Proceedings of Machine Learning Research vol 336:1–22, 2026 39th Annual Conference on Learning Theory

Learning Conditional Averages

Marco Bressan MARCO.BRESSAN@UNIMI.IT
Università degli Studi di Milano, Italy

Nataly Brukhim NBRUKHIM@PRINCETON.EDU
Institute for Advanced Study, USA

Nicolò Cesa-Bianchi NICOLO.CESA-BIANCHI@UNIMI.IT
Università degli Studi di Milano, Italy

Emmanuel Esposito EMMANUEL@EMMANUELESPOSITO.IT
Università degli Studi di Milano, Italy

Yishay Mansour MANSOUR.YISHAY@GMAIL.COM
Tel Aviv University, Israel
Google Research

Shay Moran SMORAN@TECHNION.AC.IL
Technion, Israel
Google Research

Maximilian Thiessen MAXIMILIAN.THIESSEN@UNI-TUEBINGEN.DE
TU Wien, Austria
University of Tübingen and Tübingen AI Center, Germany

Editors: Steve Hanneke and Tor Lattimore

Abstract

We introduce the problem of learning *conditional averages* in the PAC framework. The learner receives a sample labeled by an unknown target concept from a known concept class, as in standard PAC learning. However, instead of learning the target concept itself, the goal is to predict, for each instance, the average label over its *neighborhood*—an arbitrary subset of points that contains the instance. In the degenerate case where all neighborhoods are singletons, the problem reduces exactly to classic PAC learning. More generally, it extends PAC learning to a setting that captures learning tasks arising in several domains, including explainability, fairness, and recommendation systems. Our main contribution is a complete characterization of when conditional averages are learnable, together with sample complexity bounds that are tight up to logarithmic factors. The characterization hinges on the joint finiteness of two novel combinatorial parameters, which depend on both the concept class and the neighborhood system, and are closely related to the independence number of the associated neighborhood graph.

Keywords: supervised learning, PAC learning, graphs

1. Introduction

We introduce the following problem of learning conditional averages over neighborhoods. We are given an arbitrary domain $\mathcal{X}$ and a concept class $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$. Each point $x \in \mathcal{X}$ in the domain has an associated *neighborhood* $N[x]$, an arbitrary subset of $\mathcal{X}$ containing x itself. For example, if $\mathcal{X}$ is a metric space, then $N[x]$ could be a ball centered on x. We can think of the neighborhoods as encoded by a directed graph $G = (\mathcal{X}, E)$, so that $(x, y) \in E$ whenever $y \in N[x]$. Now, the learner receives a labeled sample $S = (x_1, c(x_1)), \ldots, (x_m, c(x_m))$ from an unknown distribution $\mathcal{D}$, where $c \in \mathcal{C}$ is an unknown target concept. The goal of the learner is to predict good estimates of the *average* label in the neighborhood of x,

$$\bar{c}(x) = \mathbb{E}_{x' \sim \mathcal{D}}\big[c(x') \mid x' \in N[x]\big] \,. \tag{1}$$

More precisely, like in standard PAC learning, we want the learner to output a predictor that, with probability $1 - \delta$, has expected squared loss at most ε over $\mathcal{D}$.

This problem is interesting in that it captures tasks that appear in various settings related to fairness and explainability. In fairness we want to learn classifiers whose average predictions (say, whether a person is hired or not) are roughly equal across certain subgroups (Hardt et al., 2016; Rothblum and Yona, 2021), or check whether a given classifier is fair (Chugg et al., 2023; Cherian and Candès, 2024; Hsu et al., 2024). We can encode these subgroups as neighborhoods (e.g., cliques) in the graph G. In explainability, local explainers typically use neighborhoods around data points to describe the local behavior of a complex global classifier (Ribeiro et al., 2018). Both scenarios require good estimates of the average predictions in each neighborhood, i.e., to estimate the conditional probabilities in (potentially small) regions. A different type of applications can be envisioned in scenarios where classification is just an initial step in a pipeline of learning tasks where the final goal is to predict various aggregate statistics of the given population. For example, given a sample of people and their income, we might want to predict average incomes across certain demographic subgroups, such as age, gender, or location. Moreover, a certain level of anonymity might be required: while the machine learning system has access to the individual income data, users should only see the predicted averages. Furthermore, in recommendation tasks such as collaborative filtering (Mnih and Salakhutdinov, 2007), systems provide recommendations based only on partial knowledge of the user features. This corresponds to our learning problem of predicting the average label over neighborhoods, which here correspond to all users with the same values over a subset of features. Another related learning problem is *learning from label proportions*. An important instantiation of this setting (Kück and de Freitas, 2005; Busa-Fekete et al., 2023, 2025; Brahmbhatt et al., 2023; Li et al., 2024) can be viewed as complementary to ours: there, the learner only gets to see the average labels and must predict the true labels of the individual data points.

Our problem looks interesting from a technical perspective, too. To begin with, it encompasses and generalizes other standard learning tasks. For instance, when all the neighborhoods are singletons, $N[x] = \{x\}$, we recover standard supervised PAC learning. When instead all neighborhoods are the full domain, $N[x] = \mathcal{X}$, the problem becomes estimating the average label under $\mathcal{D}$. Interestingly, we can also show that our problem is at least as expressive as learning with partial concept classes (see Appendix D). On the other hand, the problem is interesting in that natural approaches and standard combinatorial parameters seem to fail. For example, estimating $\bar{c}$ via uniform convergence leads to suboptimal bounds and additional dependencies; see, e.g., Grunewalder (2018) and Balsubramani et al. (2019). Moreover, when learning conditional averages, neither the VC dimension of the concept class nor the VC dimension of the neighborhood system succeed in capturing

learnability. In fact, they appear very far from doing so: there are cases where learning with the full class $\mathcal{C} = \{0,1\}^{\mathcal{X}}$ is possible, and (perhaps more surprisingly) cases where $\bar{c}$ is not learnable even when the class consists of just a single known concept, $\mathcal{C} = \{c\}$. The reason is that the average labels $\bar{c}$ depend both on the concept c and the distribution $\mathcal{D}$, and this intertwinement makes things subtly different from other, more standard settings.

1.1. Main results

Our main result is a full combinatorial characterization of learnability of conditional averages, for any concept class $\mathcal{C}$ and neighborhood graph G, as well as bounds on the required sample complexity tight up to logarithmic factors. Interestingly, neither the class $\mathcal{C}$ nor the graph G alone provide any meaningful characterizations. We introduce two novel combinatorial parameters that depend on both $\mathcal{C}$ and G, and related to the independence number $\alpha(G)$ of G. The first one, $\alpha_1(G,\mathcal{C})$, is the cardinality of the largest independent set in G that is shattered by $\mathcal{C}$. In particular, $\alpha_1(G,\mathcal{C})$ specializes to the standard VC dimension of $\mathcal{C}$ when neighborhoods are singletons, i.e., G has no edges. The second parameter, $\alpha_2(G,\mathcal{C})$, is the cardinality of the largest independent set I for which there exists a concept $c \in \mathcal{C}$ such that every $x \in I$ has a neighbor in G with opposite c-label.

We show that the finiteness of both these parameters is necessary and sufficient for learning. Moreover, we prove lower and upper bounds on the sample complexity $m(\varepsilon,\delta)$ of any class $\mathcal{C}$ and graph G, of the following form (we omit the dependency of α_1, α_2 on $(G,\mathcal{C})$ for brevity):

$$\Omega\left(\frac{\alpha_1 + \alpha_2/\log\alpha_2 + \log(1/\delta)}{\varepsilon}\right) \leqslant m(\varepsilon,\delta) \leqslant \mathcal{O}\left(\frac{\alpha_1 + \alpha_2\log(1/\varepsilon)}{\varepsilon}\log\frac{1}{\delta}\right), \tag{2}$$

where ε is the accuracy (w.r.t. the square loss) and δ the confidence parameter.

Our learning algorithm is as follows: if a test point x is adjacent to other points in the training sample, we return the empirical average of their labels. Otherwise, we run the one-inclusion graph algorithm (Haussler et al., 1994) on all isolated points in the graph induced by the training set and the test point. This leads to an in-expectation guarantee, which we subsequently turn to a high-probability guarantee by amplifying the confidence through a median-based argument.

1.2. Further related work

One motivation to study this particular learning problem comes from a line of work on the theory of explainable machine learning. *Local explainers* are some of the main tools to study and explain the local behavior of a complex machine learning system. One well-known approach is that of *anchors* (Ribeiro et al., 2018): to explain the label of a point, a small region around the point is used such that most points in the region have the same label. Many similar such local explanation methods exist (Ribeiro et al., 2016; Ancona et al., 2018). This problem was formalized, among others, by Dasgupta et al. (2022). In a follow-up work, Bhattacharjee and von Luxburg (2024) studied the problem of *auditing* local explanations, i.e., verifying whether the given explanation—the local classifier—has high accuracy in the chosen region. Our results are directly applicable to this auditing problem in the case of constant local classifiers (i.e., anchors). We also provide further support for their negative result: they state that typical local explainers require extremely small regions (e.g., exponentially small in the dimension of $\mathcal{X} = \mathbb{R}^d$) to be accurate, which leads to a large sample complexity to audit them. This corresponds in our case to graphs with large or infinite

independence number $\alpha(G)$. For example, many small disjoint regions in $\mathbb{R}^d$ lead to a graph with large $\alpha(G)$ in our case.

Another related work is the smoothed analysis of PAC learning by Chandrasekaran et al. (2024). Instead of having to predict averaged labels, their goal is to output a regular binary classifier that, however, competes with the best possible loss averaged over Gaussian perturbations of the test point, similar to our averages over neighborhoods. While they provide a novel (smoothed) analysis of standard binary classification, we tackle a novel learning problem instead.

2. Learning average labels

We start with the graph-theoretic notation used in the rest of the paper. We denote by $G = (V, E)$ a simple directed graph with anti-parallel edges allowed. That is, for any distinct $u, v \in V$, both (u, v) and (v, u) may belong to E (but no parallel edges are allowed). We remark that V can be infinite, and it will also be convenient to think of E as a relation over V. The *out-neighborhood* of $x \in V$ is $N^+(v) = \{v' \in V \mid (v, v') \in E\}$, and the *in-neighborhood* is $N^-(v) = \{v' \in V \mid (v', v) \in E\}$. We may also use $N(v) = N^+(v)$ for brevity. The *closed* out-neighborhood of $v \in V$ is the set $N[v] = N^+(v) \cup \{v\}$; we often refer to it simply as the neighborhood of v. A subset $V' \subseteq V$ is independent in G if for every distinct $u, v \in V'$ we have $u \notin N^+(v) \cup N^-(v)$. The independence number $\alpha(G)$ of G is the cardinality of its largest independent set (note that we may have $\alpha(G) = \infty$).

We now define the setting of the learning problem. Let $\mathcal{X}$ be a non-empty instance space and $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ a concept class. We assume a fixed neighborhood structure over $\mathcal{X}$ encoded by a simple directed graph $G = (\mathcal{X}, E)$, with anti-parallel edges allowed. The learner does not need to know G in advance; it is sufficient that it can access G through an edge oracle, i.e., an oracle that given $(u, v) \in V^2$ tells whether $(u, v) \in E$. As in realizable PAC learning, the learner receives a labeled sample $S = \big((x_1, c(x_1)), \ldots, (x_m, c(x_m))\big)$ with each x_i i.i.d. from an unknown distribution $\mathcal{D}$ and where $c \in \mathcal{C}$ is the unknown target concept. In contrast to standard PAC learning, our goal is to predict the *average labels* over each (closed) neighborhood:[1]

$$\bar{c}(x) = \mathbb{E}_{x' \sim \mathcal{D}}\Big[c(x') \mid x' \in N[x]\Big], \tag{3}$$

where we suppress the dependence of $\bar{c}$ on both the underlying graph G and distribution $\mathcal{D}$ when it is clear from context. In particular, the goal is to design a learning rule $\mathcal{A}$ that given a sample S outputs a predictor $h : \mathcal{X} \to [0, 1]$ such that with probability at least $1 - \delta$,

$$L(h) = L_{\mathcal{D}}(h) = \mathbb{E}_{x \sim \mathcal{D}}\left[\big(h(x) - \bar{c}(x)\big)^2\right] \leqslant \varepsilon\,, \tag{4}$$

which corresponds to the risk of h with respect to the square loss. For convenience, we will sometimes use the loss notation $\ell_x(h) = \ell_{x,\mathcal{D}}(h) = \big(\bar{c}(x) - h(x)\big)^2$ for individual points x. We remark that other natural loss functions are possible; see further discussion in Section 5.

Observe that, although the labels determined by the concept c are binary, rather than a classification problem our learning problem is a regression task. We define it formally below.

Definition 1 ($(G, \mathcal{C})$-learner) *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ and let $G = (\mathcal{X}, E)$ be a directed graph. A learning rule $\mathcal{A}$ is a* $(G, \mathcal{C})$-learner *if there exists a sample complexity $m : (0,1)^2 \to \mathbb{N}$ such that for every*

1. We assume throughout that the directed edge relation of the graph is measurable, as well as any additional measurability assumptions that are required by the one-inclusion graph algorithm (Haussler et al., 1994).

distribution $\mathcal{D}$ over $\mathcal{X}$, every $c \in \mathcal{C}$, and every $\varepsilon, \delta \in (0,1)$, the following holds. When given a multiset S of $m \geqslant m(\varepsilon,\delta)$ i.i.d. examples generated by $\mathcal{D}$ and labeled by c, then the learning rule returns a predictor $h_S = \mathcal{A}(S)$ such that $L(h_S) \leqslant \varepsilon$ with probability at least $1-\delta$ over S.

We now define two combinatorial parameters that characterize the learnability of our problem. The first can be thought of as corresponding to a *shattered independent set*, and the second to a *bichromatic independent set*. Formally, these parameters are defined as follows.

Definition 2 (Parameters α_1 and α_2) *Let $G = (\mathcal{X}, E)$ be a directed graph, $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ a class, and $c \in \mathcal{C}$ a concept.*

- **Largest shattered IS.** *Denote by $\alpha_1(G,\mathcal{C})$ the size of a largest independent set $I \subseteq \mathcal{X}$ of G that is shattered by $\mathcal{C}$, that is, $|\{c \cap I : c \in \mathcal{C}\}| = 2^{|I|}$.*
- **Largest bichromatic IS.** *Denote by $\alpha_2(G,c)$ the size of a largest independent set $I \subseteq \mathcal{X}$ of G such that each vertex in I has a neighbor in G with opposite label with respect to c. Define $\alpha_2(G,\mathcal{C}) = \sup_{c\in\mathcal{C}} \alpha_2(G,c)$.*

We remark that we can equivalently define $\alpha_2(G,c)$ in the following way. Let $X_c \subseteq \mathcal{X}$ be the set of vertices x that have a neighbor $x' \in N(x)$ in G such that $c(x) \neq c(x')$. Denote by G_c the subgraph of G induced by X_c. Then, we have $\alpha_2(G,c) = \alpha(G_c)$.

Our main result is the following characterization of learnability.

Theorem 3 *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ and let $G = (\mathcal{X}, E)$. Then:*

$$\textit{There exists a } (G,\mathcal{C})\textit{-learner} \iff \alpha_1(G,\mathcal{C}) + \alpha_2(G,\mathcal{C}) < \infty\ .$$

3. Examples and special cases

Before turning to the proof of our main result, we discuss some special cases to illustrate the role of the two parameters α_1 and α_2. Recall that, in the degenerate case where G has no edges, the problem coincides with standard PAC learning. In particular, if G has no edges, then for every concept class $\mathcal{C}$ we have $\alpha_2(G,\mathcal{C}) = 0$, while $\alpha_1(G,\mathcal{C})$ becomes just the VC dimension of $\mathcal{C}$.

The following are two extreme examples of concept classes and corresponding corollaries of Theorem 3. In a nutshell, Corollaries 4 and 5 show that the VC dimension of $\mathcal{C}$ alone does not determine learnability in our model. The first case we consider is the full concept class $\mathcal{C} = \{0,1\}^{\mathcal{X}}$. Since any independent set in the graph G is shattered by $\mathcal{C}$, we have $\alpha_1(G,\mathcal{C}) = \alpha(G)$. Moreover, as $\alpha_2(G,\mathcal{C}) \leqslant \alpha(G)$ by definition, from Theorem 3 we obtain the following corollary.

Corollary 4 (Full class) *Let $G = (\mathcal{X}, E)$. There is a $(G, \{0,1\}^{\mathcal{X}})$-learner if and only if $\alpha(G) < \infty$.*

Thus, even on the full class $\mathcal{C} = \{0,1\}^{\mathcal{X}}$, one may still be able to learn conditional averages, depending on the graph G. For a concrete example, take G to be the full graph, so that $N[x] = \mathcal{X}$ for every $x \in \mathcal{X}$. In this case $\alpha(G) = 1$, so by Corollary 4 a $(\mathcal{C}, G)$-learner exists; and, indeed, in this case $\bar{c}(x) = \mathbb{E}_{x\sim D} c(x)$ for every x, so the problem is just learning the average label under $\mathcal{D}$.

The other extreme case is the class consisting of a single concept, i.e., $\mathcal{C} = \{c\}$ for some $c : \mathcal{X} \to \{0,1\}$. As in this case $\alpha_1(G,\{c\}) = 0$, Theorem 3 yields the next corollary.

Corollary 5 (Singleton class) *Let $G = (\mathcal{X}, E)$ and $c : \mathcal{X} \to \{0,1\}$. There exists a $(G, \{c\})$-learner if and only if $\alpha_2(G, c) < \infty$.*

Thus, learning conditional averages can be hard even when the learner knows the target concept c. The reason is that, even with c known, learning $\bar{c}$ requires one to estimate the average labels in each neighborhood, and those depend on the (unknown) distribution.

Special neighborhood graphs. We now turn to special cases of graphs that illustrate the role of the graph structure in our task. If $G = (\mathcal{X}, E)$ is a complete graph, that is, $N[x] = \mathcal{X}$ for all $x \in \mathcal{X}$, then our learning problem simply requires to estimate the average label $\mathbb{E}_{x\sim\mathcal{D}}[c(x)]$ under the distribution $\mathcal{D}$ and ground truth c. In this case, $\alpha_1(G, \mathcal{C}), \alpha_2(G, \mathcal{C}) \leqslant \alpha(G) = 1$ as there are no larger independent sets in G. The problem is learnable—independently of the complexity of $\mathcal{C}$—by simply predicting the empirical mean of labels using a sample of size $\mathcal{O}(1/\varepsilon)$.

Similarly, on tournament graphs, which are orientations of complete (undirected) graphs, learning is easy. By definition we again have $\alpha_1(G, \mathcal{C}), \alpha_2(G, \mathcal{C}) \leqslant \alpha(G) = 1$ and thus we only need a sample of size $\widetilde{\mathcal{O}}(1/\varepsilon)$ even for $\mathcal{C} = \{0,1\}^{\mathcal{X}}$. This is somewhat surprising, since unlike the complete graph case, neighborhoods in a tournament graph can overlap in complex ways. Specifically, viewing neighborhoods as a set system, the *VC dimension of the set system* in tournament graphs can be as large as $\log|\mathcal{X}|$, whereas in the complete graph case it is exactly 1. These examples show that neighborhood complexity does not determine learnability.

Independence of α_1 and α_2. Finally, we show that the two parameters α_1 and α_2 are in general not related to each other. We mentioned above that for the graph graph with no edges, $\alpha_1(G, \mathcal{C})$ equals the VC dimension of $\mathcal{C}$, while $\alpha_2(G, \mathcal{C}) = 0$. A slightly less trivial example is the following. Take k disjoint cliques and let the class $\mathcal{C}$ consists of all labelings that are monochromatic on each clique. Here $\alpha_1(G, \mathcal{C}) = k$, while $\alpha_2(G, \mathcal{C}) = 0$.

An example of a graph G with a *large* α_2 is a star graph: a tree with one root node and all other nodes as leaves. Let c be a concept that assigns the same label to all leaves, and the opposite label to the root. In this case $\alpha_2(G, c)$ will correspond to the number of leaves, while $\alpha_1(G, \{c\}) = 0$

4. Main results

In this section we describe our algorithmic approach to the conditional average learning problem, followed by the analysis of upper and lower bounds on the sample complexity of a $(G, \mathcal{C})$-learner.

4.1. Upper bound

We first present our learning algorithm, described in Algorithm 1 below. Then, in Theorem 6 we give a bound on the expected error of Algorithm 1, and in Theorem 10 we give the high-probability bound. The following Algorithm 1 relies on the classic One-Inclusion Graph (OIG) algorithm as a sub-routine (see Appendix B for the algorithm and further details).

Theorem 6 *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ and let $G = (\mathcal{X}, E)$ be a directed graph with $\alpha_1(G, \mathcal{C}) + \alpha_2(G, \mathcal{C}) < \infty$. Then, for every distribution $\mathcal{D}$ over $\mathcal{X}$, every $c \in \mathcal{C}$ and every $\varepsilon \in (0,1)$, if S is an i.i.d. sample over $\mathcal{D}$ of size $m \geqslant m(\varepsilon) = \mathcal{O}\left(\frac{\alpha_1(G,\mathcal{C})+\alpha_2(G,\mathcal{C})\log(1/\varepsilon)}{\varepsilon}\right)$ then $\mathbb{E}_{S\sim\mathcal{D}^m}[L(h_S)] \leqslant \varepsilon$, where $h_S = \mathcal{A}(S)$ is defined by applying Algorithm 1 over S and an input point x.*

Algorithm 1 Conditional Average Learning Algorithm for Graphs

Input: Training set $S = \{(x_i, y_i)\}_{i=1}^m$, test point x.

Let $\widehat{G}$ be the subgraph of G induced by $\{x_1, \ldots, x_m, x\}$.
if *x has a neighbor in $\widehat{G}$* **then**
  return the empirical fraction of neighbors with label 1 in $N_{\widehat{G}}[x]$.
else
  Let I be the set of all isolated nodes in $\widehat{G}$, i.e., nodes with no out-going edges.
  Run the OIG predictor (see Algorithm 2) on $(I, \mathcal{C}|_I)$ and **return** its prediction.

The proof of this result relies on the following two technical lemmas. The first lemma controls the loss of the predictor returned by Algorithm 1 over all points whose neighborhood has sufficiently large mass. In turn, we use this result to determine the sample size required to guarantee that such a loss is sufficiently small in expectation.

Lemma 7 *Let $\lambda \in (0, 1]$, $c : \mathcal{X} \to \{0, 1\}$, $G = (\mathcal{X}, E)$, $\mathcal{D}$ a distribution over $\mathcal{X}$, and S a random multiset of $m \in \mathbb{N}_+$ i.i.d. samples from $\mathcal{D}$ and labeled by c. Let h_S be the output of Algorithm 1 given input S. Then, all points $x \in \mathcal{X}$ with $\mathcal{D}(N[x]) \geqslant \lambda$ satisfy $\mathbb{E}_{S\sim\mathcal{D}^m}\left[\ell_x(h_S)\right] \leqslant \frac{7}{2m\lambda}$.*

The proof of Lemma 7 is in Appendix A. We remark that the only part where we explicitly use the squared loss is Lemma 7. The whole approach can easily be adapted to other loss functions, by simply modifying this particular aspect; see Section 5 for an example. The second lemma, while simple, is crucial in enabling our result. Lemma 8 bounds the total mass of points whose neighborhoods are light.

Lemma 8 (Light-neighborhood nodes are light) *Let $G = (V, E)$ be a (possibly infinite) directed graph and $\mathcal{D}$ a distribution over V such that E is measurable. Then, for any $\lambda \in [0, 1]$,*

$$\mathcal{D}\left(\{v : \mathcal{D}(N[v]) \leqslant \lambda\}\right) \leqslant 2\lambda \cdot \alpha(G) \ .$$

Proof Let V_λ denote the subset of nodes v of G for which $\mathcal{D}(N[v]) \leqslant \lambda$, and consider the induced subgraph over V_λ, denoted G_λ. We would like to bound $\mathcal{D}(V_\lambda)$ from above. To that end, we will iteratively remove nodes from V_λ and bound the overall mass of removed points. By Proposition 9, there must exist $v \in V_\lambda$ with $\mathcal{D}\left(N^+(v)\right) \geqslant \mathcal{D}\left(N^-(v)\right)$, where the neighborhoods N^+, N^- are with respect to G_λ. Remove all nodes in $N^+(v) \cup N^-(v)$ and v itself from V_λ. Observe that the overall mass of nodes we have removed is at most 2λ. We repeat this process until there are no more vertices left in V_λ. Let S denote the set of nodes v we picked in each round. Observe that S forms an independent set in G_λ, and so $|S| \leqslant \alpha(G_\lambda) \leqslant \alpha(G)$ bounds the total number of rounds in the process. Overall, we obtain that $\mathcal{D}(V_\lambda) \leqslant 2\lambda \cdot \alpha(G)$ as claimed. ■

We remark that in the undirected case, Lemma 8 is related to a weighted version of the Caro-Wei inequality (Caro, 1979; Wei, 1981);[2] our analogous result, which applies more generally to possibly uncountable domains and arbitrary (measurable) edge sets, might be of independent interest. Its proof relies on the following result.

2. Interestingly, the Caro-Wei inequality is a key result in the line of work on online learning with feedback graphs, where various weighted versions of it have been studied (Mannor and Shamir, 2011; Alon et al., 2015, 2017; Esposito et al., 2022; Eldowa et al., 2023).

Proposition 9 *Let $G = (V, E)$ be a (possibly infinite) directed graph and let $\mathcal{D}$ be a probability distribution on V such that the edge relation E is measurable. Then, there exists a vertex $v \in V$ such that $\mathcal{D}(N^+(v)) \geqslant \mathcal{D}(N^-(v))$.*

Proof We start by showing that we have $\mathbb{E}_{v\sim\mathcal{D}}[\mathcal{D}(N^+(v))] = \mathbb{E}_{v\sim\mathcal{D}}[\mathcal{D}(N^-(v))]$, which corresponds to a weighted version of the degree sum formula for directed graphs. Let x, x' be independent random variables distributed according to $\mathcal{D}$. Define the indicator random variable $I_E \coloneqq \mathbb{I}\{(x, x') \in E\}$. Conditioning on x, we have

$$\mathbb{E}[I_E \mid x] = \mathbb{P}\big((x, x') \in E \mid x\big) = \mathbb{P}\big(x' \in N^+(x) \mid x\big) = \mathcal{D}\big(N^+(x)\big) \,. \tag{5}$$

By the law of total expectation (which applies since the edge relation is measurable), we have $\mathbb{E}[I_E] = \mathbb{E}[\mathbb{E}[I_E \mid x]] = \mathbb{E}_{x\sim\mathcal{D}}\big[\mathcal{D}(N^+(x))\big]$. Similarly, conditioning on x', we have

$$\mathbb{E}[I_E \mid x'] = \mathbb{P}\big((x, x') \in E \mid x'\big) = \mathbb{P}\big(x \in N^-(x') \mid x'\big) = \mathcal{D}\big(N^-(x')\big) \,. \tag{6}$$

Again by the law of total expectation, $\mathbb{E}[I_E] = \mathbb{E}[\mathbb{E}[I_E \mid x']] = \mathbb{E}_{x\sim\mathcal{D}}\big[\mathcal{D}(N^-(x))\big]$. Thus, we have $\mathbb{E}_{v\sim\mathcal{D}}[\mathcal{D}(N^+(v))] = \mathbb{E}_{v\sim\mathcal{D}}[\mathcal{D}(N^-(v))]$. Next, if $\mathcal{D}(N^+(v)) < \mathcal{D}(N^-(v))$ held for all $v \in V$, then we would have $\mathbb{E}_{v\sim\mathcal{D}}[\mathcal{D}(N^+(v))] < \mathbb{E}_{v\sim\mathcal{D}}[\mathcal{D}(N^-(v))]$, in contradiction with the above. Hence, there must exist $v \in V$ such that $\mathcal{D}(N^+(v)) \geqslant \mathcal{D}(N^-(v))$. ■

Finally, we have all the main tools to prove the upper bound on the sample complexity for achieving the in-expectation guarantee on the loss.

Proof of Theorem 6 Let $c \in \mathcal{C}$ be the target concept. Let $\alpha_1 = \alpha_1(G, \mathcal{C})$ and $\alpha_2 = \alpha_2(G, \mathcal{C})$, and let $m \geqslant m_{\mathcal{A}}(\varepsilon)$ where the constants in the $\mathcal{O}(\cdot)$ notation are large enough. We consider a learner $\mathcal{A}$ that gets a training set S as input and outputs a predictor $h_S : \mathcal{X} \to [0, 1]$ by applying Algorithm 1 over S and the input point $x \in \mathcal{X}$. For any $c \in \mathcal{C}$ and distribution $\mathcal{D}$ over $\mathcal{X}$ denote by $\mathcal{D}_c$ the joint distribution over $(x, y) \in \mathcal{X} \times \{0, 1\}$ determined by $x \sim \mathcal{D}$ and $y = c(x)$. With some abuse of notation, we will omit c when it is clear from context. Then, note that rather than bounding the expected error of $\mathcal{A}$, we can instead analyze its leave-one-out error:

$$\mathbb{E}_{S\sim\mathcal{D}^m}[L(\mathcal{A}(S))] = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m, x\sim\mathcal{D}}[\ell_x(\mathcal{A}(S))] = \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^{m+1}}\Big[\mathop{\mathbb{E}}_{i\sim\mathcal{U}_{[m+1]}}\big[\ell_{x_i}(\mathcal{A}(S'_{-i}))\big]\Big] \,. \tag{7}$$

That is, we consider the following process: (i) a sample S' of $m + 1$ i.i.d. points is drawn from $\mathcal{D}^{m+1}$, (ii) an integer i is drawn uniformly from $[m + 1]$, (iii) the algorithm $\mathcal{A}$ is ran with $S = S'_{-i} = S' \backslash \{(x_i, c(x_i)\}$ as (labeled) training sample, and $x = x_i$ as test point, incurring error $\ell_x(h_S)$ where $h_S = \mathcal{A}(S) = \mathcal{A}(S'_{-i})$. Our goal is to show that: $\mathbb{E}_{S',i}[\ell_{x_i}(h_S)] \leqslant \varepsilon$, where it is understood that $S' \sim \mathcal{D}^{m+1}$ and i is uniform over $[m+1]$. We consider the following events (where $N = N_G^+$):

$$\mathcal{E}_1^{(i)} : N[x_i] \text{ is bichromatic,} \tag{8}$$

$$\mathcal{E}_2^{(i)} : N[x_i] \text{ is monochromatic and } x_i \notin I, \text{ and} \tag{9}$$

$$\mathcal{E}_3^{(i)} : N[x_i] \text{ is monochromatic and } x_i \in I. \tag{10}$$

One can check that the three events are disjoint and that $\mathcal{E}_1^{(i)} \cup \mathcal{E}_2^{(i)} \cup \mathcal{E}_3^{(i)}$ corresponds to the sample space. Thus, by the law of total expectation,

$$\mathbb{E}_{S',i}[\ell_{x_i}(h_S)] = \sum_{r=1}^{3} \mathbb{P}_{S',i}(\mathcal{E}_r^{(i)}) \cdot \mathbb{E}_{S',i}\Big[\ell_{x_i}(h_S) \,\Big|\, \mathcal{E}_r^{(i)}\Big] \,. \tag{11}$$

In what follows, we show that each term of the summation in the right-hand side is sufficiently small; precisely, the terms corresponding to $\mathcal{E}_1^{(i)}$ and $\mathcal{E}_3^{(i)}$ are each bounded from above by $\frac{\varepsilon}{2}$, while the one relative to $\mathcal{E}_2^{(i)}$ is equal to zero.

Loss under $\mathcal{E}_1^{(i)}$. The idea is to bound the error by simultaneously applying Lemmas 7 and 8. However, their respective guarantees have an inverse dependence on the range of possible values that $\mathcal{D}(N[x_i])$ can take, and thus a straightforward application of these two technical results can lead to a larger guarantee than desired, with a worse dependence on α_2 or m. We show this can be avoided by carefully defining a partitioning of the range of possible values for $\mathcal{D}(N[x_i])$ into a sufficiently small number of intervals determined by a geometric sequence.

Let $n = \lceil \log_2(m/\alpha_2) \rceil + 1$ and define $1 \geqslant b_1 \geqslant \cdots \geqslant b_n \geqslant 0$ to be a non-increasing sequence such that $b_j = 2^{-j+1}/\alpha_2$ for all $j \in [n]$. Note that this sequence satisfies $b_1 = 1/\alpha_2$ and $b_n \leqslant 1/m$. For every $j \in [n]$, denote by A_j the event that $\mathcal{D}(N[x_i]) \leqslant b_j$, and define $B_j = A_j \backslash A_{j+1}$. By a further application of the law of total expectation, and using the fact that $\ell_x(h) \in [0,1]$ for every $x \in \mathcal{X}$, we can bound the error conditioned on $\mathcal{E}_1^{(i)}$ as

$$\mathbb{E}_{S',i}\big[\ell_{x_i}(h_S) \mid \mathcal{E}_1^{(i)}\big] \leqslant \sum_{j=1}^{n-1} \mathbb{E}_{S',i}\big[\ell_{x_i}(h_S) \mid \mathcal{E}_1^{(i)}, B_j\big] \cdot \mathbb{P}_{S',i}(B_j \mid \mathcal{E}_1^{(i)}) \tag{12}$$

$$+ \mathbb{P}_{S',i}(A_n \mid \mathcal{E}_1^{(i)}) + \mathbb{E}_{S_i'}\big[\ell_{x_i}(h_S) \mid \mathcal{E}_1^{(i)}, \overline{A}_1\big] \,. \tag{13}$$

For each $j \in [n-1]$, by symmetrization and Lemma 7 we have that

$$\mathbb{E}_{S',i}\big[\ell_{x_i}(h_S) \mid \mathcal{E}_1^{(i)}, B_j\big] \tag{14}$$

$$= \mathbb{E}_{x\sim\mathcal{D}}\Big[\mathbb{E}_{S\sim\mathcal{D}^m}\big[\ell_x(h_S)\big] \mid N[x] \text{ is bichromatic}, \mathcal{D}(N[x]) \in (b_{j+1}, b_j]\Big] \tag{15}$$

$$\leqslant \sup_{x:\mathcal{D}(N[x])\geqslant b_{j+1}} \mathbb{E}_{S\sim\mathcal{D}^m}\big[\ell_x(h_S)\big] \leqslant \frac{7}{2b_{j+1}m} \,, \tag{16}$$

where we rely on the fact that S and x_i are independent, and both events $\mathcal{E}_1^{(i)}$ and B_j only affect x_i. By a similar reasoning, it also holds that $\mathbb{E}_{S',i}\big[\ell_{x_i}(h_S) \mid \mathcal{E}_1^{(i)}, \overline{A}_1\big] \leqslant \frac{7}{2b_1m}$. Moreover, we have that $\mathbb{P}_{S',i}(B_j \mid \mathcal{E}_1^{(i)}) \leqslant \mathbb{P}_{S',i}(A_j \mid \mathcal{E}_1^{(i)})$ as $B_j \subseteq A_j$. These observations allow us to show that

$$\mathbb{E}_{S',i}\big[\ell_{x_i}(h_S) \mid \mathcal{E}_1^{(i)}\big] \leqslant \frac{7}{2m} \sum_{j=1}^{n-1} \frac{\mathbb{P}_{S',i}(A_j \mid \mathcal{E}_1^{(i)})}{b_{j+1}} + \mathbb{P}_{S',i}(A_n \mid \mathcal{E}_1^{(i)}) + \frac{7}{2b_1m} \,, \tag{17}$$

and consequently that

$$\mathbb{P}_{S',i}(\mathcal{E}_1^{(i)})\mathbb{E}_{S',i}\big[\ell_{x_i}(h_S) \mid \mathcal{E}_1^{(i)}\big] \leqslant \frac{7}{2m} \sum_{j=1}^{n-1} \frac{\mathbb{P}_{S',i}(A_j)}{b_{j+1}} + \mathbb{P}_{S',i}(A_n) + \frac{7}{2b_1m} \tag{18}$$

$$\leqslant \frac{7\alpha_2}{m} \sum_{j=1}^{n-1} \frac{b_j}{b_{j+1}} + 2b_n\alpha_2 + \frac{7}{2b_1m} \tag{19}$$

$$\leqslant \frac{14\alpha_2}{m}(n-1) + \frac{2\alpha_2}{m} + \frac{7\alpha_2}{2m} \tag{20}$$

$$\leqslant \frac{2\alpha_2}{m}(7\log_2(m/\alpha_2) + 11) \,, \tag{21}$$

where the second step holds because $\mathbb{P}_{S',i}(A_j) \leqslant 2b_j\alpha_2$ for each $j \in [n]$, as we show next, by applying Lemma 8. Using Lemma 8 over the entire G would lead to a worse upper bound of $2b_j\alpha(G)$. Instead, we apply it to the subgraph G_c induced by all nodes with bichromatic neighborhoods $N[x]$ with respect to c. As remarked before (after Definition 2), $\alpha(G_c) = \alpha_2(G, c) \leqslant \alpha_2$.

Then, the right-hand side of Equation (21) is no larger than $\frac{\varepsilon}{2}$ given $m \geqslant \mathcal{O}\big(\frac{\alpha_2}{\varepsilon}\log(1/\varepsilon)\big)$.

Loss under $\mathcal{E}_2^{(i)}$. Here, the algorithm predicts the empirical average of the labels of $S \cap N[x_i]$, which is the true label $\bar{c}(x_i) = c(x_i)$ since $N[x_i]$ is monochromatic. Thus $\mathbb{E}_{S',i}\big[\ell_{x_i}(h) \mid \mathcal{E}_2^{(i)}\big] = 0$.

Loss under $\mathcal{E}_3^{(i)}$. By non-negativity of the loss $\ell_{x_i}(h)$, we have

$$\mathbb{P}_{S',i}(\mathcal{E}_3^{(i)}) \cdot \mathbb{E}_{S',i}\Big[\ell_{x_i}(h) \,|\, \mathcal{E}_3^{(i)}\Big] \leqslant \mathbb{P}_{S',i}(x_i \in I) \cdot \mathbb{E}_{S',i}\Big[\ell_{x_i}(h) \,|\, \mathcal{E}_3^{(i)}\Big] \,. \tag{22}$$

We now proceed to show that $\mathbb{P}_{S',i}(x_i \in I) \cdot \mathbb{E}_{S',i}\big[\ell_{x_i}(h) \mid \mathcal{E}_3^{(i)}\big] \leqslant \frac{\varepsilon}{2}$ for *every* sequence $S' \in \mathcal{X}^{m+1}$ with m large enough. First, since i is uniform over $[m+1]$, then

$$\mathbb{P}_{S',i}(x_i \in I) \cdot \mathbb{E}_{S',i}\Big[\ell_{x_i}(h) \,|\, \mathcal{E}_3^{(i)}\Big] = \frac{|I|}{m+1} \cdot \mathbb{E}_{S',i}\Big[\ell_{x_i}(h) \,|\, \mathcal{E}_3^{(i)}\Big] \,, \tag{23}$$

Note that in this case we have $\bar{c}(x_i) = c(x_i)$ as $N[x_i]$ is monochromatic. Thus, for all such x_i the learning task is a binary classification with the zero-one loss (instead of regression with the square loss).

Now, under $\mathcal{E}_3^{(i)}$ (as $x_i \in I$), the point x_i is distributed uniformly over I. Further, in this case the algorithm returns the OIG prediction for x_i. This together implies that $\mathbb{E}_{S',i}\big[\ell_{x_i}(h) \mid \mathcal{E}_3^{(i)}\big]$ is the expected leave-one-out error of the OIG predictor ran on I. As the vertices in I are isolated in $\widehat{G}$, I is an independent set in $\widehat{G}$, and so I is independent in G too. Therefore, the projection $\mathcal{C}_{|I}$ of $\mathcal{C}$ on I has VC dimension at most α_1, by definition of α_1. The OIG predictor (Haussler et al., 1994) thus yields $\mathbb{E}_{S',i}\big[\ell_{x_i}(h) \mid \mathcal{E}_3^{(i)}\big] \leqslant \frac{\alpha_1}{|I|}$. We remark that it may be that I contains x alone, in which case the OIG may predict arbitrarily. Overall, we have that Equation (23) satisfies

$$\mathbb{P}_{S',i}(x_i \in I) \cdot \mathbb{E}_{S',i}\Big[\ell_{x_i}(h) \,|\, \mathcal{E}_3^{(i)}\Big] \leqslant \frac{|I|}{m+1} \cdot \frac{\alpha_1}{|I|} = \frac{\alpha_1}{m+1} \,, \tag{24}$$

which, for $m \geqslant \frac{2\alpha_1}{\varepsilon}$, is at most $\frac{\varepsilon}{2}$. This concludes the proof. ■

As a final step, we adopt a confidence amplification approach to turn the in-expectation guarantee of Theorem 6 into a high-probability one.

Theorem 10 *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ and let $G = (\mathcal{X}, E)$ be a directed graph with $\alpha_1(G,\mathcal{C}) + \alpha_2(G,\mathcal{C}) < \infty$. There exists a $(G,\mathcal{C})$-learner with overall sample complexity*

$$m(\varepsilon,\delta) = \mathcal{O}\left(\frac{\alpha_1(G,\mathcal{C}) + \alpha_2(G,\mathcal{C})\log(1/\varepsilon)}{\varepsilon}\log(1/\delta)\right) .$$

Proof The proof follows by applying Theorem 6 to show that Algorithm 1 has a bounded expected error, and then applying Lemma 11 to obtain a high-probability guarantee. ■

The high-probability guarantee is obtained by using the following lemma. The main idea is to use Markov's inequality to obtain a guarantee with probability strictly larger than $1/2$ from the in-expectation one provided by Theorem 6, and then take the pointwise median over $\mathcal{O}(\log(1/\delta))$ independently obtained predictors from Algorithm 1 to amplify the success probability.

Lemma 11 *Let $\mathcal{A}$ be an algorithm that satisfies an expected error $\mathbb{E}_S[L(\mathcal{A}(S))] \leqslant \varepsilon$ with sample size $m(\varepsilon)$. There exists a $(G,\mathcal{C})$-learner $\mathcal{A}_{\mathrm{whp}}$ with oracle access to $\mathcal{A}$ and overall sample complexity $m(\varepsilon,\delta) = \mathcal{O}\big(m(\varepsilon/100)\log(1/\delta)\big)$.*

Proof Let $k \in \mathbb{N}$ and $\varepsilon' \in (0,1)$ to be determined later. Let S be an i.i.d. sample from $\mathcal{D}$ of size $k \cdot m(\varepsilon)$ and partition S into subsamples $S_1,\dots,S_k$ each of size $m(\varepsilon)$. Let $h_i = \mathcal{A}(S_i)$ for each $i \in [k]$ be the predictor satisfying $\mathbb{E}_{S_i}[L(h_i)] \leqslant \varepsilon'$. Denoting by A_i the event $L(h_i) > 10\varepsilon'$, we have by Markov's inequality $\mathbb{E}_{S_i}[\mathbb{I}\{A_i\}] = \mathbb{P}_{S_i}(A_i) \leqslant 1/10$. By a multiplicative Chernoff bound, there is an absolute constant $c > 0$ such that $\mathbb{P}_S\left(\sum_{i=1}^k \mathbb{I}\{A_i\} \geqslant k/5\right) \leqslant e^{-ck}$. The right-hand side of the latter is at most δ for $k = \lceil (1/c)\log(1/\delta)\rceil$. Thus, with probability at least $1-\delta$, we have $\frac{4}{5}k$ classifiers h_i satisfying $L(h_i) \leqslant 10\varepsilon'$. We now apply Lemma 15 from Appendix A and see that the pointwise median $h_{\mathrm{med}} = \mathrm{median}(h_1,\dots,h_k)$ satisfies $L(h_{\mathrm{med}}) \leqslant 70\varepsilon' \leqslant 100\varepsilon'$ with probability $1-\delta$. We define $\mathcal{A}_{\mathrm{whp}}$ as an algorithm that takes a sample S as described and returns h_{med}. The claim follows by choosing $\varepsilon' = \varepsilon/100$. ■

We note that directly applying the online-to-batch conversion of, e.g., Aden-Ali et al. (2023) to our setting seems not possible. They require a leave-one-out guarantee to hold for any sample, while we only satisfy such a guarantee in expectation.

4.2. Lower bounds

We prove a separate lower bound for each parameter $\alpha_1(G,\mathcal{C})$ and $\alpha_2(G,\mathcal{C})$, which together imply tightness of our algorithmic result from Theorem 10 up to log-factors.

Lemma 12 *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ and $G = (\mathcal{X},E)$. If $\alpha_1(G,\mathcal{C}) \geqslant 2$, then the sample complexity achievable by any $(G,\mathcal{C})$-learner is $m(\varepsilon,\delta) = \Omega\left(\frac{\alpha_1(G,\mathcal{C})+\log(1/\delta)}{\varepsilon}\right)$.*

Proof Let $I \subseteq \mathcal{X}$ with $|I| \geqslant 2$ be an independent set that is shattered by $\mathcal{C}$ and $c \in \mathcal{C}$ a target concept. As the neighborhoods of the vertices in I are disjoint, any distribution with support on I has $\overline{c}(x) = c(x)$ for all $x \in I$. Since I is shattered by $\mathcal{C}$, we can apply PAC lower bounds to get a $\Omega(|I|/\varepsilon)$ lower bound (Ehrenfeucht et al., 1989). Further, we get an additional $\Omega\big(\log(1/\delta)/\varepsilon\big)$ lower bound. ■

Before proving the second lower bound, we first provide some intuition. Take a set of pairwise disjoint edges $\{a_i,b_i\}$ on some node sets A and B, i.e., a matching. Assume that $c \in \mathcal{C}$ labels each node in A by 0 and each node in B by 1. Even if the algorithm knows c there can be uncertainty about the averaged labels $\overline{c}$. In particular, let the distribution $\mathcal{D}$ be uniform over A with $\mathcal{D}(A) = 1/2$ and uniform over an unknown subset $C \subseteq B$ of size $|C| = |B|/2$ with $\mathcal{D}(C) = 1/2$. Under this family of distributions (parameterized by the choice of C), we have $\overline{c}(a_i) \in \{0,1/2\}$. For each i, the learner can only distinguish between $\overline{c}(a_i) = 0$ and $\overline{c}(b_i) = 1/2$ by receiving b_i in the training sample. In some sense the averaged labels $\overline{c}(a_i)$ on A are "shattered" by the choice of the distribution on $\mathcal{X}$ (and not as is typically the case in PAC learning by the class $\mathcal{C}$). The following lemma generalizes this idea to arbitrary graphs and incorporates the required dependence on ε.

Lemma 13 *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ and $G = (\mathcal{X}, E)$. Then, the sample complexity achievable by any $(G,\mathcal{C})$-learner is $m(\varepsilon, 1/2) = \Omega\left(\frac{\alpha_2(G,\mathcal{C})}{\varepsilon \log(\alpha_2(G,\mathcal{C}))}\right)$.*

Proof If $\alpha_2 = \alpha_2(G,\mathcal{C})$ is finite then there exist such c, A with $|A| = \alpha_2$ (and otherwise we can let A be arbitrarily large). To simplify the proof we assume all points of A have the same c-label (obviously, at least half of them have). For every $x \in A$ fix some $z_x \in N(x)$ with $c(z_x) \neq c(x)$, and let $B = \{z_x : x \in A\}$. Clearly $|B| \leqslant |A|$. Moreover, for every $x \in A$ let $d_B(x) = |N(x) \cap B|$. By construction, $1 \leqslant d_B(x) \leqslant |B| \leqslant |A|$. By a binning argument, then, there exists a subset $A' \subseteq A$ with $|A'| \geqslant \frac{|A|}{\lg_2 |A|}$ and some $d \in \{1, \ldots, |B|/2\}$ such that $d \leqslant d_B(x) \leqslant 2d$ for all $x \in A'$. Let then $B' = B \cap \bigcup_{x \in A'} N(x)$. Note that for all $x \in A'$ we have $|N(x) \cap B'| = |N(x) \cap B| = d_B(x)$.

We now use A' and B' as support of a family of distributions. We then show that, when we pick a distribution uniformly at random from this family, achieving expected error ε requires drawing $\Omega(|A'|/\varepsilon)$ samples. By Yao's minimax principle this implies that for *every* algorithm there exists *some* distribution from the family that yields the same bound.

To begin with, define $p(z) = \frac{1}{d|A'|+|B'|}$ for every $z \in B'$, and $p(x) = \frac{d}{d|A'|+|B'|}$ for every $x \in A'$. Note that $p(z) = \frac{p(x)}{d}$ for every $z \in B'$ and $x \in A'$. Let $A' = \{x_i : i = 1, \ldots, K\}$. For simplicity, and without loss of generality, we assume K is even. Now, for every string $\boldsymbol{s} \in \{-1,+1\}^K$ that sums to zero, define the distribution $\mathcal{D}_{\boldsymbol{s}}$ as follows: $\mathcal{D}_{\boldsymbol{s}}(z) = p(z)$, for $z \in B'$ and $\mathcal{D}_{\boldsymbol{s}}(x_i) = p(x)(1 + s_i \cdot \sqrt{\varepsilon})$ for $i \in [K]$. One can check that $\mathcal{D}_{\boldsymbol{s}}$ is indeed a distribution. Denote the average label of x under the distribution $\mathcal{D}_{\boldsymbol{s}}$ by $\overline{c}_{\boldsymbol{s}}$. Now let $\boldsymbol{s}$ be chosen uniformly at random (again so that it sums to zero).

Since $\mathcal{D}_{\boldsymbol{s}}(A') \geqslant \frac{1}{2}$, for any estimate h of the average labels then

$$\mathbb{E}_{\boldsymbol{s}}[\ell_{\mathcal{D}_{\boldsymbol{s}}}(h)] \geqslant \frac{1}{2}\mathbb{E}_{\boldsymbol{s}}\mathbb{E}_{S \sim \mathcal{D}_{\boldsymbol{s}}^m}\left[\frac{1}{K}\sum_{i=1}^{K} \ell_{x_i,\mathcal{D}_{\boldsymbol{s}}}(h)\right] . \tag{25}$$

Now, without loss of generality, we may assume the learner knows the concept $c \in \mathcal{C}$, the sets A' and B', as well as the function $p(\cdot)$, see above; it only ignores $\boldsymbol{s}$. We may also assume the predictor's output on $x_i \in A'$ is a function only of the number of times $M_{\boldsymbol{s},i}^m$ that x_i appears in the training set.[3] Clearly, $M_{\boldsymbol{s},i}^m \sim \mathrm{Bin}(m, \mathcal{D}_{\boldsymbol{s}}(x_i))$. The Kullback-Leibler divergence and Pinsker's inequality then yield, for any two $\boldsymbol{s}, \boldsymbol{s}'$ as above:

$$\left\| M_{\boldsymbol{s},i}^m - M_{\boldsymbol{s}',i}^m \right\|_{\mathrm{TVD}} \leqslant \mathcal{O}\left(\sqrt{\frac{m}{\mathcal{D}_{\boldsymbol{s}}(x_i)}} \cdot |\mathcal{D}_{\boldsymbol{s}}(x_i) - \mathcal{D}_{\boldsymbol{s}'}(x_i)|\right) \leqslant \mathcal{O}\left(\sqrt{\frac{m\,\varepsilon}{K}}\right) . \tag{26}$$

Moreover, if $\boldsymbol{s}, \boldsymbol{s}'$ differ on the i-th coordinate, then

3. Formally, the conditional mutual information $I(c(x_i);\, M \mid M_{s,i}^m)$ between the true label $c(x_i)$ of x_i and the sample M is zero when conditioned on the random variable $M_{s,i}^m$.

$$|\bar{c}_{\boldsymbol{s}}(x_i) - \bar{c}_{\boldsymbol{s}'}(x_i)| \geqslant \frac{p(x)(1+\sqrt{\varepsilon})}{p(x)(1+\sqrt{\varepsilon}) + d_B(x)p(z)} - \frac{p(x)(1-\sqrt{\varepsilon})}{p(x)(1-\sqrt{\varepsilon}) + d_B(x)p(z)} \tag{27}$$

$$= \frac{1+\sqrt{\varepsilon}}{1+\sqrt{\varepsilon} + d_B(x)/d} - \frac{1-\sqrt{\varepsilon}}{1-\sqrt{\varepsilon} + d_B(x)/d} \tag{28}$$

$$= \frac{(1+\sqrt{\varepsilon})(1-\sqrt{\varepsilon}+d_B(x)/d) \;-\; (1-\sqrt{\varepsilon})(1+\sqrt{\varepsilon}+d_B(x)/d)}{(1+\sqrt{\varepsilon}+d_B(x)/d)(1-\sqrt{\varepsilon}+d_B(x)/d)} \tag{29}$$

$$\geqslant \frac{2\sqrt{\varepsilon}d_B(x)/d}{12} \geqslant \Omega(\sqrt{\varepsilon})\,, \tag{30}$$

using that $1 \leqslant d_B(x)/d \leqslant 2$. Since $\boldsymbol{s}$ is uniformly random, the expected square loss at x_i is therefore at least

$$\Omega(\varepsilon)\left(1 - \left\|M^m_{\boldsymbol{s},i} - M^m_{\boldsymbol{s}',i}\right\|_{\mathrm{TVD}}\right) = \Omega(\varepsilon)\left(1 - \mathcal{O}\left(\sqrt{\frac{m\,\varepsilon}{K}}\right)\right)\,. \tag{31}$$

By averaging over all x_i (see right hand side of Equation (25)), to have total expected loss at most ε, one needs a sample of size $m = \Omega\left(\frac{K}{\varepsilon}\right) = \Omega\left(\frac{\alpha_2}{\varepsilon \log \alpha_2}\right)$, which concludes the proof. ■

Note that the lower bound in Lemma 13 applies even if the target concept c is known. The uncertainty comes from the marginal distribution $\mathcal{D}$ that determines the averaged labels $\bar{c}$.

5. Discussion

We provide some comments and discuss multiple interesting extensions of our learning problem.

First note that, despite speaking of a graph G in our exposition, our results also apply on infinite domains which may even be uncountable. In particular, they are applicable in standard geometric settings such as in Euclidean spaces. For example, the neighborhoods can be hyperrectangles in $\mathbb{R}^d$ or balls in a general metric space. In this case, the independence number of the graph $\alpha(G)$ becomes the packing number of the metric space. Further note that we can easily extend our analysis to other loss functions, which typically only requires an alternative of Lemma 7. For example, if we care about the absolute loss instead of the squared loss, we have to use $\mathcal{O}(1/\varepsilon^2)$ samples to estimate the average label in each neighborhood instead of $\mathcal{O}(1/\varepsilon)$ as before. The remaining arguments are otherwise independent of the particular loss adopted.

Beyond just predicting averaged labels, there can be situations where also the input sample consists of averaged labels instead of the original labels of the instances. In particular, there could be a second graph $G_{\mathrm{in}} = (\mathcal{X}, E_{\mathrm{in}})$ that models how the labels are averaged for the training set: the training set $S \subseteq \mathcal{X}$ contains points x labeled by $\bar{c}_{\mathrm{in}}(x) = \mathbb{E}_{x' \sim \mathcal{D}}\big[c(x') \mid x' \in N_{G_{\mathrm{in}}}[x]\big]$ or alternatively by the empirical average $\bar{c}_{\mathrm{in}}(x) = \frac{1}{|N_{G_{\mathrm{in}}}[x] \cap S|} \sum_{x' \in N_{G_{\mathrm{in}}}[x] \cap S} c(x')$. Studying the learnability of $(G_{\mathrm{in}}, G, \mathcal{C})$—with input labels $\bar{c}_{\mathrm{in}}$ and target labels $\bar{c}$—is a promising research direction. Our studied learning problem corresponds to the case $E_{\mathrm{in}} = \varnothing$ but arbitrary E, while different variants of the *learning from label proportions* problem correspond to either $E = \varnothing$ but arbitrary E_{in} (see the mentioned references in Section 1) or to $E = E_{\mathrm{in}}$ (Iyer et al., 2016; Fish and Reyzin, 2017).

Moreover, as noted in Section 1.2, extending our analysis from estimating $\mathbb{E}[c(x') \mid x' \in N[x]]$ to $\mathbb{E}[\mathbb{I}\{g_x(x') \neq c(x')\} \mid x' \in N[x]]$ would be interesting. In this case, $g_x : N[x] \to \{0,1\}$ is

a local classifier from a known (partial) hypothesis class $\mathcal{H}$ (say linear classifiers) defined on each neighborhood. This setup would encompass our learning problem here (with g_x being constant functions) as well as the auditing framework of Bhattacharjee and von Luxburg (2024).

Also, extensions to a regression setting with target labels in $[0, 1]$ are possible. This would allow to formulate estimation problems like *"what is the average income of all people with at least my age?"*, where *income* is the target label and the directed neighborhoods are given by *age*.

Additionally, there might be cases where we want to assign similarities to each neighbor. Instead of $\mathbb{E}_{x'}[c(x') \mid N[x]]$ we want to estimate $\mathbb{E}_{x'}[w(x, x')c(x') \mid N[x]]$ for a given similarity function $w : \mathcal{X} \times \mathcal{X} \to [0, 1]$, e.g., Gaussian. Our results correspond to the uniform similarity $w(x, x') = 1$. Learning such weighted averages in the special case of $N[x] = \mathcal{X}$ would bring our setup closer to the smoothed analysis of Chandrasekaran et al. (2024).

Finally, in Appendix C we discuss how empirical risk minimization (ERM) can be applied to our problem. In Appendix D we discuss a generalization of our results to a setting where instead of a fixed graph G, graphs G and concepts c come as pairs from a known class $\mathcal{F} \subseteq \mathcal{G} \times \mathcal{C}$.

Acknowledgments

MB, NCB, and EE are partially supported by the EU Horizon CL4-2022-HUMAN-02 research and innovation action under grant agreement 101120237, project ELIAS (European Lighthouse of AI for Sustainability).

YM has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 882396), by the Israel Science Foundation, the Yandex Initiative for Machine Learning at Tel Aviv University and a grant from the Tel Aviv University Center for AI and Data Science (TAD).

SM is supported by the Technion Center for Machine Learning and Intelligent Systems (MLIS), and by 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

MT is supported by the German Research Foundation through the project 560788681.

References

Ishaq Aden-Ali, Yeshwanth Cherapanamjeri, Abhishek Shetty, and Nikita Zhivotovskiy. Optimal PAC bounds without uniform convergence. In *Annual Symposium on Foundations of Computer Science, FOCS*, pages 1203–1223, 2023.

Noga Alon, Nicolò Cesa-Bianchi, Ofer Dekel, and Tomer Koren. Online learning with feedback graphs: Beyond bandits. In *Conference on Learning Theory, COLT*, 2015.

Noga Alon, Nicolò Cesa-Bianchi, Claudio Gentile, Shie Mannor, Yishay Mansour, and Ohad Shamir. Nonstochastic multi-armed bandits with graph-structured feedback. *SIAM Journal on Computing*, 46(6):1785–1826, 2017.

Noga Alon, Steve Hanneke, Ron Holzman, and Shay Moran. A theory of PAC learnability of partial concept classes. In *Annual Symposium on Foundations of Computer Science, FOCS*, 2022.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations, ICLR*, 2018.

Akshay Balsubramani, Sanjoy Dasgupta, and Shay Moran. An adaptive nearest neighbor rule for classification. *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

Robi Bhattacharjee and Ulrike von Luxburg. Auditing local explanations is hard. *Advances in Neural Information Processing Systems, NeurIPS*, 2024.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

Anand Brahmbhatt, Rishi Saket, and Aravindan Raghuveer. PAC learning linear thresholds from label proportions. *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Marco Bressan, Nataly Brukhim, Nicolò Cesa-Bianchi, Emmanuel Esposito, Yishay Mansour, Shay Moran, and Maximilian Thiessen. A fine-grained characterization of PAC learnability. In *Conference on Learning Theory, COLT*, volume 291, 2025.

Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In *Annual Symposium on Foundations of Computer Science, FOCS*, 2022.

Robert Busa-Fekete, Heejin Choi, Travis Dick, Claudio Gentile, and Andres Munoz Medina. Easy learning from label proportions. In *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Robert Istvan Busa-Fekete, Travis Dick, Claudio Gentile, Haim Kaplan, Tomer Koren, and Uri Stemmer. Nearly optimal sample complexity for learning with label proportions. In *International Conference on Machine Learning, ICML*, 2025.

Yair Caro. New results on the independence number. Technical report, Tel-Aviv University, 1979.

Gautam Chandrasekaran, Adam Klivans, Vasilis Kontonis, Raghu Meka, and Konstantinos Stavropoulos. Smoothed analysis for learning concepts with low intrinsic dimension. In *Conference on Learning Theory, COLT*, 2024.

John J Cherian and Emmanuel J Candès. Statistical inference for fairness auditing. *Journal of Machine Learning Research*, 25(149):1–49, 2024.

Ben Chugg, Santiago Cortes-Gomez, Bryan Wilder, and Aaditya Ramdas. Auditing fairness by betting. *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *Conference on Learning Theory, COLT*, 2014.

Sanjoy Dasgupta, Nave Frost, and Michal Moshkovitz. Framework for evaluating faithfulness of local explanations. In *International Conference on Machine Learning*, 2022.

Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989.

Khaled Eldowa, Emmanuel Esposito, Tom Cesari, and Nicolò Cesa-Bianchi. On the minimax regret for online learning with feedback graphs. In *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

Emmanuel Esposito, Federico Fusco, Dirk van der Hoeven, and Nicolò Cesa-Bianchi. Learning on the edge: Online learning with stochastic feedback graphs. In *Advances in Neural Information Processing Systems, NeurIPS*, 2022.

Benjamin Fish and Lev Reyzin. On the complexity of learning from label proportions. In *International Joint Conference on Artificial Intelligence, IJCAI*, 2017.

Steffen Grunewalder. Plug-in estimators for conditional expectations and probabilities. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems, NeurIPS*, 2016.

David Haussler, Nick Littlestone, and Manfred K Warmuth. Predicting $\{0,1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.

Daniel Hsu, Jizhou Huang, and Brendan Juba. Distribution-specific auditing for subgroup fairness. In *Symposium on Foundations of Responsible Computing, FORC*, 2024.

Arun Shankar Iyer, J Saketha Nath, and Sunita Sarawagi. Privacy-preserving class ratio estimation. In *International Conference on Knowledge Discovery and Data Mining, KDD*, 2016.

Hendrik Kück and Nando de Freitas. Learning about individuals from group statistics. In *Conference on Uncertainty in Artificial Intelligence, UAI*, 2005.

Gene Li, Lin Chen, Adel Javanmard, and Vahab Mirrokni. Optimistic rates for learning from label proportions. In *Conference on Learning Theory, COLT*, 2024.

Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In *Advances in Neural Information Processing Systems, NeurIPS*, 2011.

Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems, NeurIPS*, 2007.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *International Conference on Knowledge Discovery and Data Mining, KDD*, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, 2018.

Guy N Rothblum and Gal Yona. Multi-group agnostic PAC learnability. In *International Conference on Machine Learning, ICML*, 2021.

Benjamin I. P. Rubinstein, Peter L. Bartlett, and J. Hyam Rubinstein. Shifting, one-inclusion mistake bounds and tight multiclass expected risk bounds. *Advances in Neural Information Processing Systems, NeurIPS*, 2006.

Victor K. Wei. A lower bound on the stability number of a simple graph. Technical memorandum, Bell Laboratories, 1981.

Appendix A. Missing proofs

This section contains the missing proofs and details which enable our main results from Section 4. We use the following simple estimation lemma for the squared loss.

Lemma 7 *Let $\lambda \in (0,1]$, $c : \mathcal{X} \to \{0,1\}$, $G = (\mathcal{X}, E)$, $\mathcal{D}$ a distribution over $\mathcal{X}$, and S a random multiset of $m \in \mathbb{N}_+$ i.i.d. samples from $\mathcal{D}$ and labeled by c. Let h_S be the output of Algorithm 1 given input S. Then, all points $x \in \mathcal{X}$ with $\mathcal{D}(N[x]) \geqslant \lambda$ satisfy $\mathbb{E}_{S\sim\mathcal{D}^m}\big[\ell_x(h_S)\big] \leqslant \frac{7}{2m\lambda}$.*

Proof For any fixed $x \in \mathcal{X}$, define $f(x)$ to be the empirical mean of c-labels over points in $N[x] \cap S$; that is for any $x \in \mathcal{X}$ we can write

$$f(x) = \frac{1}{M_x} \sum_{i=1}^{m} c(x_i)\mathbb{I}\{x_i \in N[x]\} \ , \tag{32}$$

where $M_x = \sum_{i=1}^{m} \mathbb{I}\{x_i \in N[x]\}$ is the number of sampled points in S that belong to $N[x]$. Given this, we know that $h_S(x) = f(x)$ whenever $M_x > 0$, and otherwise $h_S(x)$ is the output returned by the OIG predictor.

Fix $x \in \mathcal{X}$ to be any point satisfying $\mathcal{D}(N[x]) \geqslant \lambda$. Defining $p_x = \mathbb{P}_{x'\sim\mathcal{D}}(x' \in N[x]) = \mathcal{D}(N[x])$, we can notice that $M_x \sim \mathrm{Bin}(m, p_x)$ is a binomial random variable with mean $\mathbb{E}_S[M_x] = mp_x$. Consequently, a multiplicative Chernoff bound for binomials shows that

$$\mathbb{P}_S\big(M_x \leqslant mp_x/2\big) \leqslant e^{-mp_x/8} \leqslant e^{-m\lambda/8} \, . \tag{33}$$

where the last inequality is due to the assumption that $p_x = \mathcal{D}(N[x]) \geqslant \lambda$. On the other hand, we will now bound the expected error, conditioned on the event that M_x is not small. Specifically, for a fixed x and a fixed integer $M \geqslant mp_x/2$, let $\widehat{y}(S) = \frac{1}{M}\sum_{i=1}^{M} c(x_i)$ given a sample S of size M, and $y = \overline{c}(x)$. Observe that

$$\mathbb{E}_S\big[(f(x) - \overline{c}(x))^2 \mid M_x = M\big] = \mathbb{E}_{S_x\sim\mathcal{D}_x^M}\big[(\widehat{y}(S_x) - y)^2\big] \, , \tag{34}$$

where $\mathcal{D}_x$ is the distribution $\mathcal{D}$ conditioned on $N[x]$. Then, notice that the quantity we are considering here is the variance of an empirical mean estimator over i.i.d. random variables bounded in $[0,1]$, and so by Proposition 14 we have $\mathbb{E}_{S_x\sim\mathcal{D}_x^M}\big[(\widehat{y}(S_x) - y)^2\big] \leqslant 1/(4M)$. It follows that

$$\mathbb{E}_S\big[\ell_x(h_S) \mid M_x = M\big] = \mathbb{E}_S\big[(f(x) - \overline{c}(x))^2 \mid M_x = M\big] \leqslant \frac{1}{4M} \leqslant \frac{1}{2mp_x} \leqslant \frac{1}{2m\lambda} \, . \tag{35}$$

By then taking the law of total expectation for all possible values $M \geqslant mp_x/2$, we also get that,

$$\mathbb{E}_S\big[\ell_x(h_S) \mid M_x \geqslant mp_x/2\big] \leqslant \frac{1}{2m\lambda} \, . \tag{36}$$

Lastly, combining all these observations together, and using the fact that $\ell_x(h_S) \in [0,1]$, we finally derive that

$$\mathbb{E}_S\big[\ell_x(h_S)\big] \leqslant \mathbb{E}_S\big[(f(x) - \overline{c}(x))^2 \mid M_x \geqslant mp_x/2\big] + \mathbb{P}_S\big(M_x \leqslant mp_x/2\big) \tag{37}$$

$$\leqslant \frac{1}{2m\lambda} + e^{-m\lambda/8} \leqslant \frac{7}{2m\lambda} \, , \tag{38}$$

which concludes the proof. ∎

The proof of the above lemma relies on this standard result about the variance of the empirical mean of bounded random variables.

Proposition 14 *For a positive integer $M \in \mathbb{N}_+$, let $a_1, \dots, a_M$ be i.i.d. $[0,1]$-valued random variables with $\mathbb{E}[a_i] = \mu$, and define $\bar{a}$ to be their average. Then,*

$$\mathbb{E}\big[(\bar{a} - \mu)^2\big] \leqslant \frac{1}{4M} \,.$$

Proof Since $a_1, \dots, a_M$ are i.i.d. with mean μ, we have $\mathbb{E}[\bar{a}] = \mu$ and hence $\mathbb{E}\big[(\bar{a}-\mu)^2\big] = \mathrm{Var}(\bar{a})$. Moreover, by independence of the a_i's we also have $\mathrm{Var}(\bar{a}) = \mathrm{Var}(a)/M$. Since $a \in [0,1]$, we have $\mathrm{Var}(a) \leqslant 1/4$ by Popoviciu's inequality. Combining the above yields the claim. ■

For the high-probability upper bound in Theorem 10, we rely on the following lemma showing how the expected squared distance of the median of bounded random variables from a target random variable improves whenever more than half of them are sufficiently close in expectation.

Lemma 15 *Let $y_1, \dots, y_k, y$ be random variables and let $\varepsilon \geqslant 0$. If $p > k/2$ of the $i \in [k]$ satisfy $\mathbb{E}[(y_i - y)^2] \leqslant \varepsilon$ then the median $y_{\mathrm{med}} = \mathrm{median}(y_1, \dots, y_k)$ satisfies $\mathbb{E}[(y_{\mathrm{med}} - y)^2] \leqslant \frac{2\varepsilon}{p/k - 1/2}$.*

Proof We use the notation $(\cdot)_+ = \max(0, \cdot)$ and $(\cdot)_- = \min(0, \cdot)$. Note that

$$(y_i - y)^2 = ((y_i - y)_+)^2 + ((y_i - y)_-)^2 \tag{39}$$

holds for all i and the same holds for y_{med} instead of y_i. Let us first assume that $y \leqslant y_{\mathrm{med}}$. By definition of a median half of the $i \in [k]$ satisfy $y_i \leqslant y_{\mathrm{med}}$ and half $y_i \geqslant y_{\mathrm{med}}$. Call an i *good* if $\mathbb{E}[(y_i - y)^2] \leqslant \varepsilon$ and denote the set of all good indices by $P \subseteq [k]$. As we have p good i's, there must be $p - k/2 > 0$ good i's such that $y_i \geqslant y_{\mathrm{med}} \geqslant y$. Denote the good indices i satisfying $y_i \geqslant y_{\mathrm{med}}$ by $P_+ \subseteq P$. By the choice of P_+, we have

$$\begin{aligned}
((y_{\mathrm{med}} - y)_+)^2 &\leqslant \frac{1}{|P_+|} \sum_{i \in P_+} ((y_i - y)_+)^2 && (40)\\
&\leqslant \frac{1}{|P_+|} \sum_{i \in P_+} ((y_i - y)_+)^2 + \frac{1}{|P_+|} \sum_{i \in P \setminus P_+} ((y_i - y)_+)^2 && (41)\\
&= \frac{1}{|P_+|} \sum_{i \in P} ((y_i - y)_+)^2 \,. && (42)
\end{aligned}$$

From Equation (39) we also have that $((y_i - y)_+)^2 \leqslant (y_i - y)^2$. Taking expectations, we see that

$$\begin{aligned}
\mathbb{E}[((y_{\mathrm{med}} - y)_+)^2] &\leqslant \frac{1}{|P_+|} \sum_{i \in P} \mathbb{E}[((y_i - y)_+)^2] && (43)\\
&\leqslant \frac{1}{|P_+|} \sum_{i \in P} \mathbb{E}[(y_i - y)^2] && (44)\\
&\leqslant \frac{\varepsilon |P|}{|P_+|} \leqslant \frac{\varepsilon k}{p - k/2} \,. && (45)
\end{aligned}$$

By a symmetric argument the same holds for the case $y \geqslant y_{\mathrm{med}}$ (and correspondingly $(y_{\mathrm{med}} - y)_-$). Overall we get

$$\mathbb{E}[(y_{\mathrm{med}} - y)^2] = \mathbb{E}[((y_{\mathrm{med}} - y)_+)^2] + \mathbb{E}[((y_{\mathrm{med}} - y)_-)^2] \leqslant \frac{2\varepsilon}{p/k - \frac{1}{2}} \,. \tag{46}$$

■

Appendix B. The One-Inclusion Graph algorithm

In this section we describe the classic One-Inclusion Graph algorithm (OIG), that is used in our Algorithm 1. We start by defining the one-inclusion graph of a concept (or hypothesis) class $\mathcal{C}$, where the idea is to translate a classification learning problem to the language of graphs.

Definition 16 (One-inclusion graph; Haussler et al., 1994) *The* one-inclusion graph *of* $\mathcal{C} \subseteq \{0,1\}^n$ *is a graph* $\mathcal{G}(\mathcal{C}) = (V, E)$ *defined as follows. The vertex set is* $V = \mathcal{C}$*, and the edge set* E *corresponds to all pairs* $(v, v') \in V$ *such that the Hamming distance between* v, v' *is exactly* 1*; that is, there exists* $i \in [n]$ *such that* $v(i) \neq v'(i)$ *and for all* $j \neq i$*,* $v(j) = v'(j)$*.*

We now define an orientation of the graph in the standard way.

Definition 17 *An* orientation *of the graph* (V, E) *is a mapping* $\sigma : E \to V$ *such that* $\sigma(e) \in e$ *for each edge* $e \in E$*.*

The OIG captures a model for transduction in machine learning. A key observation of Haussler et al. (1994) is that this model captures an essential ingredient of general PAC learnability; see also Rubinstein et al. (2006); Daniely and Shalev-Shwartz (2014); Brukhim et al. (2022); Bressan et al. (2025). The OIG algorithm is presented in Algorithm 2 below.

Algorithm 2 The one-inclusion algorithm $\mathcal{A}_{\mathcal{C}}$ for $\mathcal{C} \subseteq \mathcal{Y}^{\mathcal{X}}$

Input: A $\mathcal{C}$-realizable sample $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big)$.
Output: A hypothesis $\mathcal{A}_{\mathcal{C}}(S) = f_S : \mathcal{X} \to \mathcal{Y}$.
For each $x \in \mathcal{X}$, the value $f_S(x)$ is computed as follows.

1: Consider the class of all patterns over the *unlabeled data* $\mathcal{C}|_{(x_1,\ldots,x_m,x)} \subseteq \mathcal{Y}^{m+1}$.
2: Find an orientation σ of $\mathcal{G}(\mathcal{C}|_{(x_1,\ldots,x_m,x)})$ that minimizes the maximum out-degree.
3: Let $e = e_{y_1,\ldots,y_m}$ denote the edge determined by the labels in S over the first m coordinates.
4: Set $f_S(x) \coloneqq \sigma(e)_{m+1}$. That is, the label $f_S(x)$ is determined by the label in the last coordinate of the node points to by the orientation σ.

The algorithm gets as input a realizable training sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ as well as an additional test point x. Its goal is to provide a good prediction for the label of x. An orientation of the graph provides the prediction for the label of x.

Appendix C. VC classes allow ERM

We briefly note that under stronger assumptions than required by our main theorem (Theorem 6), we can instead use (an appropriate variant of) Empirical Risk Minimization (ERM) to design a $(G, \mathcal{C})$-learner. In particular, since the analysis of ERM leverages uniform convergence for VC classes, we additionally require that $\mathcal{C}$ has finite VC dimension $d = \mathrm{VC}(\mathcal{C})$.

The alternative ERM-based algorithm performs what follows. Using an i.i.d. sample S from $\mathcal{D}$ of size $\mathcal{O}\big(\frac{d\log(1/\varepsilon)+\log(1/\delta)}{\varepsilon}\big)$ labeled by any fixed concept $c \in \mathcal{C}$, use ERM over S to obtain a predictor h_{erm} that satisfies $\mathbb{P}_{x\sim\mathcal{D}}(h_{\mathrm{erm}}(x) \neq c(x)) \leqslant \varepsilon/2$ with probability at least $1 - \delta/2$; this follows from standard results on realizable PAC learning for VC classes (Blumer et al., 1989).

We now use a second sample S' of size $\mathcal{O}\big(\frac{\alpha_2(G,\mathcal{C})\log(1/\varepsilon)}{\varepsilon}\big)$ to compute the empirical mean labels over the neighborhood of a test point $x \sim \mathcal{D}$ using both the sampled points within $N[x]$ as well as the label $h_{\mathrm{erm}}(x)$ predicted by the ERM classifier. Now let $M_x = |N[x] \cap S'|$ be the number of examples from S'. Then, the resulting predictor $h_{S'}$ outputs

$$h_{S'}(x) = \frac{1}{M_x + 1}\left(h_{\mathrm{erm}}(x) + \sum_{x' \in N[x] \cap S'} c(x')\right) \tag{47}$$

for any $x \in \mathcal{X}$. While we can show that this predictor provides the desired guarantee in expectation, by a confidence amplification argument as in Theorem 10, given the success of h_{erm}, we can extend the loss guarantee to one that holds with probability $1-\delta/2$. We can show that the learning algorithm we just described yields the following sample complexity.

Proposition 18 *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ be a class with VC dimension $d = \mathrm{VC}(\mathcal{C})$ and $G = (\mathcal{X}, E)$ a directed graph. There exists an ERM-based $(G,\mathcal{C})$-learner with sample complexity*

$$m_{\mathrm{erm}}(\varepsilon,\delta) = \mathcal{O}\left(\frac{\big(d + \alpha_2(G,\mathcal{C})\log(1/\delta)\big)\log(1/\varepsilon)}{\varepsilon}\right).$$

Proof Fix $c \in \mathcal{C}$ to be the ground-truth labeling. As already mentioned, we know that the ERM predictor h_{erm} guarantees $\mathbb{P}_{x\sim\mathcal{D}}(h_{\mathrm{erm}}(x) \neq c(x)) \leqslant \varepsilon/2$ with probability $1 - \delta/2$. By standard realizable PAC learning results, as mentioned above, this can be done using $\mathcal{O}\big(\frac{d\log(1/\varepsilon)+\log(1/\delta)}{\varepsilon}\big)$ i.i.d. samples. We condition on this event in what follows.

On a mass of at most $\varepsilon/2$ of the points, h_{erm} errs. For these points we can simply assume the worst-case upper bound of 1 on the loss, and condition now on the event that h_{erm} predicts the true label $c(x)$ on the test point x.

Next define $h_{S'}$ as in Equation (47) using the second sample S'. The proof of Theorem 6 goes through almost exactly as before. In particular, the cases $\mathcal{E}_1^{(i)}$ and $\mathcal{E}_2^{(i)}$ remain valid and we can choose the constants in the sample complexity such that the expected error is at most $\varepsilon/2$ in case $\mathcal{E}_1^{(i)}$ (and 0 in case $\mathcal{E}_2^{(i)}$). The only difference is here that we average over $M_x + 1$ labels in $N[x]$. This is valid due to the following two reasons. First, as we get the correct label $h_{\mathrm{erm}} = c(x)$ for the test point $x \in N[x]$. Second, as x is i.i.d. from $\mathcal{D}$ averaging over $(N[x] \cap S') \cup \{x\}$ is equivalent to sampling $M_x + 1$ points from $N[x]$ and the analysis proceeds as before (with one additional labeled point). In case $\mathcal{E}_3^{(i)}$, where $M_x = 0$, it simply holds that $\bar{c}(x) = c(x) = h_{\mathrm{erm}}(x)$ and thus we predict correctly.

Finally, we again apply Lemma 11 to turn the in-expectation guarantee of $h_{S'}$ into one with probability $1 - \delta/2$ instead of just in expectation, always given that h_{erm} succeeds. We recall it suffices to independently construct $\mathcal{O}(\log(1/\delta))$ predictors $h_{S'}$ as described above, where each one of them requires $\mathcal{O}\big(\frac{\alpha_2(G,\mathcal{C})\log(1/\varepsilon)}{\varepsilon}\big)$ i.i.d. samples to provide its individual guarantee.

A union bound over the failure of h_{erm} and that of the median-based predictor from Lemma 11 concludes the proof. ■

Note that $d \geqslant \alpha_1(G,\mathcal{C})$ by definition, and indeed the gap can be arbitrarily large; e.g., if G is a clique then $d = \mathrm{VC}(\mathcal{C})$ grows arbitrarily as the class $\mathcal{C}$ becomes more complex while $\alpha_1(G,\mathcal{C}) \leqslant 1$ for any $\mathcal{C}$. This shows that this simpler ERM-based approach is sufficient for learning if $\mathcal{C}$ is a

VC class, but can lead to a significantly worse sample complexity than the one achieved by our Algorithm 1, and proved in Theorem 10.

We remark that the question of whether ERM also works with a sample complexity depending on $\alpha_1(G, \mathcal{C})$ instead of $d = \mathrm{VC}(\mathcal{C})$ remains open. The next section shows that at least in a more general setting, such ERM-based approaches are not sufficient for learnability.

Appendix D. Extension to multiple graphs

In a possible generalization of our problem, instead of having a single fixed graph, we can imagine that the concept c and graph G come as a pair $(G, c) \in \mathcal{F}$ from a known joint class $\mathcal{F} \subseteq \mathcal{G} \times \mathcal{C}$. Here $\mathcal{G}$ is a family of graphs all with the same node set $\mathcal{X}$. The target graph G itself is fixed yet unknown by the learner and only accessible through a neighborhood oracle on the sample, which allows to check whether any two points in the sample are adjacent in the graph. This clearly generalizes our setting as it can be recovered by $\mathcal{G} = \{G\}$. We denote by $L_G(h)$ the averaged loss of a classifier h as in Equation (4) with neighborhoods and averaged labels $\bar{c}(\cdot)$ given by the graph G.

Definition 19 *Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$, let $\mathcal{G}$ be a family of directed graphs on $\mathcal{X}$, and let $\mathcal{F} \subseteq \mathcal{G} \times \mathcal{C}$. A learning rule $\mathcal{A}$ is an* $\mathcal{F}$-learner *if there exists a sample complexity $m : (0,1)^2 \to \mathbb{N}$ such that for every distribution $\mathcal{D}$ over $\mathcal{X}$, every $(G, c) \in \mathcal{F}$, and every $\varepsilon, \delta \in (0,1)$ the following holds. When given a set S of $m \geqslant m(\varepsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$ and labeled by c, then the learning rule returns a predictor $h_S = \mathcal{A}(S)$ such that $L_G(h_S) \leqslant \varepsilon$ with probability at least $1 - \delta$ over S.*

Our lower bounds apply also in this setting after a small modification. In particular, $\alpha_1(\mathcal{F})$ is the maximal size of a set $I \subseteq \mathcal{X}$ that is shattered by a set $S \subseteq \mathcal{C}$ (in the usual sense) with the additional requirement that each $c \in S$ can be extended to a tuple $(G, c) \in \mathcal{F}$ where I is independent in G. That is, $\alpha_1(\mathcal{F})$ is the size of a largest subset $I \subseteq \mathcal{X}$ such that $|\{c \cap I : (G, c) \in \mathcal{F}, I \text{ independent in } G\}| = 2^{|I|}$. We also adapt $\alpha_2(\mathcal{F}) = \sup_{(G,c)\in\mathcal{F}} \alpha_2(G, c)$. We see that we can replace α_1 and α_2 in the previous lower bounds with these modifications.

Also our algorithmic upper bound remains valid. Indeed, one can check that Theorem 6 goes through exactly as before when we replace the fixed graph G with the chosen $(G, c) \in \mathcal{F}$. The reason is that already for the original problem we had no dependence on the full graph and only used neighborhood oracle access.

We thus get the following sample complexity bounds for this generalized problem:

$$\Omega\left(\frac{\alpha_1(\mathcal{F}) + \alpha_2(\mathcal{F})/\log(\alpha_2(\mathcal{F})) + \log(1/\delta)}{\varepsilon}\right) \leqslant m_{\mathcal{F}}(\varepsilon, \delta) \leqslant \mathcal{O}\left(\frac{\alpha_1(\mathcal{F}) + \alpha_2(\mathcal{F})\log(1/\varepsilon)}{\varepsilon}\log\frac{1}{\delta}\right). \tag{48}$$

Note that this problem generalizes not only standard PAC learning but also learning with *partial concept classes* (Alon et al., 2022). In particular, let $\widetilde{\mathcal{C}} \subseteq \{0, 1, \star\}^{\mathcal{X}}$ be a partial concept class. We can encode any partial concept $\widetilde{c} \in \widetilde{\mathcal{C}}$ by a pair (G, c) where c agrees with $\widetilde{c}$ on the support of $\widetilde{c}$ and outside of the support we set c to 1. The graph G is given by isolated vertices on the support of $\widetilde{c}$ and a clique on the rest. Call the collection of all these pairs $\mathcal{F}$. It is easy to verify that there exists an $\mathcal{F}$-learner if and only if the partial class $\widetilde{\mathcal{C}}$ is PAC learnable.

As uniform convergence (and thus ERM and proper learning) does not succeed for general partial concept classes, this shows that uniform convergence will fail for the more general problem of learning with pairs $\mathcal{F} \subseteq \mathcal{G} \times \mathcal{C}$ too.