PAC Learning with Bandit Feedback: Sharp Sample Complexity in the Realizable Setting

Steve Hanneke
Department of Computer Science
Purdue University
West Lafayette, IN, USA
steve.hanneke@gmail.com

Qinglin Meng
Department of Computer Science
Purdue University
West Lafayette, IN, USA
meng160@purdue.edu

Shay Moran
Faculties of Mathematics, Computer Science, and Data and Decision Sciences
Technion – Israel Institute of Technology and Google Research
Haifa, Israel
smoran@technion.ac.il

Amirreza Shaeiri
Department of Computer Science
Purdue University
West Lafayette, IN, USA
amirreza.shaeiri@gmail.com

Abstract

We study the problem of multiclass PAC learning with bandit feedback in the realizable setting. In this framework, there is an unknown data distribution over an instance space $\mathcal{X}$ and a label space $\mathcal{Y}$, as in classical multiclass PAC learning, but the learner does not observe the labels of the i.i.d. training examples. Instead, in each round, it receives an unlabeled instance, predicts its label, and receives bandit feedback indicating only whether the prediction is correct. Despite this restriction, the goal remains the same as in classical PAC learning.

We provide a general characterization of the optimal sample complexity of this problem, sharp for every non-trivial concept class, up to logarithmic factors. Our characterization is based on a new combinatorial dimension, termed the *bandit* DS *dimension*, defined via generalized combinatorial structures we call *pseudo-boxes*. These extend the pseudo-cubes underlying the DS dimension by allowing a different number of neighbors in each coordinate. In contrast to the DS dimension, which governs the full-information setting by counting the number of coordinates in the pseudo-cube, the bandit DS dimension aggregates the number of neighbors across coordinates, leading to a characterization in which the sample complexity scales with the total number of neighbors.

We also propose a general learning algorithm achieving the upper bound, based on an algorithmic principle called *ListCascade*, which connects bandit learning to list learning and may be of independent interest.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

A natural variant of multiclass PAC learning arises when the learner does not observe labels, but only receives binary feedback indicating whether its predictions are correct. This setting, known as multiclass PAC learning with bandit feedback, was introduced and studied in the PAC framework by Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011, 2015]. Formally, there is an unknown distribution $\mathcal{D}$ over an instance space $\mathcal{X}$ and a label space $\mathcal{Y}$. The training process proceeds over n rounds: in each round, a sample $(x, y) \sim \mathcal{D}$ is drawn, the instance x is revealed, and the learner predicts a label $\hat{y} \in \mathcal{Y}$. Instead of observing the true label, the learner only receives bandit feedback indicating whether $\hat{y} = y$. The goal is to output a hypothesis $\hat{h} : \mathcal{X} \to \mathcal{Y}$ that generalizes well to $\mathcal{D}$. We assume a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and focus on the realizable setting, where $\mathcal{D}$ is consistent with some $h \in \mathcal{H}$.

There are at least two primary motivations for studying multiclass learning under bandit feedback. First, bandit feedback is often substantially cheaper to obtain than full information. For instance, answering the question "Is this an image of a horse?" is significantly faster than selecting the correct label from a long list of possible categories. Moreover, in many real-world scenarios, this restricted form of feedback is not merely a cost-saving choice but an inherent limitation of the environment. As an illustration, imagine a pandemic caused by an unknown virus, where we aim to develop an AI model capable of accurately recommending the appropriate medication for patients drawn i.i.d. from the same underlying population. Suppose, for simplicity, that each patient responds positively to exactly one drug from a fixed set of available drugs. Due to various challenges associated with human clinical trials, including ethical and financial constraints, only a limited number of patients can be enrolled. Naturally, the only feedback we can obtain from each trial is binary—whether the drug tested was effective or not for that patient. We refer the reader to Woodroofe [1979], Li et al. [2010], Agarwal et al. [2016] for additional practical examples.

In this work, we study the following fundamental and naturally arising question:

Given a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, how many training samples are necessary and sufficient for PAC learning $\mathcal{H}$ when the learner receives only bandit feedback?

The work of Daniely et al. [2011, 2015] provided an initial result on the above question, among others. However, their upper and lower bounds have a multiplicative gap of K up to logarithmic factors, where K is the size of the effective label space. In particular, at constant confidence and for a concept class $\mathcal{H}$ with $\mathrm{ND}(\mathcal{H}) \geq 2$, they showed a lower bound of $\widetilde{\Omega}\left(\mathrm{ND}(\mathcal{H})/\epsilon\right)$ and an upper bound of $\widetilde{O}\left(K\,\mathrm{ND}(\mathcal{H})/\epsilon\right)$ on the optimal sample complexity of multiclass PAC learning with bandit feedback, where $\mathrm{ND}(\mathcal{H})$ denotes the Natarajan dimension of $\mathcal{H}$, the standard multiclass analogue of the VC dimension, defined by the largest set of points that can be shattered using two candidate labels per point.

1.1 Overview of the Main Results

Our main contribution is a new combinatorial complexity parameter that sharply characterizes the sample complexity of PAC learning with bandit feedback. The definition is based on a natural generalization of classical combinatorial structures that underlie PAC learning theory.

We begin with a simple object. An $N_1 \times N_2 \times \cdots \times N_d$ *box* is a cartesian product $A_1 \times A_2 \times \cdots \times A_d$, where each $A_i \subseteq \mathcal{Y}$ has size N_i. Boxes capture independent variation across coordinates and play a central role in classical dimensions. For example, a Natarajan cube corresponds to a $2 \times 2 \times \cdots \times 2$ box, while its natural generalization to list learning corresponds to $(\ell+1) \times \cdots \times (\ell+1)$ boxes.

In our setting, however, we require a more flexible notion. We say that a finite non-empty set $T \subseteq \mathcal{Y}^d$ is an $N_1 \times \cdots \times N_d$ *pseudo-box* if, every vector $f \in T$ and every $i \in [d]$, there are at least N_i distinct vectors in T that differ from f at coordinate i and agree with f at every other coordinate. Pseudo-boxes relax the rigid Cartesian product structure while preserving its local combinatorial richness. In particular, pseudo-cubes (the case $N_i = 1$) recover the classical structure underlying the DS dimension, while larger values of N_i correspond to its extensions used in list learning Daniely and Shalev-Shwartz [2014], Brukhim et al. [2022], Charikar and Pabbaraju [2023].

The key point is that such structures precisely capture the difficulty of learning. In the full-information setting, complexity is governed by the number of coordinates in a pseudo-box, leading to dimensions such as the DS dimension. In contrast, under bandit feedback, the difficulty accumulates across coordinates: each direction contributes separately to the complexity.

Motivated by this perspective, we define the *bandit* DS *dimension* of a concept class $\mathcal{H}$, denoted $\mathrm{BDS}(\mathcal{H})$, as the maximum total size $\sum_{i=1}^{d} N_i$ of a pseudo-box realizable by $\mathcal{H}$. We refer the reader to subsection B.2 for the formal definition and further discussion.

Equipped with this dimension, we establish matching upper and lower bounds on the optimal sample complexity of multiclass PAC learning with bandit feedback, resolving an open problem posed by Daniely et al. [2011, 2015] up to logarithmic factors.

Theorem 1.1. *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a non-trivial concept class with $\mathrm{BDS}(\mathcal{H}) < \infty$, and denote by $\mathbf{m}^{B}_{\mathcal{H}}$ the optimal sample complexity of multiclass PAC learning with bandit feedback (see Definition 3.3). Then, for every $0 < \epsilon \leq 1/16$ and $0 < \delta \leq 1/16$,*

$$\mathbf{m}^{B}_{\mathcal{H}}(\epsilon, \delta) \in \widetilde{\Theta}\left(\frac{\mathrm{BDS}(\mathcal{H})}{\epsilon}\right),$$

where $\widetilde{\Theta}$ hides polylogarithmic factors in K, $1/\epsilon$, and $1/\delta$.

The upper bound in Theorem 1.1 is based on a new algorithmic framework, which we call *ListCascade*. At a high level, ListCascade uses a sequence of list learners with progressively shrinking list sizes, thereby converting list learning guarantees into a bandit-feedback learner whose complexity is governed by the bandit DS dimension. A key ingredient in the analysis is a refined sample complexity bound for list PAC learning, which may also be of independent interest; see Section 1.2 for more details. It remains interesting to understand to what extent the polylogarithmic overhead in this general upper bound can be reduced.

We also place our results in the broader landscape of bandit learning. Although multiclass learning with bandit feedback can be viewed as a special case of contextual bandits, a direct reduction to this general setting misses a key structural feature of classification: the induced reward function is sparse. Recent works have exploited this sparsity to obtain sharper sample-complexity guarantees Erez et al. [2024b,a], Erez and Koren [2025], primarily for finite concept classes and in the agnostic regime. Our results show that, in the realizable PAC setting, this additional structure enables a combinatorial characterization of the near-optimal sample complexity, in contrast to general bandit learning problems where such characterizations are provably impossible Hanneke and Yang [2023], Brukhim et al. [2025].

1.2 Overview of the Techniques

List learning. Our first ingredient is list learning, which serves as the intermediate object connecting bandit feedback and multiclass prediction. In standard multiclass learning, the learner predicts a single label for each instance. In list learning, the learner is allowed to output a short list of candidate labels, and the prediction is considered correct as long as the true label is contained in the list. This relaxation is particularly useful under bandit feedback: even if the learner cannot immediately identify the correct label, it can progressively shrink the set of plausible labels while preserving the true one with high probability.

ListCascade. Our main algorithmic idea is a cascading procedure that repeatedly learns smaller lists from bandit feedback. Here, for simplicity, we assume that the size of the label space and effective label space are both K. This should be contrasted with the classical reduction of Daniely et al. [2011], which explores by predicting a uniformly random label from $\mathcal{Y}$. Since the correct label is revealed only with probability $1/K$, one needs about K bandit examples to obtain one full-information example, leading to the $\widetilde{O}(K\,\mathrm{ND}(\mathcal{H})/\epsilon)$ upper bound.

ListCascade replaces this one-shot exploration by a sequence of list-learning stages. Initially, the learner considers all K labels as possible. In the first epoch, it predicts uniformly from this full list and uses the revealed examples to learn a list predictor of size roughly $K/2$. In the next epoch, the learner only explores within this shorter list, and then learns an even shorter list. Repeating this process, the list size is halved from K to $K/2$, then to $K/4$, and so on, until the final list has size one; this final list predictor is the returned multiclass hypothesis.

The key point is that the cost of each epoch scales with the current list size, rather than with the original number of labels K. In epoch t, the exploration list has size at most $2L_{t-1}$, and the majority vote of the L_t-list predictors has size at most $2L_t - 1$. Writing $r_t = \lceil L_t/2 \rceil$, the list-learning guarantee below gives an epoch cost of $\widetilde{O}\big(L_{t-1} \cdot (r_t\text{-DS}(\mathcal{H}))/\epsilon\big)$. Since $L_{t-1} \leq 4r_t$ and $r_t \cdot (r_t\text{-DS}(\mathcal{H})) \leq \mathrm{BDS}(\mathcal{H})$, summing over the logarithmically many epochs gives the desired upper bound, up to logarithmic factors.

Sample complexity of list learning. It remains to establish the list-learning guarantee required in each epoch of ListCascade. We use the one-inclusion list algorithm studied by Charikar and Pabbaraju [2023]. By the analysis of Hanneke et al. [2026], its sample complexity is controlled by the corresponding list exponential dimension: learning an L-list predictor requires $\widetilde{O}(\text{L-E}(\mathcal{H})/\epsilon)$ samples, where $\text{L-E}(\mathcal{H})$ denotes the L-exponential dimension. Informally, this dimension is the list-learning analogue of the exponential dimension: it captures the largest set of points on which the class realizes exponentially many label patterns at scale L. Prior comparisons between list exponential dimension and list DS dimension yield a bound of order $\widetilde{O}(L \cdot \text{L-DS}(\mathcal{H})/\epsilon)$. While sufficient for characterizing list learning, this extra factor L is too costly for our bandit application, since it would multiply the exploration cost in ListCascade and lead to a suboptimal bound.

Our contribution is to sharpen this dimension comparison in the regime needed by the cascade. Using the optimal Sauer lemma of Hanneke et al. [2026], we show that the L-exponential dimension is controlled, up to logarithmic factors in K, by the $\lceil \text{L}/2 \rceil$-DS dimension. Hence the suffix-majority aggregation of the one-inclusion L-list predictors learns a list predictor of size $\min\{2L-1, K\}$ with sample complexity $\widetilde{O}\big((\lceil L/2 \rceil\text{-DS}(\mathcal{H}))/\epsilon\big)$. This removes the extraneous list-size factor and is the key ingredient that allows ListCascade to achieve the near-optimal bandit sample complexity.

Lower Bound. The lower bound is obtained by turning the pseudo-box witnessing $\mathrm{BDS}(\mathcal{H})$ into a hard bandit instance for non-trivial classes. Suppose $x_1, x_2, \ldots, x_d$ are BDS-shattered with multiplicities $N_1, N_2, \ldots, N_d$, where $\sum_i N_i = \mathrm{BDS}(\mathcal{H})$ and $d \geq 2$; after reordering, assume $N_1 = \min_i N_i$, so that $\sum_{i=2}^{d} N_i \geq \mathrm{BDS}(\mathcal{H})/2$. We place most of the distributional mass on one anchor point and distribute the remaining $\Theta(\epsilon)$ mass over the other points in proportion to N_i. The target restriction is then chosen uniformly from the finite pseudo-box witness, after fixing one representative hypothesis for each distinct restriction. For each point x_i, the multiplicity N_i represents the number of locally indistinguishable alternatives that differ only at x_i. Conditioned on all the other coordinates of the target, the label at x_i is uniform among at least $N_i + 1$ possible labels, while each visit to x_i only tells the learner whether one guessed label is correct; hence, after c such visits and one final prediction, the learner can correctly identify at most $c + 1$ of these possibilities. With fewer than $\Theta(\sum_{i=2}^{d} N_i/\epsilon) = \Theta(\mathrm{BDS}(\mathcal{H})/\epsilon)$ samples, this yields error at least ϵ with constant probability for some target concept. Therefore any bandit PAC learner requires $\Omega(\mathrm{BDS}(\mathcal{H})/\epsilon)$ samples, matching our upper bound up to logarithmic factors.

2 Related Work

Contextual Bandit. The bandit model occupies a central place in decision theory. Its origins trace back to the pioneering work of Thompson [1933], and the field gained prominence following Herbert Robbins' short but influential contribution Robbins [1952], which introduced the classical multi-armed bandit problem. Subsequently, Michael Woodroofe wrote the first paper on the fundamental problem of contextual bandits Woodroofe [1979]. Moreover, the literature on this topic has expanded dramatically since the landmark contributions of Auer et al. [2002a,b]. Beginning with works such as Dudik et al. [2011], Agarwal et al. [2014], a particularly influential technical idea has been the use of reduction-based approaches, which relate contextual bandits to supervised learning primitives, through importance weighting. For a comprehensive treatment of this subject, we refer the reader to Bubeck et al. [2012], Slivkins [2019], Lattimore and Szepesvári [2020], Foster et al. [2023].

Additionally, there is a recent paper on the problem studied in the current work within the framework of universal learning Hanneke et al. [2025d]. Nevertheless, in this framework, the dependence on the number of labels appears to be fundamentally different from that in the PAC learning framework. We refer the reader to Hanneke et al. [2024c, 2025d,c] for more discussion of this phenomenon.

List Learning. In recent years, there has been a growing body of work on list learning across both the theoretical and empirical branches of machine learning. On the theoretical side, list learners play a key role in the breakthrough result of Brukhim et al. [2022], which provides a combinatorial characterization of multiclass PAC learnability. This topic has since developed in several directions, including PAC learning Charikar and Pabbaraju [2023], Hanneke et al. [2024b, 2025a], Cohen et al. [2026], online learning Moran et al. [2023], boosting Brukhim et al. [2023, 2024], sample compression schemes and uniform convergence Hanneke et al. [2024a], real-valued regression Pabbaraju and Sarmasarkar [2024], transductive online learning Hanneke and Shaeiri [2025], differentially private PAC learning Hanneke et al. [2025b], combinatorial semi-bandits Erez and Koren [2025], language generation Charikar et al. [2025], and SSP lemmas Hanneke et al. [2026]. Moreover, list learning is also naturally connected to the fundamental framework of list-decodable learning, which is inspired by list decoding in coding theory. This framework was introduced by Balcan et al. [2008] in the context of clustering (see also Balcan et al. [2009]) and subsequently developed by Charikar et al. [2017] in the context of robust statistics. For a detailed exposition on list-decodable learning, for example, refer to Chapter 5 of the recent textbook by Diakonikolas and Kane [2023].

3 Preliminaries

We consider a learning setting in which a learner is tasked with classifying instances from an instance space $\mathcal{X}$ with a single label from a set of K possible labels $\mathcal{Y} = \{1, \cdots, K\}$. A stochastic multiclass classification instance is specified by a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and a joint distribution over example-label pairs over $\mathcal{X} \times \mathcal{Y}$.

We make the standard measurability assumptions that the hypotheses are measurable and that the fixed orientations, list-padding rules, and pointwise relabelings used below can be chosen so that the resulting prediction rules are jointly measurable in the sample, test instance, and internal randomness.

We say that a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is *non-trivial* if $0 < \mathrm{BDS}(\mathcal{H}) < \infty$ and there exist $d \geq 2$, a sequence $S \in \mathcal{X}^d$, a vector $N \in \mathbb{N}^d$, and a finite nonempty $\mathcal{F} \subseteq \mathcal{H}$ such that (S, N) is BDS-shattered by $\mathcal{H}$ with witness $\mathcal{F}$ and

$$\sum_{i=1}^{d} N_i = \mathrm{BDS}(\mathcal{H}).$$

In other words, the bandit DS dimension of $\mathcal{H}$ is attained by a BDS witness supported on at least two instances.

Before describing the bandit feedback model, we introduce the necessary notions used to state the PAC guarantee. The performance of a hypothesis is measured by its expected classification error.

Definition 3.1 (Error rate)**.** Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ and let $h \in \mathcal{Y}^{\mathcal{X}}$. We define the error rate of h with respect to $\mathcal{D}$ as follows:

$$\mathcal{L}_{\mathcal{D}}(h) = \Pr_{(x,y)\sim\mathcal{D}}\left[h(x) \neq y\right].$$

We focus on the realizable setting, where the hypothesis class contains a perfect classifier for the data distribution. Formally, we have the following definition.

Definition 3.2 (Realizable Distribution)**.** We say that a data distribution $\mathcal{D}$ is realizable by a concept class $\mathcal{H}$ if there exists $h^\star \in \mathcal{H}$ such that $\mathcal{L}_{\mathcal{D}}(h^\star) = 0$.

We now specify the bandit feedback learning protocol. In the bandit feedback setup, the learner does not observe the true label y_i directly; instead, it only observes whether its prediction is correct. Learning proceeds through the following protocol, over $i = 1, 2, \cdots, m$:

1. The environment samples an instance (x_i, y_i) from $\mathcal{D}$ and reveals x_i to the learner.
2. The learner predicts a label $\hat{y}_i$ from $\mathcal{Y}$.
3. Nature reveals the feedback $f_i = \mathbb{1}\{y_i = \hat{y}_i\}$ to the learner.

Finally, based on $\{(x_1, \hat{y}_1, f_1), \ldots, (x_m, \hat{y}_m, f_m)\}$, the learner returns a hypothesis $\hat{h} : \mathcal{X} \to \mathcal{Y}$.

The goal of PAC learning is to output, with high probability, a hypothesis whose expected error is small and this requirement is formalized by the following sample complexity notion.

Definition 3.3 (Sample complexity for realizable multiclass PAC learning with bandit feedback)**.** For $\epsilon, \delta \in (0,1)$, the realizable PAC sample complexity of $\mathcal{H}$ with bandit feedback, denoted by $\mathbf{m}^B_{\mathcal{H}}(\epsilon, \delta)$, is the smallest integer m such that there exists a learning algorithm satisfying the following: for every distribution $\mathcal{D}$ realizable by $\mathcal{H}$, after m rounds of interaction with bandit feedback, the returned hypothesis $\hat{h}$ satisfies

$$\Pr\left[\mathcal{L}_{\mathcal{D}}\left(\hat{h}\right) > \epsilon\right] \leq \delta,$$

where the probability is over the sample $S \sim \mathcal{D}^m$ and any internal randomness of the algorithm.

Next, we introduce list learning [Charikar and Pabbaraju, 2023], which will be used as an intermediate object in our analysis. In list learning, instead of predicting a single label, the learner outputs a list of candidate labels for each instance. A list predictor of size L is a mapping $\mu : \mathcal{X} \to \{Y \subseteq \mathcal{Y} : |Y| \leq L\}$. Its error rate is defined as $\mathcal{L}_{\mathcal{D}}(\mu) = \Pr_{(\mathcal{X},\mathcal{Y})\sim\mathcal{D}}[y \notin \mu(x)]$.

Next, we define the majority vote over lists, which we will use later to convert list predictors with expected-error guarantees into a list predictor satisfying a PAC guarantee.

Definition 3.4 (Majority vote for lists [Hanneke et al., 2024b])**.** For n lists $\mu_1, \ldots, \mu_n$ each of size L, we define their majority vote to be $\mu = \mathrm{Maj}(\mu_1, \ldots, \mu_n)$ with

$$\mu(x) = \mathrm{Maj}(\mu_1, \ldots, \mu_n)(x) := \{y \in \mathcal{Y} : |i \in [n] : y \in \mu_i(x)| > n/2\}, \forall x \in \mathcal{X}.$$

Note that the majority-vote list μ has size at most $2L-1$, rather than L. For notational convenience, we pad $\mu(x)$ with arbitrary labels from $\mathcal{Y} \setminus \mu(x)$ so that all output lists have size exactly $\min\{2L-1, K\}$. When $L = 1$, this is strict-majority voting, with an arbitrary fallback label when no label receives a strict majority.

4 Algorithm and Analysis

In this section, we present and analyze one of our main contributions: a near optimal bandit multiclass classification algorithm, detailed in Algorithm 1, which will be shown to satisfy the following realizable PAC guarantee.

Remark 4.1. For simplicity of presentation, throughout this section we state the bounds in terms of the label space size $K = |\mathcal{Y}|$. The same arguments can be formulated in terms of the effective label space size of the class, which can be substantially smaller than K. We discuss this refinement in the Appendix D.

Theorem 4.2 (PAC Sample Complexity Upper Bound for Multiclass Classification with Bandit-feedback)**.** *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a concept class with bandit DS dimension $\mathrm{BDS}(\mathcal{H}) = d^B_{DS} < \infty$. If $K = 1$, then $\mathbf{m}^B_{\mathcal{H}}(\epsilon, \delta) = 0$. For $K \geq 2$, we have*

$$\mathbf{m}^B_{\mathcal{H}}(\epsilon, \delta) = O\left(\frac{d^B_{DS}(\log K)^3 + K \log K \log\left(\frac{2\log K}{\delta}\right)}{\epsilon}\right).$$

For comparison, we also prove the following matching lower bound up to logarithmic factors.

Theorem 4.3 (PAC Sample Complexity Lower Bound for Multiclass Classification with Bandit-feedback)**.** *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a non-trivial concept class with bandit DS dimension $\mathrm{BDS}(\mathcal{H}) = d^B_{DS} < \infty$. Let $K = |\mathcal{Y}| \geq 2$. Then, for every $0 < \epsilon \leq 1/16$ and $0 < \delta \leq 1/16$, we have*

$$\mathbf{m}^B_{\mathcal{H}}(\epsilon, \delta) = \Omega\left(\frac{d^B_{DS} + \log\left(\frac{1}{\delta}\right)}{\epsilon}\right).$$

Algorithm 1 proceeds by gradually shrinking the list of candidate labels. In epoch t, the learner explores using the previous list predictor μ_{t-1} by predicting uniformly from $\mu_{t-1}(x)$, and keeps an example only when the bandit feedback is positive. The retained examples form the labeled sample S_t. If $n_t < 4$, let μ_t be any fixed list predictor of the required size; otherwise define μ_t by Equation (1):

$$\mu_t = \mathrm{Maj}\left(\mu^{L_t}_{S_{t,\leq\lceil n_t/4\rceil}}, \ldots, \mu^{L_t}_{S_{t,\leq n_t-1}}\right), \tag{1}$$

Algorithm 1 ListCascade: Bandit PAC Multiclass Classification via List Learners

Input: An unlabeled sample $S = (x_1, x_2, \ldots, x_m)$. **Output:** A multiclass classification hypothesis h_S.

Set $m_0 = 0$, $\mu_0(x) = \mathcal{Y}$ for any $x \in \mathcal{X}$, and $L_0 = \lceil K/2 \rceil$.
for epoch $t = 1, 2, \cdots, \lfloor \log K \rfloor$ **do**
  Initialize $S_t = ()$.
  for $i = m_{t-1} + 1, m_{t-1} + 2, \cdots, m_t$ **do**
    Observe x_i, predict $\hat{y}_i \sim \mathrm{Unif}\left(\mu_{t-1}(x_i)\right)$ and receive feedback $\mathbb{1}\{y_i = \hat{y}_i\}$.
    Update $S_t \leftarrow (S_t, (x_i, y_i))$. if $y_i = \hat{y}_i$, otherwise S_t is unchanged.
  end for
  Let $L_t = \lceil K/2^{t+1} \rceil$ and construct μ_t from S_t according to Equation (1).
end for
Set $h_S(x)$ to be the unique element of $\mu_{\lfloor \log K \rfloor}(x)$ for all $x \in \mathcal{X}$.

where $n_t = |S_t|$, and $\mu^{L_t}_{S_{t,\le n}}$ is the output of the one-inclusion list learner of Charikar and Pabbaraju [2023], Hanneke et al. [2026] trained on the first n examples of S_t with list size L_t. The majority vote increases the list size to $2L_t - 1$, so with $L_t = \lceil K/2^{t+1} \rceil$, the effective list size remains on the order of $K/2^t$. Thus the algorithm reduces the candidate label set by a constant factor in each epoch. For completeness, we include the one-inclusion list learner in Algorithm 2 in the appendix. The precise epoch lengths $m_t - m_{t-1}$ are specified in the proof of Lemma 4.7.

4.1 Proof of Theorem 4.2

In this section, we will provide the analysis of Algorithm 1 and the proof of Theorem 4.2. We start with a technique lemma that upper bounds the L-exponential dimension by $(L/2)$-DS dimension.

Lemma 4.4 (Controlling the L-exponential dimension by $(L/2)$-DS dimension)**.** *For every $n \in \mathbb{N}$, $K \ge 2$, $L \ge 1$, and for every concept class $\mathcal{H} \subseteq [K]^n$ with $\lceil \mathrm{L}/2 \rceil\text{-}\,\mathrm{DS}(\mathcal{H}) = d_{\mathrm{DS}}^{\lceil L/2 \rceil}$ and $\mathrm{L}\text{-}\,\mathrm{E}(\mathcal{H}) = d_{\mathrm{E}}^{L} < \infty$, we have $d_{\mathrm{E}}^{L} \le 6 d_{\mathrm{DS}}^{\lceil L/2 \rceil} \log K$.*

Proof of Lemma 4.4. If $d_{DS}^{\lceil L/2 \rceil} = 0$ or $d_E^L = 0$, then the claim is immediate. If $d_E^L \le d_{DS}^{\lceil L/2 \rceil}$, then the claim is also immediate. Hence, assume

$$d_E^L > d_{DS}^{\lceil L/2 \rceil} \ge 1.$$

Consider the set S which realizes d_{E}^{L}, so that $|S| = d_{\mathrm{E}}^{L}$. By the definition of d_{E}^{L}, we have

$$|\mathcal{H}|_S| \ge (L+1)^{d_{\mathrm{E}}^{L}}.$$

On the other hand, by Theorem 2.1 of Hanneke et al. [2026],

$$|\mathcal{H}|_S| \le \left\lceil \frac{L}{2} \right\rceil^{d_{\mathrm{E}}^{L} - d_{\mathrm{DS}}^{\lceil L/2 \rceil}} \left(\frac{eK d_{\mathrm{E}}^{L}}{d_{\mathrm{DS}}^{\lceil L/2 \rceil}} \right)^{d_{\mathrm{DS}}^{\lceil L/2 \rceil}}.$$

Combining the preceding inequalities and proceeding as in Section C.1 gives

$$d_E^L \le 6 d_{DS}^{\lceil L/2 \rceil} \log K.$$

The full calculation is deferred to Section C.1. □

Based on Lemma 4.4 as well as Lemma 3.20 and Lemma 3.22 of Hanneke et al. [2026], we can upper bound the leave-one-out error of Algorithm 2 as follow.

Lemma 4.5 (Leave-one-out error bound for list learning)**.** *For every concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\lceil \mathrm{L}/2 \rceil\text{-}\,\mathrm{DS}(\mathcal{H}) = d_{\mathrm{DS}}^{\lceil L/2 \rceil}$, let $\mathcal{D}$ be an $\mathcal{H}$-realizable distribution and let $n > 0$ be an integer.*

Moreover, for every realizable sample $S = ((x_i, y_i))_{i=1}^n$,

$$\sum_{i=1}^{n} \mathbb{1}\left\{ y_i \notin \mu_{S^{-i}}^{L}(x_i) \right\} \le 6 d_{DS}^{\lceil L/2 \rceil} \log K,$$

where S^{-i} denotes S with its ith example removed. As a result, the prediction error can be bounded as

$$\Pr_{(S,(x,y))\sim\mathcal{D}^{n+1}}\left[y \notin \mu_S^L(x)\right] \leq \frac{6d_{\mathrm{DS}}^{\lceil L/2\rceil}\log K}{n+1},$$

where $\mu_S^L = \mathcal{A}_{\mathcal{H}}^L(S)$ is the output of Algorithm 2.

We now state and prove our list learning PAC bound, building on Lemma 4.5 and Theorem 2.1 in Aden-Ali et al. [2023].

Lemma 4.6. *Fix a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\lceil \mathrm{L}/2\rceil\text{-}\mathrm{DS}(\mathcal{H}) = d_{\mathrm{DS}}^{\lceil L/2\rceil}$. There exists a learning rule, depending only on $\mathcal{H}$ and L, that maps each realizable sample S to a list predictor $\mu = \mu(S)$ of size $\min\{2L-1, K\}$, such that for every $\mathcal{H}$-realizable distribution D, every integer $n \geq 1$, every $\delta \in (0,1)$, and $S \sim D^n$, the output satisfies*

$$\mathcal{L}_{\mathcal{D}}(\mu) \leq 14.46\left(\frac{6d_{\mathrm{DS}}^{\lceil L/2\rceil}\log K}{n} + \frac{1}{n}\log\left(\frac{2}{\delta}\right)\right),$$

with probability at least $1-\delta$ over the randomness of S.

Proof of Lemma 4.6. For $n < 4$, the claimed bound is trivial, so assume $n \geq 4$. Given the full training sample S, let $(\mu_{S_{\leq t}}^L)_{t=\lceil n/4\rceil}^{n-1}$ be the one-inclusion hypergraph predictors trained on the sequence of growing training samples $(S_{\leq t})_{t=\lceil n/4\rceil}^{n-1}$ output by Algorithm 2. Because these predictors are symmetric in their training sample and they have a leave-one-out error guarantee (Lemma 4.5), we can apply Theorem 2.1 in Aden-Ali et al. [2023] to the suffix of the sequence to get that

$$\frac{4}{3n}\sum_{t=\lceil n/4\rceil}^{n-1}\mathcal{L}_{\mathcal{D}}\left(\mu_{S_{\leq t}}^L\right) \leq 4.82\left(\frac{6d_{\mathrm{DS}}^{\lceil L/2\rceil}\log K}{n} + \frac{1}{n}\log\left(\frac{2}{\delta}\right)\right),$$

with probability at least $1-\delta$ over the randomness of S. Their proof applies with $\lceil n/4\rceil$ in place of $n/4$; the only change is that

$$\sum_{i=\lceil n/4\rceil+1}^{n}\frac{1}{i} \leq \ln\frac{n}{\lceil n/4\rceil} \leq \ln 4.$$

Whenever this bound holds, the majority vote of the one-inclusion hypergraph predictors in the suffix, $\mu = \mathrm{Maj}\left(\mu_{S_{\leq\lceil n/4\rceil}}^L, \ldots, \mu_{S_{\leq n-1}}^L\right)$ satisfies

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}}(\mu) &= \mathbb{E}_{(x,y)\sim D}\left[\mathbb{1}\left[y \notin \mu(x)\right]\right] \\
&\leq \mathbb{E}_{(x,y)\sim D}\left[3\left(\frac{4}{3n}\sum_{t=\lceil n/4\rceil}^{n-1}\mathbb{1}\left[y \notin \mu_{S_{\leq t}}^L(x)\right]\right)\right] \\
&= 3\cdot\frac{4}{3n}\sum_{t=\lceil n/4\rceil}^{n-1}\mathcal{L}_{\mathcal{D}}\left(\mu_{S_{\leq t}}^L\right) \\
&\leq 14.46\left(\frac{6d_{\mathrm{DS}}^{\lceil L/2\rceil}\log K}{n} + \frac{1}{n}\log\left(\frac{2}{\delta}\right)\right).
\end{aligned}$$

In the first inequality above, we used the fact that when the majority vote is wrong, at least half of the predictors in the suffix are wrong, and the suffix contains $n - \lceil n/4\rceil \geq n/2$ predictors. This completes the proof. □

The proof is given by inductively proving the following lemma.

Lemma 4.7. *With probability at least $1-(t\delta)/\log K$, the list predictor μ_t in Equation (1) has error rate at most $(t\epsilon)/\log K$ for $t \in \{0, 1, \ldots, \lfloor \log K\rfloor\}$.*

Proof of Lemma 4.7. For the base case, when $t = 0$, by definition we know that $\mu_0(x) = \mathcal{Y}$ for any $x \in \mathcal{X}$, which satisfies our inductive assumption.

For the inductive step, we assume that $\mathcal{L}_{\mathcal{D}}(\mu_{t-1}) \leq ((t-1)\epsilon)/\log K$ holds with probability at least $1 - ((t-1)\delta)/\log K$. According to Algorithm 1, we predict the label $\hat{y}_i$ uniformly randomly from the list $\mu_{t-1}(x_i)$ for each instance x_i we encountered in epoch t, we know that

$$\Pr[\hat{y}_i = y_i] \geq \frac{1}{2L_{t-1}} \left(1 - \frac{(t-1)\epsilon}{\log K}\right).$$

Also, notice that, the number of samples we encountered in epoch t is

$$m_t - m_{t-1} := \left\lceil \frac{4L_{t-1}}{1 - \frac{(t-1)\epsilon}{\log K}} \left(15 \frac{\left(6d_{DS}^{\lceil L_t/2 \rceil} \log K + \log\left(\frac{4\log K}{\delta}\right)\right)\log K}{\epsilon} + \log\left(\frac{2\log K}{\delta}\right)\right)\right\rceil.$$

Conditioned on the history before epoch t, the indicators $\mathbb{1}[\hat{y}_i = y_i]$ are independent Bernoulli random variables, a multiplicative lower-tail Chernoff bound yields that, with probabilty at least $1 - \delta/(2\log K)$ we can collect a labeled sample S_t whose size $n_t = |S_t|$ satisfies

$$n_t \geq 15 \frac{\left(6d_{DS}^{\lceil L_t/2 \rceil} \log K + \log\left(\frac{4\log K}{\delta}\right)\right)\log K}{\epsilon}.$$

Since $\mu_{t-1}(x)$ is padded to have the same size for every x, conditional on $y \in \mu_{t-1}(x)$, every example is retained with the same probability. Hence, conditional on μ_{t-1} and on $n_t = |S_t|$, the retained examples in S_t are i.i.d. from

$$\mathcal{D}_t := \mathcal{D} \mid \{(x, y) : y \in \mu_{t-1}(x)\}.$$

By Lemma 4.6, with probability at least $1 - \delta/(2\log K)$,

$$\mathcal{L}_{\mathcal{D}_t}(\mu_t) \leq \frac{\epsilon}{\log K}.$$

Moreover,

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}}(\mu_t) &= \Pr[y \notin \mu_t(x),\, y \in \mu_{t-1}(x)] + \Pr[y \notin \mu_t(x),\, y \notin \mu_{t-1}(x)] \\
&\leq \Pr[y \in \mu_{t-1}(x)]\, \mathcal{L}_{\mathcal{D}_t}(\mu_t) + \mathcal{L}_{\mathcal{D}}(\mu_{t-1}) \\
&\leq \mathcal{L}_{\mathcal{D}_t}(\mu_t) + \mathcal{L}_{\mathcal{D}}(\mu_{t-1}) \\
&\leq \frac{\epsilon}{\log K} + \frac{(t-1)\epsilon}{\log K} = \frac{t\epsilon}{\log K}.
\end{aligned}$$

Condition on the event from the induction hypothesis, which holds with probability at least $1 - (t-1)\delta/\log K$. The Chernoff step and the list-learning PAC step each fail with probability at most $\delta/(2\log K)$. A union bound therefore gives the desired probability $1 - t\delta/\log K$. □

Now, based on Lemma 4.7, we start to prove Theorem 4.2.

Proof of Theorem 4.2. The case $K = 1$ is trivial, so throughout the proof assume $K \geq 2$. By Lemma 4.7, we know that with probability at least $1 - \delta$, the hypothesis h_S returned by Algorithm 1 has error at most ϵ. Thus the sample complexity $\mathbf{m}_{\mathcal{H}}^{B}(\epsilon, \delta)$ can be upper bounded by the total number of samples used by Algorithm 1. Formally, we have

$$\begin{aligned}
\mathbf{m}_{\mathcal{H}}^{B}(\epsilon, \delta) &\leq \sum_{t=1}^{\lfloor \log(K) \rfloor} (m_t - m_{t-1}) \\
&\leq O\left(\frac{\max_{1 \leq L \leq K-1} \left(L d_{DS}^{L}\right) (\log K)^3}{\epsilon} + \frac{K \log K \log\left(\frac{2\log K}{\delta}\right)}{\epsilon} \right).
\end{aligned}$$

By the definition of the Bandit-DS dimension, for any concept class $\mathcal{H}$,

$$\mathrm{BDS}(\mathcal{H}) \geq \max_{1 \leq L \leq K-1} \left(L \cdot \text{L-DS}(\mathcal{H})\right).$$

Combining the preceding inequalities proves the desired upper bound. The full calculation is deferred to Section C.2. □

5 Conclusion and Future Directions

In this work, we introduce the bandit DS dimension, a new combinatorial complexity measure for multiclass PAC learning with bandit feedback. In terms of this dimension, we establish a near-optimal characterization of the realizable sample complexity, resolving, up to logarithmic factors, a longstanding open question posed by Daniely et al. [2011]. Our upper bound is obtained via LISTCASCADE, a new algorithmic framework that connects bandit learning with list learning by using list learners as intermediate predictors in a cascade, progressively reducing the ambiguity in the candidate labels until a single-label predictor is obtained. Along the way, we also prove a refined sample complexity bound for list PAC learning, which may be of independent interest. Two natural directions are particularly important: extending the approach to the agnostic setting, and determining whether the remaining logarithmic factors are necessary. In particular, it remains open whether every concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $\mathrm{BDS}(\mathcal{H}) < \infty$ admits a bandit-feedback PAC learner with sample complexity $O((\mathrm{BDS}(\mathcal{H}) + \log(1/\delta))/\epsilon)$.

Acknowledgments and Disclosure of Funding

We thank the anonymous NeurIPS 2026 reviewers for their helpful feedback, which led to several minor changes in the paper.

Steve Hanneke acknowledges support by grant no. 2024243 from the United States - Israel Binational Science Foundation (BSF). Shay Moran acknowledges support by Israel by ISF grant 2362/26, 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.

References

Ishaq Aden-Ali, Yeshwanth Cherapanamjeri, Abhishek Shetty, and Nikita Zhivotovskiy. Optimal pac bounds without uniform convergence. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1203–1223. IEEE, 2023.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1638–1646, Bejing, China, 22–24 Jun 2014. PMLR.

Alekh Agarwal, Sarah Bird, Markus Cozowicz, Luong Hoang, John Langford, Stephen Lee, Jiaji Li, Daniel R. Melamed, Gal Oshri, Oswaldo Ribas, Siddhartha Sen, and Alex Slivkins. Making contextual decisions with low technical debt. *arXiv: Learning*, 2016.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002a.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002b.

Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, page 671–680, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580470. doi: 10.1145/1374376.1374474.

Maria Florina Balcan, Heiko Röglin, and Shang-Hua Teng. Agnostic clustering. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory*, ALT'09, page 384–398, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 3642044131.

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

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.

Nataly Brukhim, Steve Hanneke, and Shay Moran. Improper multiclass boosting. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 5433–5452. PMLR, 12–15 Jul 2023.

Nataly Brukhim, Amit Daniely, Yishay Mansour, and Shay Moran. Multiclass boosting: simple and intuitive weak learning criteria. *Advances in Neural Information Processing Systems*, 36, 2024.

Nataly Brukhim, Aldo Pacchiano, Miroslav Dudik, and Robert Schapire. On the hardness of bandit learning, 2025.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. *Regret analysis of stochastic and nonstochastic multi-armed bandit problems*. Now Publishers, Inc., 2012.

Moses Charikar and Chirag Pabbaraju. A characterization of list learnability. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1713–1726, 2023.

Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th annual ACM SIGACT symposium on theory of computing*, pages 47–60, 2017.

Moses Charikar, Chirag Pabbaraju, and Ambuj Tewari. A characterization of list language identification in the limit, 2025.

Alon Cohen, Liad Erez, Steve Hanneke, Tomer Koren, Yishay Mansour, Shay Moran, and Qian Zhang. Sample complexity of agnostic multiclass classification: Natarajan dimension strikes back. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC 2026)*, 2026. To appear.

Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *Conference on Learning Theory*, pages 287–316. PMLR, 2014.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 207–232. JMLR Workshop and Conference Proceedings, 2011.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. *J. Mach. Learn. Res.*, 16(1):2377–2404, 2015.

Ilias Diakonikolas and Daniel M Kane. *Algorithmic high-dimensional robust statistics*. Cambridge university press, 2023.

Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. *arXiv preprint arXiv:1106.2369*, 2011.

Liad Erez and Tomer Koren. From contextual combinatorial semi-bandits to bandit list classification: Improved sample complexity with sparse rewards. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Liad Erez, Alon Cohen, Tomer Koren, Yishay Mansour, and Shay Moran. Fast rates for bandit PAC multiclass classification. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Liad Erez, Alon Cohen, Tomer Koren, Yishay Mansour, and Shay Moran. The real price of bandit information in multiclass classification. In *Annual Conference Computational Learning Theory*, 2024b.

Dylan J. Foster, Sham M. Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making, 2023.

Steve Hanneke and Amirreza Shaeiri. A trichotomy for list transductive online learning. In *Forty-second International Conference on Machine Learning*, 2025.

Steve Hanneke and Liu Yang. Bandit learnability can be undecidable. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 5813–5849. PMLR, 12–15 Jul 2023.

Steve Hanneke, Shay Moran, and Waknine Tom. List sample compression and uniform convergence. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 2360–2388. PMLR, 2024a.

Steve Hanneke, Shay Moran, and Qian Zhang. Improved sample complexity for multiclass PAC learning. *Advances in Neural Information Processing Systems*, 37:42798–42839, 2024b.

Steve Hanneke, Amirreza Shaeiri, and Hongao Wang. For universal multiclass online learning, bandit feedback and full supervision are equivalent. In *36th International Conference on Algorithmic Learning Theory*, 2024c.

Steve Hanneke, Qinglin Meng, and Amirreza Shaeiri. Representation preserving multiclass agnostic to realizable reduction. In *Forty-second International Conference on Machine Learning*, 2025a.

Steve Hanneke, Shay Moran, Hilla Schefler, and Iska Tsubari. Private list learnability vs. online list learnability. In Nika Haghtalab and Ankur Moitra, editors, *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 5173–5213. PMLR, 30 Jun–04 Jul 2025b.

Steve Hanneke, Amirreza Shaeiri, and Hongao Wang. Non-uniform multiclass learning with bandit feedback. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025c.

Steve Hanneke, Amirreza Shaeiri, and Qian Zhang. Universal rates for multiclass learning with bandit feedback. *Proceedings of Machine Learning Research vol*, 291:1–53, 2025d.

Steve Hanneke, Qinglin Meng, Shay Moran, and Amirreza Shaeiri. An optimal sauer lemma over k-ary alphabets, 2026. URL https://arxiv.org/abs/2604.12952.

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

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Shay Moran, Ohad Sharon, Iska Tsubari, and Sivan Yosebashvili. List online classification. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 1885–1913. PMLR, 2023.

Chirag Pabbaraju and Sahasrajit Sarmasarkar. A characterization of list regression, 2024.

Herbert E. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

Benjamin Rubinstein, Peter Bartlett, and J Rubinstein. Shifting, one-inclusion mistake bounds and tight multiclass expected risk bounds. *Advances in Neural Information Processing Systems*, 19, 2006.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019. ISSN 1935-8237. doi: 10.1561/2200000068.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Michael Woodroofe. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74(368):799–806, 1979.

A Proof of Theorem 4.3

In this section, we prove Theorem 4.3 by constructing two hard cases.

proof of Theorem 4.3. We first show that we at least need $\Omega(\mathrm{BDS}(\mathcal{H})/\epsilon)$ samples. For a concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, let $\mathrm{BDS}(\mathcal{H}) = d^B_{DS} < \infty$. By non-triviality, choose a BDS-shattered pair $Z = (x_1, \ldots, x_d) \in \mathcal{X}^d$ and $N \in \mathbb{N}^d$ with $d \geq 2$ and $\sum_{i=1}^d N_i = d^B_{\mathrm{DS}}$. Without loss of generality, we assume N_i's are in ascending order. Since $N_1 \leq N_2$, we have

$$\sum_{i=2}^{d} N_i \geq \frac{1}{2} d^B_{DS}.$$

We consider the distribution $\mathcal{D}$ over $\mathcal{X}$ as follows

$$p(x) = \begin{cases} 1 - 16\epsilon, & \text{if } x = x_1 \\ \frac{16 N_i \epsilon}{\sum_{i=2}^d N_i}, & \text{if } x \in \{x_2, \cdots, x_d\} \end{cases}$$

Let $\mathcal{F} \subseteq \mathcal{H}$, $|\mathcal{F}| < \infty$, witness the BDS-shattering of $\mathcal{Z}$, and let $V = \mathcal{F}|_{\mathcal{Z}}$. For each $v \in V$, fix one representative $c_v \in \mathcal{F}$ with $c_v|_{\mathcal{Z}} = v$. We choose v^* uniformly at random from V and set $c^* = c_{v^*}$.

For simplicity, let $\mathcal{Z}' = \{x_2, \cdots, x_d\}$. Let $m = (\sum_{i=2}^d N_i)/(64\epsilon)$ and $\mathcal{A}$ be any algorithm that uses an i.i.d. sample S with size at most m before picking a hypothesis. For ease of presentation, we define $\mathcal{L}'_{\mathcal{D}}(h) = \Pr[c^*(x) \neq h(x) \quad \text{and} \quad x \in \mathcal{Z}']$. In expectation, for each point $x_i \in \mathcal{Z}'$, there will be at most $N_i/4$ samples sampled from x_i. By Markov inequality, with probability at most $1/2$, there will be more than $N_i/2$ samples sampled from x_i. Let A_i be the event that no more than $N_i/2$ samples are sampled from x_i. In other words, we have $\Pr_S[A_i] \geq 1/2$. For the random target $c^* = c_{v^*}$ chosen above, we have

$$\begin{aligned} \mathbb{E}_{c^*,S}\left[\mathcal{L}'_{\mathcal{D}}(h)\right] &= \sum_{i=2}^{d} \Pr[x = x_i] \Pr_{c^*,S}[h(x_i) \neq y_i] \\ &\geq \sum_{i=2}^{d} \Pr[x = x_i] \Pr_{c^*,S}[h(x_i) \neq y_i | A_i] \Pr_S[A_i] \\ &\geq \frac{1}{8} \sum_{i=2}^{d} \frac{16 N_i \epsilon}{\sum_{i=2}^d N_i} = 2\epsilon. \end{aligned}$$

The first inequality is based on the law of total probability. For the second inequality, fix $i \geq 2$ and condition on all coordinates of $c^*|_{\mathcal{Z}}$ except the ith. Condition also on the learner's internal random seed and on the unlabeled sample. The label $c^*(x_i)$ is then uniform among at least $N_i + 1$ possible labels. On the event A_i, the learner has made at most $N_i/2$ guesses at x_i, so even including its final prediction, its conditional probability of error is at least

$$1 - \frac{N_i/2 + 1}{N_i + 1} = \frac{N_i}{2(N_i + 1)} \geq \frac{1}{4}.$$

This implies that there is some $c^* \in \mathcal{H}$ such that $\mathbb{E}_S\left[\mathcal{L}'_{\mathcal{D}}(h)\right] \geq 2\epsilon$. For the remainder of the proof, consider this c^*. Note that, by definition, we have $\mathcal{L}'_{\mathcal{D}}(h) \leq \Pr[x \in \mathcal{Z}'] = 16\epsilon$. Let

$$q = \Pr[\mathcal{L}'_{\mathcal{D}}(h) > \epsilon],$$

where the probability also includes the internal randomness of the learner. Since $0 \leq \mathcal{L}'_{\mathcal{D}}(h) \leq 16\epsilon$, we have

$$2\epsilon \leq \mathbb{E}_S[\mathcal{L}'_{\mathcal{D}}(h)] \leq 16\epsilon q + \epsilon(1 - q),$$

and hence $q \geq 1/15$. Consequently, for every algorithm using at most

$$m = \frac{\sum_{i=2}^d N_i}{64\epsilon}$$

samples, there exist a distribution $\mathcal{D}$ and a target $c^* \in \mathcal{H}$ such that

$$\Pr[\mathcal{L}_{\mathcal{D}}(h) > \epsilon] \geq \frac{1}{15}.$$

Since

$$\sum_{i=2}^{d} N_i \geq \frac{1}{2}\,\mathrm{BDS}(\mathcal{H})$$

and $\delta \leq 1/16 < 1/15$, this proves the $\Omega(\mathrm{BDS}(\mathcal{H})/\epsilon)$ lower bound.

Then, we show that at least $\Omega(\log(1/\delta)/\epsilon)$ samples are necessary. Following the proof technique of Blumer et al. [1989], we provide the argument here for completeness. Since $d \geq 2$, by taking an i-neighbor pair from the BDS witness and any other coordinate, there exist two points x_1, x_2 and two hypotheses $h_1, h_2 \in \mathcal{H}$ such that h_1, h_2 agree on x_1 but disagree on x_2. We consider the distribution $\mathcal{D}$ over $\mathcal{X}$ as follows

$$p(x) = \begin{cases} 1 - 2\epsilon, & \text{if } x = x_1 \\ 2\epsilon, & \text{if } x = x_2. \end{cases}$$

Let m be the sample size and $\mathcal{A}$ be any algorithm that uses an i.i.d. sample S with size at most m before picking a hypothesis. Let B be the event that m samples are not sampled from x_2. If the target is chosen uniformly from $\{h_1, h_2\}$, then conditioned on B, the learner receives the same transcript under the two targets. Therefore,

$$\frac{1}{2}\sum_{j=1}^{2} \Pr_{h_j}[\mathcal{L}_{\mathcal{D}}(h) > \epsilon] \geq \frac{1}{2}\Pr_{S}[B] \geq \frac{1}{2}(1-2\epsilon)^m.$$

Hence, for at least one of the two targets,

$$\Pr[\mathcal{L}_{\mathcal{D}}(h) > \epsilon] \geq \frac{1}{2}(1-2\epsilon)^m.$$

Thus, for $\epsilon \leq 1/16$ and $\delta \leq 1/4$, achieving failure probability at most δ requires

$$m = \Omega\left(\frac{\log(1/\delta)}{\epsilon}\right).$$

Combining the two lower bounds proves the theorem. □

B Definitions

In this section, we provide the auxilary formal definitions that will be used in our paper. We start from the notations.

B.1 Notations

In this subsection, we present the basic notation used in the paper; all of it is standard in the literature and is included for completeness. Let $\mathbb{N}$ and $\mathbb{R}$ stand for the set of natural numbers and real numbers, respectively. We denote by $\bar{\mathbb{N}}$ the extended natural number system defined as $\bar{\mathbb{N}} := \mathbb{N} \cup \{+\infty\}$. In addition, for a given $n \in \mathbb{N}$, we use $[n]$ to denote $\{1, 2, \ldots, n\}$. Next, given $n \in \mathbb{N}$, for any sequence of size n or n-tuple v, and any $i \in [n]$, let us use v_i to denote the i-th element in v. Afterward, we denote by $A \times B$ the Cartesian product of two arbitrary sets A and B. In addition, for any set A and any $n \in \mathbb{N}$, we let A^n indicate n times the Cartesian product of A with itself. Note that for any set A, we define $A^0 := \{()\}$,. Also, given a set A, we denote by A^* the set of all finite sequences of members of A; more formally, $A^* := \bigcup_{T=0}^{\infty} A^T$. Then, for arbitrary sets X and Y, we use Y^X to denote the space of all functions from X to Y. Additionally, let $\mathcal{F} \subseteq Y^X$ for the arbitrary sets X and Y. Given a subset $S \subseteq X$, we define $\mathcal{F}|_S$ as follows: $\mathcal{F}|_S := \{g \mid g \in Y^S,\ \exists_{f \in \mathcal{F}}\, \forall_{x \in S}\, g(x) = f(x)\}$. Similarly, give a sequence $S \in X^*$, we define $\mathcal{F}|_S$ as follows: $\mathcal{F}|_S := \{g \mid g \in Y^{|S|},\ \exists_{f \in \mathcal{F}}\, \forall_{i \in [|S|]}\, g_i = f(S_i)\}$. Going further, let e and $\log(\cdot)$ stand for Euler's number and the Logarithm function in the base 2, respectively. Finally, we use $\mathcal{O}(\cdot)$, $\Omega(\cdot)$, and $\Theta(\cdot)$ to denote the standard asymptotic notation in theoretical computer science. Furthermore, we use $\widetilde{O}(\cdot)$, $\widetilde{\Omega}(\cdot)$, and $\widetilde{\Theta}(\cdot)$ to hide poly-logarithmic factors.

B.2 Combinatorial Complexity Parameters

In this subsection, we present the combinatorial complexity parameters considered in this work. For each parameter, we begin by defining the associated notion of a shattered sequence or shattered set, and then define the parameter itself in terms of that notion. Before proceeding, we state the following definition.

Definition B.1 (i-neighbor)**.** Let $\mathscr{Y}$ be a non-empty set. Let $f, g \in \mathscr{Y}^d$ for some $d \in \mathbb{N}$. For every $i \in [d]$, we say that f and g are i-neighbors if $f_i \neq g_i$ and $\forall_{j \in [d] \setminus \{i\}}\ f_j = g_j$.

We adopt the convention that each dimension defined below is equal to 0 when there is no positive-dimensional witness.

Next, we formally state the definition of the bandit DS dimension.

Definition B.2 (BDS-shattered sequence)**.** Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a concept class. Let $S \in \mathcal{X}^d$ be a sequence of instances for some $d \in \mathbb{N}$. Also, let $\mathrm{N} \in \mathbb{N}^d$. We say that S and N is BDS-*shattered* by $\mathcal{H}$, if there exists $\emptyset \neq F \subseteq \mathcal{H}, |F| < \infty$ such that for all $f \in F|_S$ and for all $i \in [d]$, f has at least N_i distinct i-neighbor in $F|_S$.

Definition B.3 (Bandit DS dimension)**.** Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a concept class. The *Bandit* DS *dimension* of $\mathcal{H}$, denoted by $\mathrm{BDS}(\mathcal{H}) \in \bar{\mathbb{N}} \cup \{0\}$, is defined as the $\sup_{d \in \mathbb{N} \cup \{0\}}$ such that there exist a sequence of instances $S \in \mathcal{X}^{d'}$ for some $d' \in \mathbb{N}$ and $\mathrm{N} \in \mathbb{N}^{d'}$ such that $\sum_{i=1}^{d'} N_i = d$ that is BDS-shattered by $\mathcal{H}$.

Next, we define combinatorial complexity parameters related to list PAC learning. Notably, when the list size is equal to one, each of these dimensions coincides with its corresponding counterpart in the multiclass setting. We begin with the definition of the DS_{L} dimension.

Definition B.4 (DS_{L}-shattered sequence)**.** Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a concept class, and $\mathrm{L} \in \mathbb{N}$. Let $S \in \mathcal{X}^d$ be a sequence of instances for some $d \in \mathbb{N}$. We say that S is DS_{L}-*shattered* by $\mathcal{H}$, if there exists $\emptyset \neq F \subseteq \mathcal{H}, |F| < \infty$ such that for all $f \in F|_S$ and for all $i \in [d]$, f has at least L distinct i-neighbor in $F|_S$.

Definition B.5 (DS_{L} dimension Charikar and Pabbaraju [2023])**.** Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a concept class, and $\mathrm{L} \in \mathbb{N}$. The DS_{L} *dimension* of $\mathcal{H}$, denoted by $\mathrm{DS}_{\mathrm{L}}(\mathcal{H}) \in \bar{\mathbb{N}} \cup \{0\}$, is defined as the $\sup_{d \in \mathbb{N} \cup \{0\}}$ such that there exists a sequence of instances $S \in \mathcal{X}^d$ that is DS_{L}-shattered by $\mathcal{H}$.

Next, we introduce the L-Exponential dimension. This combinatorial complexity parameter is particularly useful in shifting arguments.

Definition B.6 (E_{L}-shattered set)**.** Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a concept class, and $\mathrm{L} \in \mathbb{N}$. Let $S \subseteq \mathcal{X}$ be a set of instances of size d for some $d \in \mathbb{N}$. We say that S is E_{L}-*shattered* by $\mathcal{H}$, if $\left|\mathcal{H}|_S\right| \geq (\mathrm{L}+1)^d$.

Definition B.7 (L-Exponential dimension Charikar and Pabbaraju [2023])**.** Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a concept class, and $\mathrm{L} \in \mathbb{N}$. The L-*Exponential dimension* of $\mathcal{H}$, denoted by $\mathrm{E}_{\mathrm{L}}(\mathcal{H}) \in \bar{\mathbb{N}} \cup \{0\}$, is defined as the $\sup_{d \in \mathbb{N} \cup \{0\}}$ such that there exists a set of instances $S \subseteq \mathcal{X}$ of size d that is E_{L}-shattered by $\mathcal{H}$.

B.3 Multiclass Learning with Bandit Feedback Algorithms

To improve readability, we use a comma to separate indices when multiple indices are involved. For example, let v be a sequence of length 5 whose elements are 2-tuples. We denote by $v_{5,1}$ the first component of the 5-th element of v. In addition, letting $m, n \in \mathbb{N}$ be such that $m \leq n$, we write $\left[(x_i, y_i, z_i)\right]_{i=m}^{n}$ to denote $\big((x_m, y_m, z_m), (x_{m+1}, y_{m+1}, z_{m+1}), \ldots, (x_n, y_n, z_n)\big)$. This section is similar in spirit to Hanneke et al. [2025d].

Let $\Sigma := \mathcal{Y} \times \{0,1\}$. Let $A = (A_1, A_2)$ be a multiclass PAC learning algorithm with bandit feedback, where A_1 specifies the conditional distribution of the next predicted label given the transcript and current instance, and A_2 specifies the conditional distribution of the final hypothesis given the transcript. For any $\mathfrak{u} \in \big(\mathcal{X} \times \Sigma\big)^*$ and any $x \in \mathcal{X}$, let us write $\mathbf{A}_1(x; \mathfrak{u})$ to denote $\mathbf{A}_1\big((\mathfrak{u}, x)\big)$. Also, for every finite sequence of examples $\mathcal{S} \in \mathcal{Z}^n = (\mathcal{X} \times \mathcal{Y})^n$ of size n for some $n \in \mathbb{N}$, we denote by $\mathbf{B}_{\mathbf{A}}(\mathcal{S})$ the random variable taking values in $(\mathcal{X} \times \Sigma)^n$ representing the first part of the input of $\mathbf{A}_1$ in round $n+1$ when the choices of nature are obtained according to $\mathcal{S}$, which is recursively defined as follows:

$$\mathbf{B}_{\mathbf{A}}(\mathcal{S})_{i,1} := \mathcal{S}_{i,1}, \quad 1 \leq i \leq n$$

$$\mathbf{B}_{\mathbf{A}}(\mathcal{S})_{i,2} := \begin{cases} y \sim \mathbf{A}_1\big(\mathcal{S}_{1,1}; \emptyset\big), & i = 1 \\ y \sim \mathbf{A}_1\big(\mathcal{S}_{i,1}; \big[\big(\mathbf{B}_{\mathbf{A}}(\mathcal{S})_{j,1}, \mathbf{B}_{\mathbf{A}}(\mathcal{S})_{j,2}, \mathbf{B}_{\mathbf{A}}(\mathcal{S})_{j,3}\big)\big]_{j=1}^{i-1}\big), & 1 < i \leq n \end{cases}$$

$$\mathbf{B}_{\mathbf{A}}(\mathcal{S})_{i,3} := \mathbb{1}\big\{\mathbf{B}_{\mathbf{A}}(\mathcal{S})_{i,2} = \mathcal{S}_{i,2}\big\}, \quad 1 \leq i \leq n$$

Building upon that, for every finite sequence of examples $S \in \mathcal{Z}^n = (\mathcal{X} \times \mathcal{Y})^n$, we denote by $\widehat{h}_A^S$ the random output hypothesis defined by

$$\widehat{h}_A^S \sim A_2(B_A(S)).$$

Thus, $\widehat{h}_A^S$ is a single random element of $\mathcal{Y}^{\mathcal{X}}$.

B.4 One-inclusion Graphs

We now introduce an important combinatorial object associated with a hypothesis class, namely, the one-inclusion hypergraph.

Definition B.8 (One-inclusion hypergraph Haussler et al. [1994], Rubinstein et al. [2006])**.** The one-inclusion hypergraph of $\mathcal{H} \subseteq \mathcal{Y}^m$ for some $m \in \mathbb{N}$ is a hypergraph $G(\mathcal{H}) = (V, E)$ that is defined as follows. The vertex set is $V = \mathcal{H}$. For each $i \in [m]$ and $f : [m] \setminus \{i\} \to \mathcal{Y}$, let $e_{i,f}$ be the set of all $h \in \mathcal{H}$ that agree with f on $[m] \setminus \{i\}$. The edge set is defined as follows:

$$E = \{e_{i,f} \mid i \in [m], f : [m] \setminus \{i\} \to \mathcal{Y}, e_{i,f} \neq \emptyset\}.$$

We regard the hyperedges as an indexed family: edges with distinct indices (i, f) remain distinct even when their vertex sets coincide.

Moreover, we say that the edge $e_{i,f} \in E$ is in the direction i, and is adjacent to the hypothesis/vertex h if $h \in e_{i,f}$. In addition, every vertex $h \in V$ is adjacent to exactly m edges. Further, the size of the edge $e_{i,f}$ is the size of the set $|e_{i,f}|$.

For the one-inclusion hypergraph, we study the degrees of its vertices and its average degree.

Definition B.9 (L-degree Charikar and Pabbaraju [2023])**.** Let $G(\mathcal{H}) = (V, E)$ be the one-inclusion hypergraph of $\mathcal{H} \subseteq \mathcal{Y}^m$ for some $m \in \mathbb{N}$. The L-degree of a vertex $v \in V$ is defined as follows:

$$\mathsf{deg}^L(v) = |\{e \in E : v \in e, |e| > L\}|.$$

Definition B.10 (Average L-degree Charikar and Pabbaraju [2023])**.** Let $G(\mathcal{H}) = (V, E)$ be the one-inclusion hypergraph of $\mathcal{H} \subseteq \mathcal{Y}^m$ for some $m \in \mathbb{N}$. The average L-degree of $\mathcal{H}$ is defined as follows:

$$\mathsf{avd}^L(\mathcal{H}) := \frac{1}{|V|} \sum_{v \in V} \mathsf{deg}^L(v) = \frac{1}{|V|} \sum_{e \in E : |e| > L} |e|.$$

Next, we restate the definition of a list orientation of the one-inclusion hypergraph from Charikar and Pabbaraju [2023]. This notion captures the behavior of a deterministic learning algorithm that receives a labeled sample together with an unlabeled test point and outputs a prediction for that test point.

Definition B.11 (List Orientation Charikar and Pabbaraju [2023])**.** Let $G(\mathcal{H}) = (V, E)$ be the one-inclusion hypergraph of $\mathcal{H} \subseteq \mathcal{Y}^m$ for some $m \in \mathbb{N}$. A list orientation σ^L of $G(\mathcal{H}) = (V, E)$ having list size L is a mapping $\sigma^L : E \to \{V' \subseteq V : |V'| \leq L\}$ such that for each edge $e \in E$, $\sigma^L(e) \subseteq e$.

For a list orientation, we define its L-outdegree.

Definition B.12 (L-outdegree of a list orientation Charikar and Pabbaraju [2023])**.** Let $G(\mathcal{H}) = (V, E)$ be the one-inclusion hypergraph of $\mathcal{H} \subseteq \mathcal{Y}^m$ for some $m \in \mathbb{N}$, and let σ^L be a L-list orientation of it. The L-outdegree of $v \in V$ in σ^L is defined as follows:

$$\mathsf{outdeg}^L(v; \sigma^L) := |\{e : v \in e, v \notin \sigma^L(e)\}|.$$

The maximum L-outdegree of σ^L is

$$\mathsf{outdeg}^L(\sigma^L) := \sup_{v \in V} \mathsf{outdeg}^L(v; \sigma^L).$$

C Deferred Proofs

Next, we present Algorithm 2, which is the subroutine of Algorithm 1, corresponding to the one-inclusion list learners in Charikar and Pabbaraju [2023], Hanneke et al. [2026]. For the sake of completeness, we restate the algorithm below.

Algorithm 2 The one-inclusion list algorithm $\mathcal{A}^L_{\mathcal{H}}$ for $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ of list size L.

Input: A sample $S = \big((x_1, y_1), \cdots, (x_m, y_m)\big)$ realizable by $\mathcal{H}$ and list size L.
Output: A L-list hypothesis $\mathcal{A}^L_{\mathcal{H}}(S) = \mu^L_S : \mathcal{X} \to \{Y \subseteq \mathcal{Y} : |Y| \leq L\}$.

For each $x \in \mathcal{X}$, the L-list $\mu^L_S(x)$ is computed as follows.

1: Consider the class $\mathcal{H}' \subseteq \mathcal{Y}^{m+1}$ of all patterns over the *unlabeled data* that are realizable by $\mathcal{H}$.
2: Find a L-list orientation σ^L of $G(\mathcal{H}')$ that *minimizes* the *maximum* L-outdegree.
3: Consider the edge in direction $m+1$ defined by S:

$$e = \{h \in \mathcal{H}' : \forall i \in [m] \;\; h(i) = y_i\}.$$

4: Set $\mu^L_S(x) = \{h(m+1) : h \in \sigma^L(e)\}$.

In Algorithm 2, repeated training examples are deleted before constructing the hypergraph, and m denotes the remaining sample size. The orientation in line 2 is the fixed orientation for $U = \{x_1, \ldots, x_m, x\}$.

For every finite unlabeled set $U \subseteq \mathcal{X}$, we fix once and for all a minimizing L-list orientation of $G(\mathcal{H}|_U)$. The chosen orientation depends only on U, and not on the order in which its elements are presented or on the distinguished query point. If the query point already appears in the training sample, the learner outputs an arbitrary L-list containing its observed label.

Next we show the complete calculation in the proof of Lemma 4.4 and Theorem 4.2

C.1 Complete calculation in the proof of Lemma 4.4

Proof of Lemma 4.4. If $d^{\lceil L/2 \rceil}_{DS} = 0$ or $d^L_E = 0$, then the claim is immediate. If $d^L_E \leq d^{\lceil L/2 \rceil}_{DS}$, then the claim is also immediate. Hence, assume

$$d^L_E > d^{\lceil L/2 \rceil}_{DS} \geq 1.$$

Consider the set S which realizes d^L_{E}, so that $|S| = d^L_{\mathrm{E}}$. Then we have that

$$|\mathcal{H}|_S| \geq (L+1)^{d^L_{\mathrm{E}}}.$$

On the other hand, by Theorem 2.1 of Hanneke et al. [2026], we know that

$$|\mathcal{H}|_S| \leq \left\lceil \frac{L}{2} \right\rceil^{d^L_{\mathrm{E}} - d^{\lceil L/2 \rceil}_{\mathrm{DS}}} \left(\frac{eK d^L_{\mathrm{E}}}{d^{\lceil L/2 \rceil}_{\mathrm{DS}}} \right)^{d^{\lceil L/2 \rceil}_{\mathrm{DS}}}.$$

Combining the above two inequalities, we have

$$(L+1)^{d^L_{\mathrm{E}}} \leq \left\lceil \frac{L}{2} \right\rceil^{d^L_{\mathrm{E}} - d^{\lceil L/2 \rceil}_{\mathrm{DS}}} \left(\frac{eK d^L_{\mathrm{E}}}{d^{\lceil L/2 \rceil}_{\mathrm{DS}}} \right)^{d^{\lceil L/2 \rceil}_{\mathrm{DS}}}.$$

Note that $\lceil L/2 \rceil \leq \frac{L+1}{2}$.

$$(L+1)^{d^L_{\mathrm{E}}} \leq \left(\frac{L+1}{2} \right)^{d^L_{\mathrm{E}} - d^{\lceil L/2 \rceil}_{\mathrm{DS}}} \left(\frac{eK d^L_{\mathrm{E}}}{d^{\lceil L/2 \rceil}_{\mathrm{DS}}} \right)^{d^{\lceil L/2 \rceil}_{\mathrm{DS}}}.$$

By rearranging the term, we have

$$\left(\frac{2L+2}{L+1} \right)^{d^L_{\mathrm{E}}} \leq \left(\frac{2eK d^L_{\mathrm{E}}}{(L+1) d^{\lceil L/2 \rceil}_{\mathrm{DS}}} \right)^{d^{\lceil L/2 \rceil}_{\mathrm{DS}}}.$$

Let $x = d_{\mathrm{E}}^{L}/d_{\mathrm{DS}}^{\lceil L/2 \rceil}$. Rearranging the preceding inequality gives

$$2^x \le \frac{2eKx}{L+1} \le eKx,$$

where we used $L \ge 1$. Equivalently,

$$x \le \log(eK) + \log x.$$

If $x \le 4$, then $x \le 4 \log K \le 6 \log K$. Otherwise, $\log x \le x/2$, and hence

$$x \le 2 \log(eK) \le 6 \log K,$$

where the last inequality uses $K \ge 2$. Therefore

$$d_{\mathrm{E}}^{L} \le 6 d_{\mathrm{DS}}^{\lceil L/2 \rceil} \log K.$$

□

C.2 Complete calculation in the proof of Theorem 4.2

Proof of Theorem 4.2. Based on Lemma 4.7, we know that with probability at least $1-\delta$, the hypothesis h_S returned by Algorithm 1 has error rate at most ϵ. Thus the sample complexity $\mathbf{m}_{\mathcal{H}}^{B}(\epsilon, \delta)$ can be upper bounded by the sample size of Algorithm 1. Formally, we have

$$\begin{aligned}
\mathbf{m}_{\mathcal{H}}^{B}(\epsilon, \delta) &\le \sum_{t=1}^{\lfloor \log K \rfloor} (m_t - m_{t-1}) \\
&\le \sum_{t=1}^{\lfloor \log K \rfloor} \frac{4L_{t-1}}{1 - \frac{(t-1)\epsilon}{\log K}} \left(15 \frac{\left(6 d_{DS}^{\lceil L_t/2 \rceil} \log K + \log\left(\frac{4 \log K}{\delta}\right)\right) \log K}{\epsilon} + \log\left(\frac{2 \log K}{\delta}\right) \right) + \lfloor \log K \rfloor \\
&\le \frac{360 \max_{1 \le t \le \lfloor \log(K) \rfloor} \left(L_{t-1} d_{DS}^{\lceil L_t/2 \rceil} \right) (\log K)^3}{\epsilon(1-\epsilon)} \\
&\quad + \frac{120 K \log K \log\left(\frac{4 \log K}{\delta}\right)}{\epsilon(1-\epsilon)} + \frac{8K \log\left(\frac{4 \log K}{\delta}\right)}{1-\epsilon}.
\end{aligned}$$

The additional $\lfloor \log K \rfloor$ term comes from rounding each epoch length upward and is absorbed by the final asymptotic bound. When $\epsilon \in (0, 0.1]$, we have

$$\begin{aligned}
\mathbf{m}_{\mathcal{H}}^{B}(\epsilon, \delta) \le 400 &\frac{\max_{1 \le t \le \lfloor \log(K) \rfloor} \left(L_{t-1} d_{DS}^{\lceil L_t/2 \rceil} \right) (\log K)^3}{\epsilon} \\
&+ 150 \frac{K \log K \log\left(\frac{4 \log K}{\delta}\right)}{\epsilon}.
\end{aligned}$$

Since

$$L_{t-1} \le 2L_t \le 4 \left\lceil \frac{L_t}{2} \right\rceil,$$

we have

$$L_{t-1} d_{DS}^{\lceil L_t/2 \rceil} \le 4 \max_{1 \le L \le K-1} \left(L d_{DS}^{L} \right).$$

Therefore,

$$\begin{aligned}
\mathbf{m}_{\mathcal{H}}^{B}(\epsilon, \delta) \le 1600 &\frac{\max_{1 \le L \le K-1} \left(L d_{DS}^{L} \right) (\log K)^3}{\epsilon} \\
&+ 150 \frac{K \log K \log\left(\frac{4 \log K}{\delta}\right)}{\epsilon},
\end{aligned}$$

In addition, for $\epsilon \ge 0.1$, run the algorithm with accuracy parameter $\epsilon' = \min\{\epsilon, 0.1\}$; since $1/\epsilon' \le 10/\epsilon$, the same asymptotic bound holds for every $\epsilon \in (0, 1)$. This finishes the proof.

□

D Effective Size of the Label Space

The dependence on $|\mathcal{Y}|$ in the current statement is an artifact of using $|\mathcal{Y}|$ in the analysis. Moreover, our results extend to an infinite label space after replacing the cardinality of the full label space with the effective label space size. In particualr, the relevant quantity is the maximum number of labels realizable at any single instance:

$$K_{\text{eff}}(\mathcal{H}) = \sup_{x \in \mathcal{X}} \left|\{h(x) \mid h \in \mathcal{H}\}\right|.$$

To begin, when $K_{\text{eff}}(\mathcal{H}) < \infty$, the class can be transformed into a class over the finite label space

$$[K_{\text{eff}}(\mathcal{H})].$$

For each $x \in \mathcal{X}$, let

$$\mathcal{H}(x) := \{h(x) \mid h \in \mathcal{H}\},$$

choose a set $\mathcal{Y}_x \subseteq \mathcal{Y}$ of size $K_{\text{eff}}(\mathcal{H})$ containing $\mathcal{H}(x)$, and fix a bijection

$$\phi_x : \mathcal{Y}_x \longrightarrow [K_{\text{eff}}(\mathcal{H})].$$

Define the transformed class

$$\widetilde{\mathcal{H}} := \left\{\widetilde{h} \in [K_{\text{eff}}(\mathcal{H})]^{\mathcal{X}} \mid \exists\, h \in \mathcal{H} \text{ such that } \forall\, x \in \mathcal{X} \widetilde{h}(x) = \phi_x(h(x))\right\}.$$

Since each ϕ_x is injective, and every prediction in $[K_{\text{eff}}(\mathcal{H})]$ can be mapped back through ϕ_x^{-1}, this transformation preserves all coordinate-wise equality and inequality relations. Consequently, restriction of the class to any finite sequence of instances preserves neighbor relations, pseudo-boxes, the relevant combinatorial dimensions, and the associated one-inclusion hypergraphs. Having bandit feedback does not pose any obstacle. Therefore, ListCascade and its analysis would yield an upper bound with $K_{\text{eff}}(\mathcal{H})$ in place of $|\mathcal{Y}|$.

We next explain why an infinite effective label space precludes uniform bandit PAC learnability. Indeed, fix any sample size $m \in \mathbb{N}$, and choose a point $x \in \mathcal{X}$ at which more than $2m+1$ distinct labels are realizable by $\mathcal{H}$. Note that such a point exists by the definition of $K_{\text{eff}}(\mathcal{H})$. Fix $2m+2$ distinct realizable labels $y_1, \ldots, y_{2m+2}$ at this point. For each realizable label y at x, let $\mathcal{D}_y$ be the distribution concentrated on the single labeled example (x, y). We then choose one of the distributions

$$\{\mathcal{D}_{y_1}, \ldots, \mathcal{D}_{y_{2m+2}}\}$$

uniformly at random. In m bandit interactions, the learner can test at most m labels, and after receiving only negative feedback, it can correctly identify at most one additional untested label through its final output. Hence, its average probability of identifying the target label is at most

$$\frac{m+1}{2m+2} \leq \frac{1}{2}.$$

Therefore, for at least one realizable target, the learner outputs an incorrect label at x with probability at least $1/2$. Since the distribution is concentrated on x, the resulting classification error is one. Thus, for any $\epsilon < 1$ and $\delta < 1/2$, no finite sample size can provide a uniform PAC guarantee.

Now, observe that having a finite BDS dimension rules out this obstruction. In fact, we have:

$$K_{\text{eff}} - 1 \leq \text{BDS}(\mathcal{H}) \leq (K_{\text{eff}} - 1)\ \text{DS}(\mathcal{H}).$$

For the first inequality, r realizable labels at one point give a one-coordinate BDS witness with $r-1$ neighbors. For the second inequality, every coordinate of a BDS witness contributes at most $K_{\text{eff}}(\mathcal{H}) - 1$ neighbors, while the number of coordinates is at most $\text{DS}(\mathcal{H})$.

As a result, the suggested possibility of a class with $\text{BDS}(\mathcal{H}) = 1$ but infinite bandit PAC sample complexity cannot occur.

If $K_{\text{eff}}(\mathcal{H}) = 1$, all hypotheses agree pointwise and zero samples suffice. The following bound applies when $K_{\text{eff}}(\mathcal{H}) \geq 2$. Finally, after replacing $|\mathcal{Y}|$ by $K_{\text{eff}}(\mathcal{H})$, our upper bound becomes

$$O\left(\frac{\text{BDS}(\mathcal{H})\big(\log K_{\text{eff}}(\mathcal{H})\big)^3 + K_{\text{eff}}(\mathcal{H}) \log K_{\text{eff}}(\mathcal{H}) \log\left(\frac{2 \log K_{\text{eff}}(\mathcal{H})}{\delta}\right)}{\epsilon}\right).$$

Since

$$K_{\text{eff}}(\mathcal{H}) \leq \text{BDS}(\mathcal{H}) + 1,$$

the apparent dependence on the effective label-space size is absorbed by the BDS dimension up to logarithmic factors.

E Concentration Inequalities

This section collects standard concentration inequalities for sums of independent random variables, which are used throughout our analysis for the reader's convenience.

Lemma E.1 (Chernoff Bound)**.** *Let $(X_1, X_2, \ldots, X_n)$ be a sequence of n independent Bernoulli random variables for some $n \in \mathbb{N}$, and define $X := \sum_{i=1}^{n} X_i$. Moreover, let $\mu = \mathbb{E}[X]$. Then, the following inequalities hold.*

1. ***(Upper Tail)*** *For any $\delta > 0$, we have:*

$$\mathbb{P}\big[X \geq (1+\delta)\mu\big] \leq \exp\left(-\frac{\delta^2 \mu}{2+\delta}\right).$$

2. ***(Lower Tail)*** *For any $0 < \delta < 1$, we have:*

$$\mathbb{P}\big[X \leq (1-\delta)\mu\big] \leq \exp\left(-\frac{\delta^2 \mu}{2}\right).$$

Lemma E.2 (Bernstein's Inequality)**.** *Let $(X_1, X_2, \ldots, X_n)$ be a sequence of n independent real-valued random variables with zero mean (i.e., for every $i \in [n]$, we have: $\mathbb{E}[X_i] = 0$). Suppose that there exists a constant $M \in \mathbb{R}^+$ such that for every $i \in [n]$, we have: $|X_i| \leq M$ almost surely. Moreover, define $S_n := \sum_{i=1}^{n} X_i$, and $\sigma^2 := \sum_{i=1}^{n} \mathbb{E}[X_i^2]$. Then for any $t > 0$, the following holds.*

$$\mathbb{P}\big[S_n \geq t\big] \leq \exp\left(-\frac{t^2}{2\sigma^2 + \frac{2}{3}Mt}\right).$$

Furthermore, applying the same bound to both S_n and $-S_n$ yields the two-sided inequality below.

$$\mathbb{P}\big[|S_n| \geq t\big] \leq 2\exp\left(-\frac{t^2}{2\sigma^2 + \frac{2}{3}Mt}\right).$$

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: All claims made in the abstract and introduction of this paper are supported by detailed proofs.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The paper clearly states all the assumptions made. In addition, we point out future directions in the Conclusion section.

   Guidelines:

   - The answer [N/A] 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: This paper provides a full set of assumptions and a complete proof for every Theoretical result.

Guidelines:

- The answer [N/A] 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: [N/A]

   Justification: The paper does not contain experiments.

   Guidelines:

   - The answer [N/A] 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
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) 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).
     (d) 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: [N/A]

Justification: The contribution of this work is mainly theoretical, thus there are no experiments requiring data or code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.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://neurips.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) necessary to understand the results?

Answer: [N/A]

Justification: This paper does not include experiments.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: This paper does not include experiments.

Guidelines:

- The answer [N/A] 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: [N/A]

   Justification: This paper does not include experiments.

   Guidelines:

   - The answer [N/A] 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 authors have reviewed the NeurIPS Code of Ethics and confirmed that the research conducted in this work conforms to it. We have also made sure to preserve anonymity.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: The research presented in this paper is foundational work in the field of learning theory; we do not anticipate any immediate societal consequences.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper is theoretical and poses no such risks.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not introduce or release new assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.