j_k) = f(k)$ for all $k \in K$ and $g(j_k) \neq f(k)$ for all $k \in [m] \setminus K$. It follows that $|\mathcal{H}_{S,\mu,d'}(\mathbf{j})| = 2^m$. If $m > d'$, by Lemma H.2, we have

$$2^m = |\mathcal{H}_{S,\mu,d'}(\mathbf{j})| \leq \sum_{\mathbf{i} \in 2_{d'}^{\mathbf{j}}} |(\mathcal{H}_{S,\mu,\mathbf{i}})_f(\mathbf{j})| \leq \sum_{\mathbf{i} \in 2_{d'}^{\mathbf{j}}} \tau_{\mathcal{H}_{S,\mu,\mathbf{i}}}(m) \leq \binom{m}{d'} \left(2^{d_1} \vee \left(\frac{em}{d_1} \right)^{d_1} \right),$$

which implies that

$$m \leq d_1 \left(1 \vee \log_2 \left(\frac{em}{d_1} \right) \right) + \log_2 \binom{m}{d'} \leq d_1 \left(1 \vee \log_2 \left(\frac{em}{d_1} \right) \right) + d' \log_2 \left(\frac{em}{d'} \right)$$

By Lemma H.11, we have

$$m \leq (2 \log_2(e) + 4)(d_1 + d') = (2 \log_2(e) + 4)(5d \log_2(p) + 2d'),$$

which implies that

$$\dim_G(\mathcal{H}_{S,\mu,d'}) \leq (2 \log_2(e) + 4)(5d \log_2(p) + 2d').$$

□

Lemma H.10. *If $x > 0$ satisfies $x \leq a \log_2(x/a) + b$ for some $a, b > 0$, then, we have $x \leq 2a + 2b$.*

Proof. Define $f(x) := x - a \log_2(x/a) - b$ for $x > 0$. Then, we have $f'(x) = 1 - \frac{a}{x \log(2)}$, which implies that f decreases with x for $x \in (0, a/\log(2))$ and increases with x for $x > a/\log(2)$. Since $2a + 2b > a/\log(2)$, it suffices to prove that $f(2a + 2b) \geq 0$. Indeed,

$$f(2a + 2b) = a + b - a \log_2((a + b)/a) = a((a + b)/a - \log_2((a + b)/a)) \geq 0.$$

□

Lemma H.11. *If $x > 0$ satisfies $x \leq a \log_2(x/a) + b \log_2(x/b) + c$ for some $a, b, c > 0$, then, we have $x \leq 4a + 4b + 2c$.*

Proof. Since

$$\begin{aligned} x &\leq a \log_2(x/a) + b \log_2(x/b) + c \\ &= (a + b) \log_2 \frac{x}{a + b} + c - (a + b) \left[\frac{a}{a + b} \log_2 \frac{a}{a + b} + \frac{b}{a + b} \log_2 \frac{b}{a + b} \right] \\ &\leq (a + b) \log_2 \frac{x}{a + b} + c + a + b, \end{aligned}$$

by Lemma H.10, we have

$$x \leq 2(a + b) + 2(a + b + c) = 4a + 4b + 2c.$$

□

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: In our abstract and introduction, we make the main claims that this paper reduces the gap between the lower and upper bounds of the multiclass PAC sample complexity and propose two possible routes towards completely resolving the optimal sample complexity, which are the paper's major contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: As is shown in Theorem 1.9, though we have reduced a log factor, there is still an $O(\sqrt{d} \log(d) \log(d/\varepsilon))$ gap between our upper and lower bounds of the multiclass PAC sample complexity. We leave the development of sharper sample complexity upper bound in Open Question 1 and Open Question 2.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: For each of our theoretical results, we provide the full set of assumptions in its statement as well as a complete and correct proof in the appendices.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: The paper is fully theoretical and does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - 4.1 If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - 4.2 If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - 4.3 If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - 4.4 We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The paper is fully theoretical and does not include experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: The paper is fully theoretical and does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: The paper is fully theoretical and does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NA]

Justification: The paper is fully theoretical and does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no societal impact of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not include experiments and poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: The paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.