Proceedings of Machine Learning Research vol 291:1–41, 2025 38th Annual Conference on Learning Theory

Private List Learnability vs. Online List Learnability

Steve Hanneke STEVE.HANNEKE@GMAIL.COM
Computer Science Department, Purdue University

Shay Moran SMORAN@TECHNION.AC.IL
Departments of Mathematics, Computer Science, & Data and Decision Sciences, Technion
Google Research

Hilla Schefler HILLAS@CAMPUS.TECHNION.AC.IL
Mathematics Department, Technion

Iska Tsubari ISKA@CAMPUS.TECHNION.AC.IL
Mathematics Department, Technion

Editors: Nika Haghtalab and Ankur Moitra

Abstract

This work explores the connection between differential privacy (DP) and online learning in the context of PAC list learning. In this setting, a k-list learner outputs a list of k potential predictions for an instance x and incurs a loss if the true label of x is not included in the list. A basic result in the multiclass PAC framework with a finite number of labels states that private learnability is equivalent to online learnability [Alon, Livni, Malliaris, and Moran (2019); Bun, Livni, and Moran (2020); Jung, Kim, and Tewari (2020)]. Perhaps surprisingly, we show that this equivalence does not hold in the context of list learning. Specifically, we prove that, unlike in the multiclass setting, a finite k-Littlestone dimension—a variant of the classical Littlestone dimension that characterizes online k-list learnability—is not a sufficient condition for DP k-list learnability. However, similar to the multiclass case, we prove that it remains a necessary condition.

To demonstrate where the equivalence breaks down, we provide an example showing that the class of monotone functions with $k+1$ labels over $\mathbb{N}$ is online k-list learnable, but not DP k-list learnable. This leads us to introduce a new combinatorial dimension, the *k-monotone dimension*, which serves as a generalization of the threshold dimension. Unlike the multiclass setting, where the Littlestone and threshold dimensions are finite together, for $k > 1$, the k-Littlestone and k-monotone dimensions do not exhibit this relationship. We prove that a finite k-monotone dimension is another necessary condition for DP k-list learnability, alongside finite k-Littlestone dimension. Whether the finiteness of both dimensions implies private k-list learnability remains an open question.

1. Introduction

Modern machine learning applications often involve handling sensitive data, making privacy preservation a critical concern. Differential privacy (DP) [Dwork, McSherry, Nissim, and Smith (2006)] offers a rigorous framework for safeguarding individuals' information by ensuring that small changes in the input data have a minimal impact on the algorithm's output. In recent years, significant research has addressed the fundamental question of determining which learning tasks can be performed under the constraints of differential privacy. A brief overview of relevant works appears in Section 1.2.

© 2025 S. Hanneke, S. Moran, H. Schefler & I. Tsubari.

Private PAC learning and online learning. An important connection has emerged between private learning and online learning. This connection can be understood through their shared reliance on the concept of stability. Differential privacy is, by definition, a form of stability, as it ensures robustness to small changes in the input data. On the other hand, stability plays a central role in online learning paradigms such as *Follow the Leader* [Kalai and Vempala (2005); Abernethy et al. (2008); Shalev-Shwartz and Singer (2007); Hazan (2016)]. This relationship has been studied within the framework of *Probably Approximately Correct* (PAC) learning [Valiant (1984)]. It reveals an equivalence: a concept class $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$, where $\mathcal{X}$ is a domain and $\mathcal{Y} = \{0, 1, \ldots, \ell - 1\}$ is the label space[1], is DP PAC learnable if and only if it is online learnable. This equivalence was first established for the binary case ($\ell = 2$) by Alon, Livni, Malliaris, and Moran (2019); Bun, Livni, and Moran (2020) and later extended to general ℓ by Jung, Kim, and Tewari (2020); Sivakumar, Bun, and Gaboardi (2021). This work focuses on the relationship between private learnability and online learnability within the context of *list learning*.

List learning. k-List learning is a generalization of supervised classification in which, instead of predicting a single label, the learner outputs a short list of k potential labels, where the goal is for the true label to appear in this list. For example, in recommendation systems, it is standard to provide users with a shortlist of items, such as movies, products, or articles, rather than a single suggestion, to improve the likelihood of presenting relevant options. Similarly, in scenarios with noisy or ambiguous data, such as medical diagnosis, presenting multiple possible diagnoses is often more practical than requiring a model to pinpoint a single, definitive outcome. Additionally, many real-world decision-making tasks involve downstream human or automated processes, where presenting a shortlist allows for more informed and flexible decision-making.

The central question motivating this work is:

Does the equivalence between private PAC learning and online learning extend to the setting of k-list learning for ℓ-labeled multiclass problems, where $k \geq 2$ and $\ell \geq 3$?

Remark 1 *The case where $k = 1$ and $\ell < \infty$ reduces to the standard multiclass setting. Additionally, the case $k = \ell = 2$ is trivial, as every class is both DP and online 2-list learnable via a deterministic constant learner that always outputs the list $\{0, 1\}$. Thus, the smallest nontrivial case where this question remains unresolved is $k = 2$ and $\ell = 3$.*

List prediction rules naturally arise in conformal learning. In this setting, a conformal learner predicts possible labels for a test point while also quantifying its confidence. For example, in multiclass classification, a conformal learner assigns scores to each possible class, reflecting the likelihood that the test point belongs to it. The final prediction list is then obtained by selecting the highest-scoring classes. For more details, see Vovk, Gammerman, and Shafer (2005) and the surveys by Shafer and Vovk (2008); Angelopoulos and Bates (2021).

From a theoretical perspective, there has been growing interest in understanding and characterizing the sample complexity of list learning tasks, including PAC k-list learning [Charikar and Pabbaraju (2023)], online k-list learning [Moran, Sharon, Tsubari, and Yosebashvili (2023b)], and

1. In this work, we focus on the multiclass setting with a finite number of labels, though our results hold more generally for an arbitrary number of labels.

k-list regression [Pabbaraju and Sarmasarkar (2024)]. For an overview of relevant works, see Section 1.2.

Our contribution. We provide a negative answer to the motivating question and demonstrate that only one direction of the equivalence between private PAC learnability and online learnability extends to the setting of list learning, making it, to the best of our knowledge, the first known setting where this equivalence fails. Specifically:

I *Private List Learnability (PLL)* $\implies$ *Online List Learnability (OLL).* We establish that one direction of the equivalence holds in the list learning setting: if a class $\mathcal{C}$ is DP PAC k-list learnable, then it is also online k-list learnable. This result leverages the k-Littlestone dimension, a variation of the classical Littlestone dimension introduced by Moran et al. (2023b) to characterize k-list online learnability.[2] Conceptually, we demonstrate that deep k-Littlestone trees[3] are an obstacle for private learning: if the k-Littlestone dimension of $\mathcal{C}$ is infinite (and thus $\mathcal{C}$ is not online k-list learnable), then no DP k-list learner exists for $\mathcal{C}$. In other words, finite k-Littlestone dimension is a necessary condition for a concept class to be privately k-list learnable.

II *Online List Learnability (OLL)* $\not\implies$ *Private List Learnability (PLL).* Let $k \geq 2$. We prove that the class of $(k+1)$-labeled monotone functions over $\mathbb{N}$ is k-list online learnable (with a mistake bound of 1), but it is not DP PAC learnable. More broadly, we introduce a combinatorial dimension, which we term the *k-monotone dimension* (Theorem 3), that serves as a generalization of the threshold dimension. We establish that a finite k-monotone dimension is a necessary condition for a concept class to be privately k-list learnable.

Note that, in the classical multiclass setting, for any class $\mathcal{C}$, the threshold dimension $\texttt{TD}(\mathcal{C})$ and the Littlestone dimension $\texttt{LD}(\mathcal{C})$ are simultaneously finite (that is, if one is finite, so is the other [Shelah (1982); Hodges (1997)]). Consequently, the finiteness of the threshold dimension fully captures the online learnability of a class. However, in the context of list learning, this relationship no longer holds: there exist classes with a finite k-Littlestone dimension but an infinite k-monotone dimension, and vice versa (Theorem 5). To summarize, we have established that the finiteness of both the k-Littlestone dimension and the k-monotone dimension is a necessary condition for private k-list learning (Corollary A). An open question remains whether the finiteness of these two dimensions is also a sufficient condition for private k-list learning.

Finally, another contribution we highlight is of a more technical nature and concerns the use of Ramsey-theoretic tools in our proofs. In establishing our two main results, we employ distinct variants of Ramsey's theorem. For the result in Item II (OLL $\not\Rightarrow$ PLL), we use the classical Ramsey theorem for hypergraphs, whereas for the result in Item I (PLL $\Rightarrow$ OLL), we extend a Ramsey theorem for binary trees introduced by Fioravanti, Hanneke, Moran, Schefler, and Tsubari (2024). Specifically, we require a version of this theorem for trees with arity $k+1$, where k is the list size.

Fioravanti et al. (2024) developed their Ramsey theorem for trees to overcome a limitation of the classical Ramsey theorem for hypergraphs in proving that private learning implies online learning for general classification problems. At first glance, one might expect the tree-based Ramsey theorem to be strictly stronger in this context. However, we show that these two variants are fundamentally

2. The k-Littlestone dimension characterizes k-list online learnability in the sense that a class $\mathcal{C}$ is online k-list learnable if and only if its k-Littlestone dimension $\texttt{LD}_k(\mathcal{C})$ is finite [Moran et al. (2023b)].
3. A k-Littlestone tree is a $(k+1)$-ary shattered mistake tree, as described in Appendix A.1.

incomparable: in both directions, there exist cases where one applies while the other does not. For a more detailed discussion, see Section 1.1.

Organization. In the remainder of this section, we present our main results and discuss additional related work. In Section 2, we outline the proof of the k-Littlestone dimension lower bound, and in Section 3, we outline the proof of the k-monotone dimension lower bound. Furthermore, in Section 4, we prove that the k-Littlestone and k-monotone dimensions are incomparable. The full proofs, along with detailed preliminaries, are provided in Appendices A to D.

1.1. Main Results

This section presents the main contributions of the paper. We use standard definitions and terminology from learning theory and differential privacy; see Appendix A for detailed definitions. In this work, we focus on the case where the label space $\mathcal{Y}$ is finite, though the results extend to infinite label spaces.

The k-Littlestone dimension is a combinatorial dimension introduced by Moran et al. (2023b) that generalizes the classical Littlestone dimension and characterizes optimal mistake and regret bounds in online k-list learning. Unlike the Littlestone dimension, which is defined using binary mistake trees, the k-Littlestone dimension is based on $(k+1)$-ary mistake trees, meaning each internal vertex has outdegree $k+1$, where k corresponds to the list size of the learner. For the formal proof, refer to Appendix A.1. The following theorem provides a lower bound on the sample complexity of privately k-list learning a concept class $\mathcal{C}$ in terms of its k-Littlestone dimension. Therefore, a finite k-Littlestone dimension is a necessary condition for DP PAC k-list learning.

Theorem 2 *Let $k \geq 1$. Let $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$ be a concept class with k-Littlestone dimension $\mathtt{LD}_k(\mathcal{C}) \geq d$, and let $\mathcal{A}$ be an $\left(\frac{k!}{10^4(k+1)^{k+2}}, \frac{k!}{10^4(k+1)^{k+2}}\right)$-accurate k-list learning algorithm for $\mathcal{C}$ with sample complexity m, satisfying $(\epsilon, \delta(m))$-differential privacy for $\epsilon = \log\left(\frac{400k^2+1}{400k^2}\right)$ and $\delta(m) \leq \frac{1}{200k^2m^2}$. Then, the following bound holds:*

$$m = \Omega(\log^\star d),$$

where the Ω notation conceals a universal numerical multiplicative constant.

The proof outline of Theorem 2 and its main ideas are presented in Section 2, while the full proof appears in Appendix B.

In the multiclass setting, the proof of Alon et al. (2019); Jung et al. (2020) showing that private learnability implies online learnability utilizes the tight connection between the Littlestone and threshold dimensions—one is finite if and only if the other is finite. In their proof, the authors first establish a lower bound on the sample complexity of privately learning threshold functions. The threshold dimension of a class $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$ is the largest number d such that there are d thresholds embedded in $\mathcal{C}$. That is, there exist points $x_1, \ldots, x_d \in \mathcal{X}$, functions $c_1, \ldots, c_d \in \mathcal{C}$, and labels $y_1 < y_2 \in \mathcal{Y}$ such that for all i, j, we have $c_i(x_j) = y_1$ if $i \leq j$, and $c_i(x_j) = y_2$ if $i > j$.

We now define the k-monotone dimension, a generalization of the threshold dimension.

Monotone functions. Let $\mathcal{X}$ be a linearly ordered domain and let $\mathcal{Y}$ be a linearly ordered label space. A function $c : \mathcal{X} \to \mathcal{Y}$ is a $\mathcal{Y}$-labeled *monotone function* over $\mathcal{X}$ if $x_1 < x_2$ implies $c(x_1) \leq c(x_2)$. Alternatively, c is monotone if there exist labels $i_0 < i_1 < \ldots < i_m \in \mathcal{Y}$ and points $x_1 < \ldots < x_m \in \mathcal{X}$, such that:

$$c(x) = \begin{cases} i_0 & \text{if } x < x_1 \\ i_j & \text{if } x \in [x_j, x_{j+1}) \text{ for } j = 1, \ldots, m-1 \\ i_m & \text{if } x \geq x_m \end{cases}.$$

Note that threshold functions are a special case of monotone functions.

Definition 3 (k-Monotone Dimension) *Let $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$ be a class. The* k-monotone dimension *of $\mathcal{C}$, denoted $\mathtt{MD}_k(\mathcal{C})$, is the largest number d such that the following holds: There exist*

(i) points $x_1, \ldots, x_d \in \mathcal{X}$,

(ii) a subset of labels $K \subset \mathcal{Y}$ of size $|K| = k+1$,

(iii) and a linear ordering of K,

such that the restriction

$$\mathcal{C}|_{\{x_1,\ldots,x_d\}} = \{c : \{x_1, \ldots, x_d\} \to \mathcal{Y} \mid c \in \mathcal{C}\}$$

contains all the K-labeled monotone functions over $\{x_1 < \ldots < x_d\}$. If such numbers d can be arbitrarily large, we say $\mathtt{MD}_k(\mathcal{C}) = \infty$.

Note that when $k = 1$, the 1-monotone dimension is exactly equal to the threshold dimension.

The following theorem provides a lower bound on the sample complexity of privately k-list learning a concept class $\mathcal{C}$ in terms of its k-monotone dimension, establishing that finite k-monotone dimension is necessary for DP PAC k-list learning.

Theorem 4 *Let $k \geq 1$. Let $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$ be a concept class with k-monotone dimension $\mathtt{MD}_k(\mathcal{C}) \geq d$, and let $\mathcal{A}$ be an $\left(\frac{1}{200k(k+1)}, \frac{1}{200k(k+1)}\right)$-accurate k-list learning algorithm for $\mathcal{C}$ with sample complexity m, satisfying $(\epsilon, \delta(m))$-differential privacy for $\epsilon = 0.1$ and $\delta(m) \leq \frac{1}{6(200km)^4 \log^2(200km)}$. Then, the following bound holds:*

$$m = \Omega(\log^\star d),$$

where the Ω notation conceals a universal numerical multiplicative constant.

The proof idea of Theorem 4 is outlined in Section 3, with the full proof in Appendix C.

In contrast to the multiclass setting, the k-Littlestone and k-monotone dimensions do not exhibit the same relationship. Specifically, the next theorem demonstrates that the gap between them can be infinite.

Theorem 5 (k-LD vs. k-MD) *Let $k > 1$. There exist classes $\mathcal{C}_L, \mathcal{C}_M$ such that*

(i) $\mathtt{LD}_k(\mathcal{C}_L) = 1$ and $\mathtt{MD}_k(\mathcal{C}_L) = \infty$,

(ii) $\texttt{LD}_k(\mathcal{C}_M) = \infty$ *and* $\texttt{MD}_k(\mathcal{C}_M) = 1$.

The proof of Theorem 5 appears in Section 4.

By combining Theorems 2 and 4, we derive a more comprehensive result:

Corollary A (k-LD and k-MD are Necessary for PLL) *Let $\mathcal{X}$ be a domain and $\mathcal{Y}$ a label space. If a concept class $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$ is DP PAC k-list learnable for $k \geq 1$, then*

(i) $\texttt{LD}_k(\mathcal{C}) < \infty$, *and*

(ii) $\texttt{MD}_k(\mathcal{C}) < \infty$.

Our lower bounds for private list learnability follow two different approaches introduced by Alon et al. (2019) and Fioravanti et al. (2024) for proving lower bounds on private learning. Alon et al. (2019) established that private learnability implies online learnability by proving a lower bound on privately learning thresholds, leveraging the classical Ramsey theorem for hypergraphs. However, since the tight connection between thresholds and the Littlestone dimension does not extend to more general settings—such as partial concept classes [Long (2001); Alon et al. (2021)] and multiclass learning with infinitely many labels—this approach cannot be applied directly. Later, Fioravanti et al. (2024) provided an alternative proof by reasoning directly about Littlestone trees, applying a Ramsey theorem for trees without relying on thresholds as an intermediate step. This approach resolved the open problem for partial concept classes and multiclass settings with infinitely many labels. While the proof of Fioravanti et al. (2024) may appear more general—since Ramsey theorem for trees was specifically developed to analyze Littlestone trees, which characterize online learnability, and has led to lower bounds in a broader range of settings—our results suggest that the two techniques are ultimately incomparable.

In the multiclass case, thresholds and Littlestone trees serve as essentially the same barrier for private learning. However, in the context of list learning, monotone functions and k-Littlestone trees form distinct and incomparable barriers. Both finite k-Littlestone and k-monotone dimensions are necessary for private k-list learning, with examples where one is finite while the other infinite. It remains an open question whether the finiteness of both k-Littlestone and k-monotone dimensions is sufficient for private k-list learning.

Open Question *Let $k > 1$ and let $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$ be a concept class with $\texttt{LD}_k(\mathcal{C}), \texttt{MD}_k(\mathcal{C}) < \infty$. Is $\mathcal{C}$ DP PAC k-list learnable?*

While no single combinatorial parameter is currently known to characterize private list learnability, the fact that both the k-Littlestone dimension and the k-monotone dimension are necessary—but neither is sufficient—suggests that a more nuanced, potentially multi-parameter or non-combinatorial characterization may be required. See Section 1.2 for further discussion.

Finally, the next theorem summarizes the relationship between private list learnability and online list learnability.

Corollary B (PLL $\overset{\rightarrow}{\not\leftarrow}$ OLL) *In the items below, $\mathcal{X}$ denotes an arbitrary domain and $\mathcal{Y}$ denotes a finite label space.*

(i) For every $k \geq 1$, if $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$ is DP PAC k-list learnable then it is online k-list learnable. Moreover, this implication applies to arbitrary (possibly infinite) label space $\mathcal{Y}$.

(ii) For $k = 1$, if $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$ is online k-list learnable then it is DP PAC k-list learnable.

(iii) For every $k > 1$, there exists a concept class $\mathcal{C}_k \subset \{0, \ldots, k\}^{\mathbb{N}}$ that is online k-list learnable but is not DP PAC k-list learnable.

Corollary B follows directly from Theorem 5 and Corollary A, with the latter being a consequence of Theorems 2 and 4. In particular, Item *(i)* in Corollary B follows directly by Theorem 2. On the other hand, Item *(iii)* follows as a corollary of Theorems 4 and 5. Specifically, the class $\mathcal{C}_L$ from Theorem 5 is online k-list learnable because it has finite k-Littlestone dimension. However, since its k-monotone dimension is unbounded Theorem 4 implies it is not DP PAC k-list learnable. The case $k = 1$ corresponds to the known result in the multiclass setting [Alon et al. (2019); Bun et al. (2020); Jung et al. (2020)].

1.2. Extended Related Work

Private learning. The study of the PAC learning model under differential privacy was initiated by Kasiviswanathan, Lee, Nissim, Raskhodnikova, and Smith (2011), who showed that every finite concept class $\mathcal{C}$ is privately learnable with a sample complexity of $O(\log |\mathcal{C}|)$. However, this bound is loose for many specific concept classes of interest and offers no guarantees for infinite classes. One of the most fundamental and well-studied classes is the class of linear classifiers (also known as threshold functions) in $\mathbb{R}^d$. While this class is well known to be PAC learnable, a seminal result by Bun, Nissim, Stemmer, and Vadhan (2015) demonstrates that it is impossible to properly DP learn this class, establishing a lower bound on the sample complexity for one-dimensional thresholds. Subsequently, in his thesis, Bun (2016) provided an alternative proof of this result using a Ramsey-theoretic argument. Later, Alon, Livni, Malliaris, and Moran (2019) extended this result in the context of binary classification beyond proper learning and showed that if a class $\mathcal{C}$ has Littlestone dimension d (and hence threshold dimension of at least $\log d$), then every (possibly improper) private PAC learner for $\mathcal{C}$ requires at least $\Omega(\log^\star d)$ samples, once again leveraging Ramsey's theorem. Jung, Kim, and Tewari (2020) and Sivakumar, Bun, and Gaboardi (2021) extended this result to the multiclass setting with a finite number of labels, while Fioravanti, Hanneke, Moran, Schefler, and Tsubari (2024) further generalized it to the multiclass setting with an arbitrary number of labels and to partial concept classes, by developing Ramsey-type theorems for binary trees. On the other hand, regarding the opposite direction, Bun, Livni, and Moran (2020) showed that every Littlestone class is privately learnable with a sample complexity of $2^{O(2^d)}$, where d denotes the Littlestone dimension. This bound was later improved to $\tilde{O}(d^6)$ by Ghazi, Golowich, Kumar, and Manurangsi (2021).

Another line of research examines private learning through the lens of stability, uncovering interesting connections to replicability and reproducibility, low information complexity, PAC-Bayes stability, and other variants of algorithmic stability [Impagliazzo, Lei, Pitassi, and Sorrell (2022); Bun, Gaboardi, Hopkins, Impagliazzo, Lei, Pitassi, Sivakumar, and Sorrell (2023); Livni and Moran

(2020); Moran, Schefler, and Shafer (2023a); Pradeep, Nachum, and Gastpar (2022); Malliaris and Moran (2022)].

Furthermore, extensive research has been dedicated to understanding which learning tasks can be performed under *pure* differential privacy. Beimel, Nissim, and Stemmer (2013, 2019) introduced the concept of *representation dimension*, a quantity that characterizes pure DP learnability. In a subsequent work, Feldman and Xiao (2015) discovered an interesting connection with communication complexity, associating every concept class $\mathcal{C}$ with a communication task whose complexity determines whether $\mathcal{C}$ is pure DP learnable. Additionally, Alon, Moran, Schefler, and Yehudayoff (2023) provided a unified characterization for both pure and approximate differential privacy, using cliques and fractional cliques of a graph corresponding to $\mathcal{C}$.

List learning. In learning theory, the framework of list learning was introduced by Brukhim, Carmon, Dinur, Moran, and Yehudayoff (2022) as a means to characterize multiclass PAC learnability. Since then, it gained traction as an area of study in its own right. Notably, Charikar and Pabbaraju (2023) provided a formal characterization of list PAC learnability, while Moran, Sharon, Tsubari, and Yosebashvili (2023b) independently characterized list learnability within the online model. Hanneke, Moran, and Waknine (2024) examined classical principles related to generalization, such as uniform convergence, Empirical Risk Minimization, and sample compression, within the context of list PAC learning. More recently, Pabbaraju and Sarmasarkar (2024) extended the scope of list learning to regression problems. Furthermore, list learning has played a significant role in the study of boosting, with notable contributions from Brukhim et al. (2023b,a); Bressan et al. (2024). It also has strong connections to the *Precision and Recall* learning framework, introduced by Cohen, Mansour, Moran, and Shao (2024).

Combinatorial parameters in learning theory. Learning-theoretic phenomena occasionally require combinations of distinct combinatorial parameters rather than single-dimensional characterizations. One notable example is proper learning. In both PAC and online settings, proper learnability depends simultaneously on the VC dimension and dual Helly number Bousquet et al. (2020); Kane et al. (2019); Hanneke et al. (2021). Interestingly, the dual Helly number may remain finite even when the VC dimension is unbounded. Another example that exhibits a related phenomenon is list replicability. While a class has finite list replicability number if and only if it has finite Littlestone dimension, no quantitative relationship exists between them—some classes exhibit arbitrarily large Littlestone dimension yet has list replicability number as low as 2. This disconnect motivates the conjecture that the list replicability number may instead be governed by VC dimension Chase et al. (2024, 2023). These examples demonstrate that, while uncommon, certain learning-theoretic phenomena may inherently require multiple combinatorial parameters. This raises the possibility that private list learnability, too, may not be captured by a single parameter, but rather by a more intricate combination of factors.

2. Private k-List-Learnability Implies Finite k-Littlestone Dimension

In this section, we outline the proof of Theorem 4 and establish a lower bound on the sample complexity of privately learning k-Littlestone classes. The full proof is presented in Appendix B. Our proof follows the approach of Fioravanti et al. (2024) for deriving a lower bound on the sample complexity of privately learning $(k = 1)$-Littlestone classes, with several necessary adaptations.

We begin with a general overview of the proof and, in Section 2.1, provide a detailed comparison between our proof and that of Fioravanti et al. (2024).

The first step of the proof is to show the following. Given a $(k+1)$-ary mistake tree T and any k-list algorithm $\mathcal{A}$ that takes T-realizable[4] samples as input, there exists a large subtree of T where the loss of $\mathcal{A}$ on every test point x depends only on comparisons within the tree. (We elaborate on this below.) The second step of the proof hinges on a reduction from the *interior point problem*, which was introduced by Bun, Nissim, Stemmer, and Vadhan (2015) in the context of properly learning thresholds (see Appendix A.2 for the formulation of the problem).

Step 1: Reduction to algorithms with comparison-based loss. The starting point of the proof is to show that, given a $(k+1)$-ary mistake tree T, for every algorithm $\mathcal{A}$ we can find a subtree of T on which the behavior of $\mathcal{A}$ is based on comparisons with respect to the natural partial order on T. A result of comparing two examples x' and x'' can be one of the following outcomes: either

1. x' is a i-th descendant[5] of x'', where $i = 0, \ldots, k$,
2. x'' is a i-th descendant of x', where $i = 0, \ldots, k$, or
3. x' and x'' are incomparable.

Notice that there are a total of $2(k+1)+1$ possible comparison outcomes. Now, what does it mean for an algorithm to behave in a comparison-based manner? Fioravanti et al. (2024) considered algorithms whose **predictions** are based solely on comparisons. Specifically, for every T-realizable input sample S and any test point x, the prediction $\mathcal{A}(S)(x)$ depends only on the comparisons of the test point x with points in S. This is a strong algorithmic guarantee, which is key to derive the impossibility result. However, it is also a natural notion, as many reasonable algorithms exhibit comparison-based behavior.

Attempting to extend the definition of comparison-based predictions to the setting of k-list algorithms would introduce unnecessary complications to the proof. Additionally, it would lead to a deterioration of the bounds, making them dependent on the arity of the tree. This outcome is counterintuitive because, as the size of the list available to the algorithm increases, one would naturally expect it to have greater predictive power. To circumvent this issue, we revisit the proof of Fioravanti et al. (2024) and identify the key property necessary for the second step of the proof (the reduction from the interior point problem). Specifically, it is sufficient for the algorithm to have comparison-based **loss**, rather than requiring its entire predictions to be comparison-based. Roughly speaking, a k-list algorithm $\mathcal{A}$ has *comparison-based loss* with respect to T if the following holds. For every T-realizable input sample S and any test point x on the branch realizing S, the probability that the list $\mathcal{A}(S)(x)$ does not contain the correct label of x depends only on the comparisons of the test point x with points in S. We remark that, in the case of binary Littlestone trees (i.e., $k = 1$), the definitions of comparison-based loss and comparison-based predictions coincide.

Finally, concluding this step required generalizing the Ramsey theorem for trees. While Fioravanti et al. (2024) proved this result for binary trees to show that any algorithm has comparison-based predictions on a deep subtree, we extended it to b-ary trees. This generalization allowed us to demonstrate that, for every k-list algorithm, there exists a deep subtree on which it exhibits comparison-based loss.

4. A T-realizable sample is a sample that is realizable by a branch of T
5. A vertex v is the i-th descendant of u in a b-ary tree T if v belongs to the subtree rooted at the child of u corresponding to its i-th outgoing edge.

Step 2: Reduction from the interior point problem. The second step of the proof consists of establishing a lower bound on the sample complexity of private k-list algorithms with comparison-based loss. We do so by showing a reduction from the interior point problem. Recall that a randomized algorithm solves the interior point problem on $[n]$ if for every input dataset $X \in [n]^m$, with high probability it returns a point that lies between $\min X$ and $\max X$. Bun et al. (2015) showed that solving the interior point problem in a private manner requires a dataset size of $m \geq \Omega(\log^\star n)$ (see Theorem 9). We use this result to derive a lower bound in our setting.

Let T be a $(k+1)$-mistake tree of depth n and let $\mathcal{A}$ be a private empirical k-list learner[6] for T with comparison-based loss. Let $d_1 < \ldots < d_m \in [n]$ be the input for the interior point problem. The reduction proceeds as follows.

1. Pick a branch B in T uniformly at random, and associate each point d_i with the point x_i on B at depth d_i. This defines an input sequence $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big)$, where the labels y_i's are determined by the branch B.
2. Run $\mathcal{A}$ on S and search for an interval of length l on B, where l is sufficiently large, such the accumulated loss over this interval is $\leq \frac{1}{2(k+1)} \cdot l$. Note that this loss guarantee outperforms guessing a random list of size k, which would result in an expected accumulated loss of $\frac{1}{k+1} \cdot l$.
3. Return the depth of the first point in the deepest such interval.

The idea that stands behind the reduction is that the predictions of the k-list learner have high correlation with the branch on which the input sample S lies on. Specifically, we show that with high probability, the output of the reduction is an interior point of $\{d_1, \ldots, d_m\}$. This is achieved by proving two key properties:

(a) With high probability, there exists such a correlated interval (as described in Item 2) beginning between x_1 and x_m.

(b) It is very unlikely that such an interval begins after x_m.

The second item is simple because $\mathcal{A}$ can only access information up to depth d_i, and the part of the (random) branch below it is therefore independent of $\mathcal{A}$'s output hypothesis. Hence, below this depth, $\mathcal{A}$ cannot significantly outperform random guessing.

The first item is more challenging, and its proof heavily relies on the fact that $\mathcal{A}$ is differentially private and has a comparison-based loss. The challenge in deriving the first item arises from the fact that an empirical learner is only guaranteed to be accurate on its training set, whereas we seek accuracy over a long continuous interval that includes many examples outside the training set. We address this by leveraging the comparison-based nature of the algorithm to argue that its loss remains the same on similar points lying between consecutive training examples, and by utilizing its differential privacy to shift training examples as needed.

We note that this part of the proof is also the densest one in Fioravanti et al. (2024)'s argument for the case of $k = 1$. Whether their analysis of this part extends to k-list learning for arbitrary k is unclear; at the very least, a naive extension would significantly complicate the calculations and lead to an unreasonably large case analysis. We circumvent this complication by identifying a way to

6. A k-list learner is empirical learner for T if it is an empirical learner with respect to input samples that are realizable by (a branch of) T. It is enough to consider private empirical PAC learners, since any private PAC learner can be transformed into a private empirical learner, while the sample complexity is increased only by a multiplicative constant factor. See Theorem 8.

simplify Fioravanti et al. (2024)'s argument. Not only does this simplification make the extension to general k more attainable, but it also simplifies the original proof by Fioravanti et al. (2024), both conceptually and technically. We discuss this step further in the next section.

2.1. Comparison to Fioravanti et al. (2024)

Comparison-based loss vs. comparison-based predictions. As elaborated above, one difference between our proof and that of Fioravanti et al. (2024) is that they considered algorithms whose entire predictions, rather than just their loss, are comparison-based. While Fioravanti et al. (2024)'s notion of comparison-based predictions is more intuitive than our notion of comparison-based loss, using the latter significantly simplifies the proof from a technical perspective and yields better bounds.

We now discuss a more substantial difference, which not only enabled the extension of Fioravanti et al. (2024)'s proof approach to list learning but also provided an insight that simplifies their original proof.

Analysis of the reduction from interior point problem. Recall that the main challenge in analyzing the reduction is to leverage the properties of the learner—specifically, privacy and comparison-based loss—to extend the guarantee on $\mathcal{A}$'s empirical loss from the sample to an entire interval (Item (a) above).

In their proof, Fioravanti et al. (2024) showed that, with high probability, one can find two consecutive points x_i and x_{i+1} on the sample S such that:

1. The empirical loss of $\mathcal{A}(S)$ on x_i and x_{i+1} is relatively small,
2. The labels of x_i and x_{i+1} are different, and
3. There exist two "matching neighbors" on the branch B. That is, there are points x' and x'' such that x' lies on the branch B between x_{i-1} and x_i, and x'' lies on B between x_{i+1} and x_{i+2}. Furthermore, the label of x' matches the label of x_i, and the label of x'' matches the label of x_{i+1}.

Then, they argued that the loss of $\mathcal{A}(S)$ on the interval between x_i and x_{i+1} is small as follows. Let x be a point on the interval between x_i and x_{i+1}. Since the tree is binary, the label of x matches the label of one of x_i or x_{i+1} (by Item 2). Suppose it matches the label of x_i. Now, define a new sample S' obtained by replacing x_i with its matching neighbor x'. Then, the following holds:

$$\mathcal{A}(S)(x) \overset{\text{DP}}{\approx} \mathcal{A}(S')(x) \overset{\text{CB}}{\approx} \mathcal{A}(S')(x_i) \overset{\text{DP}}{\approx} \mathcal{A}(S)(x_i),$$

where "$\overset{\text{DP}}{\approx}$" denotes approximate equality due to differential privacy, and "$\overset{\text{CB}}{\approx}$" denotes approximate equality due to comparison-based behavior. Therefore, the empirical loss of $\mathcal{A}$ on an internal point in the interval is controlled by the loss at the endpoints, which is small (by Item 1).

When extending this analysis to our setting of k-list learning, we encountered several challenges. The first challenge arises from the fact that if we only consider two points in the sample with different labels, we can only guarantee accuracy on points with those specific two labels, rather than across the entire interval. This is problematic because minimizing loss on just two labels can always be trivially achieved by including both labels in the prediction list.

A natural way to address this issue is to consider k consecutive intervals whose endpoints have $k+1$ distinct labels and to argue that the algorithm maintains small loss over the union of

these k intervals. However, reasoning over a union of k intervals introduces a fundamental obstacle: finding appropriate matching neighbors becomes effectively impossible. Specifically, shifting training-set points without altering their labels—an essential step in Fioravanti et al. (2024)'s approach—no longer seems feasible, making it unclear how to ensure the algorithm's loss remains small on every point in the union of the k intervals.

Can this obstacle be overcome? A closer inspection of Fioravanti et al. (2024)'s proof reveals that they took great care to ensure that the comparisons between a point x and the original sample S, as well as between a point x_i and the modified sample S', remained identical. This corresponds to the matching neighbors requirement (Item 3). Our key insight was that by leveraging the full strength of the Ramsey theorem for trees—specifically, a version that ensures all chains have colors determined solely by their order type—we gain significantly more flexibility in shifting training examples. This removes the stringent constraints that arise when attempting to extend Fioravanti et al. (2024)'s original analysis, allowing us to circumvent the fundamental obstacle discussed above. This insight also leads to a simplification of Fioravanti et al. (2024)'s proof by eliminating the matching neighbors constraint in Item 3.

3. Private k-List-Learnability Implies Finite k-Monotone Dimension

In this section, we outline the proof of Theorem 4 that establishes a lower bound on the sample complexity of privately learning monotone functions. The full proof appears in Appendix C. Our proof follows the approach of Alon et al. (2019) for deriving a lower bound on the sample complexity of privately learning thresholds, with several necessary adaptations. It consists of two main steps.

First, we show that for any k-list algorithm, we can identify a large subset $\mathcal{X}' \subseteq \mathcal{X}$ where the algorithm's predictions depend only on comparisons. In the second step, we use a packing argument to show that for private k-list algorithms that learn monotone functions, $\mathcal{X}'$ cannot be too large. Combining the bounds derived in each step yields the desired lower bound on the sample complexity.

To provide context, we briefly describe the proof of Alon et al. (2019). Their approach exploits the structure of one-dimensional thresholds and applies the classical Ramsey theorem to identify a *homogeneous set* of large size. A subset $\mathcal{X}'$ of an ordered domain $\mathcal{X}$ is considered homogeneous with respect to an algorithm $\mathcal{A}$ if whenever the input sample S and the test point x are from $\mathcal{X}'$, then the prediction of $\mathcal{A}$ for a test point x depends solely on the relative position of x within the sorted input sample S. In other words, the algorithm behaves in a comparison-based manner on inputs from $\mathcal{X}'$.

To find such a subset $\mathcal{X}'$, Alon et al. (2019) model a randomized learner trained on an input sample S as a deterministic function $\mathcal{A}(S) : \mathcal{X} \to \Delta(\{0,1\})$, where $\Delta(\{0,1\})$ denotes the space of probability distributions over $\{0,1\}$. This space is identified with the interval $[0,1]$ via the standard mapping $\mu \mapsto \mu(1)$. Under this formulation, $\mathcal{A}$ is comparison-based if the probability $\mathcal{A}(S)(x) \in [0,1]$ depends only on the labels of the sorted input sample S and the relative position of x within S.

By applying the classical Ramsey theorem to a carefully-defined coloring of subsets of size $m+1$ of $\mathcal{X}$, Alon et al. (2019) identify a large subset $\mathcal{X}' \subseteq \mathcal{X}$ on which $\mathcal{A}$ is *approximately* comparison-based in the following sense: there exists a comparison-based algorithm $\tilde{\mathcal{A}}$ such that, for every sample S and test point x from $\mathcal{X}'$,

$$|\mathcal{A}(S)(x) - \tilde{\mathcal{A}}(S)(x)| \leq O\left(\frac{1}{m}\right).$$

Having obtained $\mathcal{X}'$, Alon et al. (2019) leverage the comparison-based nature of $\mathcal{A}$ to attack privacy by identifying a family of indistinguishable distributions that satisfy a certain threshold property.

A natural approach to extending the definition of comparison-based algorithms to k-list learners is to model a k-list learner trained on a sample S as a mapping

$$\mathcal{A}(S) : \mathcal{X} \to \Delta\left(\binom{\mathcal{Y}}{k}\right),$$

where the predicted list for each point x depends only on its relative location with respect to the input sample. A k-list learner would then be considered approximately comparison-based if it is close to a comparison-based algorithm in total variation distance.

Unfortunately, this approach is infeasible. The dimension of the simplex $\Delta\left(\binom{\mathcal{Y}}{k}\right)$ is approximately $|\mathcal{Y}|^k$, which results in an enormous coloring space when applying the Ramsey argument. Moreover, the approach breaks down entirely when the label space $\mathcal{Y}$ is infinite, since the dimension of the simplex $\Delta\left(\binom{\mathcal{Y}}{k}\right)$ would also be infinite. To overcome this issue, we significantly relax the comparison-based property by requiring only that the marginal distributions

$$\Pr_{h\sim\mathcal{A}(S)}[y \in h(x)]$$

are comparison-based for every $y \in \mathcal{Y}$. This reduces the relevant simplex's dimension from $|\mathcal{Y}|^k$ to just $|\mathcal{Y}|$. We further refine this reduction by observing that it suffices to require the marginal distributions $\Pr_{h\sim\mathcal{A}(S)}[y \in h(x)]$ to be comparison-based only for the $k+1$ labels $y \in K$ that witness the monotone shattering (see Theorem 3). This further reduces the dimension to $k+1$, making the analysis much more tractable.

While these relaxations lose the intuitive algorithmic semantics of comparison-based algorithms, they significantly simplify the proof and make it feasible to implement in the context of k-list learnability.

4. k-Littlestone Dimension versus k-Monotone Dimension

In this section we prove Theorem 5.

A class $\mathcal{C}_L$ with $\mathtt{LD}_k(\mathcal{C}_L) = 1$ and $\mathtt{MD}_k(\mathcal{C}_L) = \infty$. Let $k > 1$. Let $\mathcal{M}_k(\mathcal{X})$ denote the class of all monotone functions with $k+1$ labels over $\mathcal{X}$, and set $\mathcal{C}_L = \mathcal{M}_k(\mathbb{N})$. The k-monotone dimension of $\mathcal{M}_k(\mathbb{N})$ is infinite. We claim that its k-Littlestone dimension is 1. It is straightforward to verify that $\mathcal{M}_k(\mathbb{N})$ shatters a k-Littlestone tree of depth 1. Therefore, it suffices to show that there exists an online k-list learner for $\mathcal{M}_k(\mathbb{N})$ that makes at most 1 mistake on any realizable input sequence. The learner's strategy is as follows. Initially, for every test point $x \in \mathbb{N}$, the learner predicts the list

$$\{0, \ldots, \lfloor k/2 \rfloor - 1, \lfloor k/2 \rfloor + 1, \ldots, k\},$$

until the first timestamp t where it makes a mistake on the point x_t. From that point onward, the learner modifies its strategy as follows:

- For test points $x \leq x_t$, it predicts the list $\{0, \ldots, k-1\}$.
- For test points $x > x_t$, it predicts the list $\{1, \ldots, k\}$.

This strategy ensures that the learner makes at most 1 mistake on any realizable input sequence, and therefore, $\texttt{LD}_k(\mathcal{M}_k(\mathbb{N})) = 1$, as desired.

To conclude, $\mathcal{M}_k(\mathbb{N})$ is online k-list learnable. However, by Theorem 4, whose proof is outlined in Section 3, it is not DP PAC k-list learnable. This completes the third item of Corollary B. To complete the proof of Theorem 5, it remains to show that there exist a class with finite k-monotone dimension, and infinite k-Littlestone dimension.

A class $\mathcal{C}_M$ with $\texttt{LD}_k(\mathcal{C}_M) = \infty$ and $\texttt{MD}_k(\mathcal{C}_M) = 1$. Consider $\mathcal{X}$ as the set of vertices of an infinite complete $(k+1)$-ary decision tree, where each vertex is labeled with a unique point $x \in \mathcal{X}$, and every vertex has $k+1$ outgoing edges labeled with $0, \ldots, k$. Define a concept class $\mathcal{C} \subset \{0, \ldots, k\}^{\mathcal{X}}$, consisting of all concepts that realize exactly one infinite branch of the tree, and label every point $x \in \mathcal{X}$ outside the branch with 0.

It is straightforward to verify that the k-Littlestone dimension of $\mathcal{C}$ is infinite. We claim, however, that the k-monotone dimension of $\mathcal{C}$ is 1. In fact, we prove an even stronger statement: $\mathcal{C}$ does not contain all constant functions over two points. Assume for contradiction that there exist two points $x_1, x_2 \in \mathcal{X}$ such that $\mathcal{C}_n$ realizes all $k+1$ constant functions over x_1 and x_2. Observe that one of these points must be a descendant of the other in the decision tree—otherwise, $\mathcal{C}$ cannot realize, for example, the function $(1,1)$ (or any other constant function that is not $(0,0)$).

Assume without loss of generality that x_2 is an i-descendant of x_1, meaning x_2 belongs to the i-th subtree emanating from x_1. In this case, $\mathcal{C}$ cannot realize, for example, the function $(i+1, i+1)$. This contradiction establishes that the k-monotone dimension of $\mathcal{C}$ is indeed 1.

Acknowledgments

SM is a Robert J. Shillman Fellow; he acknowledges support by ISF grant 1225/20, by BSF grant 2018385, by an Azrieli Faculty Fellowship, by Israel PBC-VATAT, by the Technion Center for Machine Learning and Intelligent Systems (MLIS), and by the European Union (ERC, GENERALIZATION, 101039692).

HS and IT acknowledge support 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

Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *COLT*, pages 263–274. Omnipress, 2008.

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite littlestone dimension. In *Proc. 51st Symp. Theory of Computing (STOC)*, pages 852–860, 2019.

Noga Alon, Steve Hanneke, Ron Holzman, and Shay Moran. A theory of PAC learnability of partial concept classes. In *Proc. 62nd Symp. Foundations of Computer Science (FOCS)*, pages 658–671, 2021.

Noga Alon, Mark Bun, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private and online learnability are equivalent. *J. ACM*, 69(4):28:1–28:34, 2022.

Noga Alon, Shay Moran, Hilla Schefler, and Amir Yehudayoff. A unified characterization of private learnability via graph theory, 2023.

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *CoRR*, abs/2107.07511, 2021.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *ITCS*. ACM, 2013.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of pure private learners. *Journal of Machine Learning Research*, 20(146):1–33, 2019. URL http://jmlr.org/papers/v20/18-269.html.

Olivier Bousquet, Steve Hanneke, Shay Moran, and Nikita Zhivotovskiy. Proper learning, helly number, and an optimal SVM bound. In *COLT*, volume 125 of *Proceedings of Machine Learning Research*, pages 582–609. PMLR, 2020.

Marco Bressan, Nataly Brukhim, Nicolò Cesa-Bianchi, Emmanuel Esposito, Yishay Mansour, Shay Moran, and Maximilian Thiessen. Of dice and games: A theory of generalized boosting, 2024. URL https://arxiv.org/abs/2412.08012.

Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In *Proc. 63rd Symp. Foundations of Computer Science (FOCS)*, pages 943–955. IEEE, 2022.

Nataly Brukhim, Amit Daniely, Yishay Mansour, and Shay Moran. Multiclass boosting: Simple and intuitive weak learning criteria. In *NeurIPS*, 2023a.

Nataly Brukhim, Steve Hanneke, and Shay Moran. Improper multiclass boosting. In *COLT*, volume 195 of *Proceedings of Machine Learning Research*, pages 5433–5452. PMLR, 2023b.

Mark Bun. *New Separations in the Complexity of Differential Privacy.* PhD thesis, Harvard University, Graduate School of Arts & Sciences, 2016.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *Proc. 56th Symp. Foundations of Computer Science (FOCS)*, pages 634–649, 2015.

Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. In *FOCS*, pages 389–402. IEEE, 2020.

Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, and Jessica Sorrell. Stability is stable: Connections between replicability, privacy, and adaptive generalization. In *STOC*, pages 520–527. ACM, 2023.

Moses Charikar and Chirag Pabbaraju. A characterization of list learnability. In *STOC*, pages 1713–1726. ACM, 2023.

Zachary Chase, Shay Moran, and Amir Yehudayoff. Stability and replicability in learning. In *FOCS*, pages 2430–2439. IEEE, 2023.

Zachary Chase, Bogdan Chornomaz, Shay Moran, and Amir Yehudayoff. Local borsuk-ulam, stability, and replicability. In *STOC*, pages 1769–1780. ACM, 2024.

Lee Cohen, Yishay Mansour, Shay Moran, and Han Shao. Probably approximately precision and recall learning. *CoRR*, abs/2411.13029, 2024.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. 3rd Conf. Theory of Cryptography (TCC)*, volume 3876, pages 265–284, 2006.

P. Erdös and R. Rado. Combinatorial theorems on classifications of subsets of a given set. *Proceedings of the London Mathematical Society*, 3(2):417–439, 1952.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. *SIAM Journal on Computing*, 44(6):1740–1764, 2015.

Simone Fioravanti, Steve Hanneke, Shay Moran, Hilla Schefler, and Iska Tsubari. Ramsey theorems for trees and a general 'private learning implies online learning' theorem. In *FOCS*, pages 1983–2009. IEEE, 2024.

Badih Ghazi, Noah Golowich, Ravi Kumar, and Pasin Manurangsi. Sample-efficient proper PAC learning with approximate differential privacy. In *STOC*, pages 183–196. ACM, 2021.

Steve Hanneke, Roi Livni, and Shay Moran. Online learning with simple predictors and a combinatorial characterization of minimax in 0/1 games. In *COLT*, volume 134 of *Proceedings of Machine Learning Research*, pages 2289–2314. PMLR, 2021.

Steve Hanneke, Shay Moran, and Tom Waknine. List sample compression and uniform convergence. In *COLT*, volume 247 of *Proceedings of Machine Learning Research*, pages 2360–2388. PMLR, 2024.

Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3-4):157–325, 2016.

Wilfrid Hodges. *A shorter model theory*. Cambridge University Press, 1997.

Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. In *STOC*, pages 818–831. ACM, 2022.

Young Hun Jung, Baekjin Kim, and Ambuj Tewari. On the equivalence between online and private learnability beyond binary classification. In *Proc. 33rd Conf. Adv. Neural Information Processing Systems (NeurIPS)*, 2020.

Adam Tauman Kalai and Santosh S. Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005.

Daniel Kane, Roi Livni, Shay Moran, and Amir Yehudayoff. On communication complexity of classification problems. In *COLT*, volume 99 of *Proceedings of Machine Learning Research*, pages 1903–1943. PMLR, 2019.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011.

Roi Livni and Shay Moran. A limitation of the pac-bayes framework. In *NeurIPS*, 2020.

Philip Long. On agnostic learning with $\{0, *, 1\}$-valued and real-valued hypotheses. In *Proc. 14th Conf. Learning Theory (COLT)*, 2001.

Maryanthe Malliaris and Shay Moran. The unstable formula theorem revisited. *CoRR*, abs/2212.05050, 2022.

Shay Moran, Hilla Schefler, and Jonathan Shafer. The bayesian stability zoo. In *NeurIPS*, 2023a.

Shay Moran, Ohad Sharon, Iska Tsubari, and Sivan Yosebashvili. List online classification. In *COLT*, volume 195 of *Proceedings of Machine Learning Research*, pages 1885–1913. PMLR, 2023b.

Chirag Pabbaraju and Sahasrajit Sarmasarkar. A characterization of list regression. *CoRR*, abs/2409.19218, 2024.

Aditya Pradeep, Ido Nachum, and Michael Gastpar. Finite littlestone dimension implies finite information complexity. In *ISIT*, pages 3055–3060. IEEE, 2022.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(12):371–421, 2008. URL http://jmlr.org/papers/v9/shafer08a.html.

Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Mach. Learn.*, 69(2-3):115–142, 2007.

S. Shelah. Classification theory and the number of nonisomorphic models. *J. Symbolic Logic*, 47 (3):694–696, 1982.

Satchit Sivakumar, Mark Bun, and Marco Gaboardi. Multiclass versus binary differentially private PAC learning. In *Proc. 34th Conf. Adv. Neural Information Processing Systems (NeurIPS)*, pages 22943–22954, 2021.

Salil P. Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer International Publishing, 2017.

Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer US, 2005. ISBN 9780387001524. URL https://books.google.co.il/books?id=pEXc4C1ymakC.

Appendix A. Preliminaries

A.1. List Learning

Multi-labeled hypotheses and list learners. Let $\mathcal{X}$ be a domain, let $\mathcal{Y}$ be a label space, and let $k \in \mathbb{N}$. We denote by $\binom{\mathcal{Y}}{k}$ the family of subsets of $\mathcal{Y}$ of size k. A *k-multi-labeled hypothesis* is a function $h : \mathcal{X} \to \binom{\mathcal{Y}}{k}$. Define the $0-1$ loss function of a k-multi-labeled hypothesis $h \in \binom{\mathcal{Y}}{k}^{\mathcal{X}}$, on a labeled example $(x, y) \in \mathcal{X} \times \mathcal{Y}$, as

$$l\big(h; (x, y)\big) = \mathbb{1}[y \notin h(x)].$$

The *empirical loss* of h with respect to a dataset $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big) \in (\mathcal{X} \times \mathcal{Y})^m$ is

$$\mathrm{L}_S(h) = \frac{1}{m} \sum_{i=1}^{m} l\big(h; (x_i, y_i)\big).$$

A dataset S is *realizable* by h if $\mathrm{L}_S(h) = 0$.

A k-list learner is a (possibly randomized) algorithm $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^\star \to \binom{\mathcal{Y}}{k}^{\mathcal{X}}$. We model a randomized a k-list learner $\mathcal{A}$ as a deterministic map $(\mathcal{X} \times \mathcal{Y})^\star \times \mathcal{X} \to [0,1]^{\mathcal{Y}}$. Specifically, given an input sample S and a test point $x \in \mathcal{X}$, we define

$$\mathcal{A}_{S,x}(y) = \Pr_{h \sim \mathcal{A}(S)} [y \notin h(x)] \text{ , where } y \in \mathcal{Y}.$$

Notice that when the label space $\mathcal{Y}$ is finite, $\sum_{y \in \mathcal{Y}} (1 - \mathcal{A}_{S,x}(y)) = k$. Equivalently, $\sum_{y \in \mathcal{Y}} \mathcal{A}_{S,x}(y) = |\mathcal{Y}| - k$.

List PAC learning. Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the *population loss* of a k-multi-labeled hypothesis h with respect to $\mathcal{D}$ is

$$\mathrm{L}_{\mathcal{D}}(h) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \big[l\big(h; (x, y)\big) \big].$$

We say that an algorithm $\mathcal{A}$ is an *(α, β)-accurate k-list learner* for $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$, with sample complexity m, if for every realizable distribution $\mathcal{D}$,

$$\Pr_{S \sim \mathcal{D}^m} [\mathrm{L}_{\mathcal{D}}(\mathcal{A}(S)) \geq \alpha] \leq \beta.$$

Here, α is called the *error* and β is called the *confidence parameter*. A class $\mathcal{C}$ is *PAC k-list learnable* if there exist vanishing $\alpha(m), \beta(m) \to 0$ and an algorithm $\mathcal{A}$ such that for all m, $\mathcal{A}$ is a $(\alpha(m), \beta(m))$-accurate k-list learner for $\mathcal{H}$ with sample complexity m.

List online learning. The framework of k-list online learning can be formulated as the following online game. At time t, an adversary presents an example $x_t \in \mathcal{X}$. The learner outputs a list of predictions of size k, $p_t \sim \mathcal{A}(S_{<t}, x_t)$, where $S_{<t} = \big((x_1, y_1), \ldots, (x_{t-1}, y_{t-1})\big)$. Then, the adversary reveals the true label $y_t \in \mathcal{Y}$, and the learner suffers loss $l(p_t; y_t) = \mathbb{1}[y_t \notin p_t]$. The goal of the learner is to minimize the expected number of mistakes

$$M(T) = \mathbb{E}_{p_t \sim \mathcal{A}(S_{<t}, x_t)} \left[\sum_{t=1}^{T} l(p_t; y_t) \right].$$

A concept class $\mathcal{C} \subset \mathcal{Y}^{\mathcal{X}}$ is k-list online learnable if there exists a k-list online learner $\mathcal{A}$ and a number M, such that for every realizable input sequence of length T, $M(T) \leq M$. A class $\mathcal{C}$ is k-list online learnable if and only if its k-Littlestone dimension is finite [Moran et al. (2023b)].

k-Littlestone dimension. The k-Littlestone dimension is a combinatorial parameter introduced by Moran et al. (2023b) that captures mistake and regret bounds in the setting of list online learning. The definition of the k-Littlestone dimension uses the notion of *mistake trees*. A mistake tree is a b-ary[7] decision tree whose vertices are labeled with instances from $\mathcal{X}$ and edges are labeled with labels from $\mathcal{Y}$ such that each internal vertex has b different labels on its b outgoing edges. A root-to-leaf path in a mistake tree corresponds to a sequence of labeled examples $\big((x_1, y_1), \ldots, (x_d, y_d)\big)$. The point x_i is the label of the i'th internal vertex in the path, and y_i is the label of its outgoing edge to the next vertex in the path. We say that a class $\mathcal{C}$ *shatters* a mistake tree if every root-to-leaf path is realizable by $\mathcal{C}$. The *k-Littlestone dimension* of $\mathcal{C}$, denoted $\texttt{LD}_k(\mathcal{C})$, is the largest number d such that there exists a complete $(k+1)$-ary mistake tree of depth d shattered by $\mathcal{C}$. If $\mathcal{C}$ shatters arbitrarily deep $(k+1)$-ary mistake trees then we write $\texttt{LD}_k(\mathcal{C}) = \infty$. Note that the Littlestione dimension is equal to the 1-Littlestone dimension $\texttt{LD}(\mathcal{H}) = \texttt{LD}_1(\mathcal{H})$.

A.2. Differential Privacy

We use standard definitions and notation from the differential privacy literature. For further background, see, e.g., the surveys Dwork and Roth (2014); Vadhan (2017). For two numbers a and b, denote $a \overset{\epsilon,\delta}{\approx} b$ if $a \leq e^{\epsilon} \cdot b + \delta$, and $b \leq e^{\epsilon} \cdot a + \delta$. Two probability distributions p, q are (ϵ, δ)-*indistinguishable* if for every event E, $p(E) \overset{\epsilon,\delta}{\approx} q(E)$.

Definition 6 *Let $\epsilon, \delta \geq 0$. A randomized learning rule $\mathcal{A}$ is (ϵ, δ)-differentially private if for every pair of training samples $S, S' \in (\mathcal{X} \times \mathcal{Y})^m$ that differ on a single example, the distributions $\mathcal{A}(S)$ and $\mathcal{A}(S')$ are (ϵ, δ)-indistinguishable.*

A basic property of differential privacy is that privacy is preserved under post-processing; it enables arbitrary data-independent transformations to differentially private outputs without affecting their privacy guarantees.

Proposition 7 (Post-Processing) *Let $\mathcal{A} : \mathcal{Z}^m \to \mathcal{R}$ be any (ϵ, δ)-differentially private algorithm, and let $f : \mathcal{R} \to \mathcal{R}'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{A} : \mathcal{Z}^m \to \mathcal{R}'$ is (ϵ, δ)-differentially private.*

Empirical learners. Let $\mathcal{C}$ be a class. An algorithm $\mathcal{A}$ is (α, β)-accurate empirical k-list learner with sample complexity m if for every realizable sample S of size m,

$$\Pr_{h \sim \mathcal{A}(S)}[\mathrm{L}_S(h) \geq \alpha] \leq \beta.$$

Bun, Nissim, Stemmer, and Vadhan (2015) proved that any private PAC learner can be transformed into a private empirical learner, while the sample complexity is increased only by a multiplicative constant factor.

Lemma 8 (Lemma 5.9 in Bun et al. (2015)) *Suppose $\mathcal{A}$ is an (ϵ, δ)-differentially private (α, β)-accurate PAC k-list learner for an hypothesis class $\mathcal{C}$ with sample complexity m. Then there is an (ϵ, δ)-differentially private (α, β)-accurate empirical k-list learner for $\mathcal{H}$ with sample complexity $n = 9m$.*

7. A b-ary tree is a tree in which every internal vertex has exactly b children.

We remark that Theorem 8 was proved in Bun et al. (2015) within the context of traditional PAC learning. However, the proof also applies to more general settings, including the setting of list learning.

Interior point problem. An algorithm $\mathcal{A} : [n]^m \to [n]$ solves the interior point problem on $[n]$ with probability $1 - \beta$ if for every input sequence $x_1 \dots x_m \in [n]$,

$$\Pr[\min x_i \leq \mathcal{A}(x_1, \dots x_m) \leq \max x_i] \geq 1 - \beta$$

where the probability is taken over the randomness of $\mathcal{A}$; the number of data points m is called the sample complexity of $\mathcal{A}$.

Theorem 9 (Theorem 3.2 in Bun et al. (2015)) *Let $0 < \epsilon < 1/4$ be a fix number and let $\delta(m) \leq 1/50m^2$. Then for every positive integer m, solving the interior point problem on $[n]$ with probability at least $3/4$ and with $(\epsilon, \delta(m))$-differential privacy requires sample complexity $m \geq \Omega(\log^\star n)$.*

A.3. General Definitions and Notations

Iterated logarithm and tower function. The tower function $\mathsf{twr}_{(t)}(x)$ and the iterated logarithm $\log_{(t)}(x)$ are defined by the recursions

$$\mathsf{twr}_{(i)}(x) = \begin{cases} x & i = 1 \\ 2^{\mathsf{twr}_{(i-1)}(x)} & i > 1 \end{cases}, \qquad \log_{(i)}(x) = \begin{cases} x & i = 0 \\ \log(\log_{(i-1)}(x)) & i > 0 \end{cases}.$$

Note that for all t, $\left(\mathsf{twr}_{(t)}(\cdot)\right)^{-1} = \log_{(t-1)}(\cdot)$. Finally,

$$\log^\star(x) = \min\{t \mid \log_{(t)}(x) \leq 1\}.$$

Appendix B. Private k-List-Learnability Implies Finite k-Littlestone Dimension

In this section we prove Theorem 2.

We start by introducing some notations and definitions which will be used in the proof. Let T be a b-ary tree. We assume an order on the outgoing edges of each internal vertex v, labeling them with $\{0, 1, \dots, b-1\}$. We refer to the vertex connected to v by the edge labeled i as the i-th child of v. An *m-chain* is a subset of vertices of size m that form a chain with respect to the natural partial order induced by T, that is $v < u$ if u is a descendant of v. The *type* of a chain $C = \{v_1 < \dots < v_m\}$ is a tuple $\boldsymbol{t} = \boldsymbol{t}(C) \in \{0, \dots, b-1\}^{m-1}$ where $\boldsymbol{t}_i = j$ if and only if v_{i+1} is a j's descendant of v_i. In other words, the type encodes the immediate turns made on the path from v_1 to v_m. See e.g. Figure 1.

Assume now that T is a Littlestone tree, with internal vertices from a domain $\mathcal{X}$, and edges from a label space $\mathcal{Y}$. To ease the notations, from this point we will assume that the b distinct labels on the outgoing edges of each internal vertex are $\{0, 1, \dots, b-1\}$ where by writing "i", we mean the label on the i'th outgoing edge. [8]

8. Note that in the case where $|\mathcal{Y}| > k + 1$, the probabilities of the learner labeling x as the labels on the $k + 1$ edges, does not necessarily sum to k. However, every k-list learner can be converted to k-list learner for which this sum is k (while maintaining utility and privacy), by a simple post-processing step: if the learner outputs a hypothesis that predicts a label different from the labels on the edges, replace it with one of them.

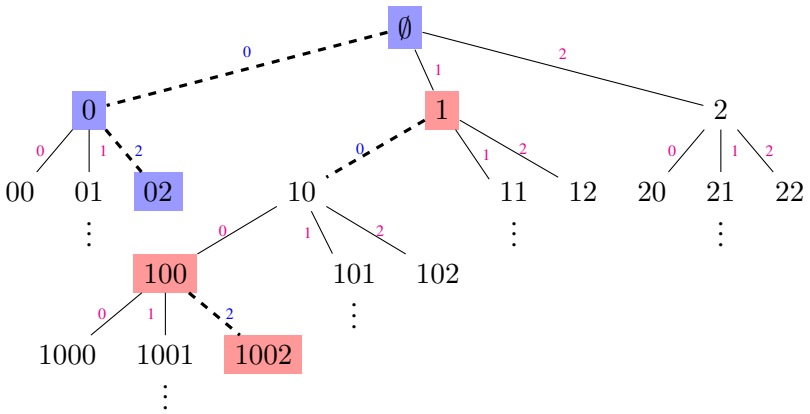

Figure 1: The 3-chain $C = \{\emptyset, 0, 02\}$ is of type $\boldsymbol{t}(C) = (0, 2)$. The 3-chain $\tilde{C} = \{1, 100, 1002\}$ has the same type as C.

We identify $(m+1)$-chains in T with T-realizable samples of size m as follows. An $(m+1)$-chain $C = \{x_1 < x_2 < \ldots < x_{m+1}\}$ in T corresponds to a sample $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big) \in (\mathcal{X} \times \mathcal{Y})^m$, such that y_i satisfies that x_{i+1} belongs to the y_i's subtree emanating from x_i. Namely, $(y_1, \ldots, y_m)$ is the type of C, $\boldsymbol{t}(C)$. An input sequence $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big)$ is T-realizable if S is realizable by a branch of T.

Let $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big)$ be a sample that is realizable by T, and assume for convenience that S is ordered, i.e. $x_1 < x_2 < \ldots < x_m$. An instance x is *compatible* with S if there exists a branch in T that realizes S and contains x. In such a case we denote by

$$S^{+x} := S \cup (x, y_x),$$

where $y_x \in \mathcal{Y}$ is a label such that $S \cup (x, y_x)$ is T-realizable. Note that if x appears earlier on the branch than x_m, then there is a unique such y_x. In the complementing case, when x appears after x_m, there are b different choices for y such that $S \cup (x, y)$ is T-realizable. In that case we arbitrarily pick $S^{+x} = S \cup (x, y_x)$ for the smallest such y, and therefore S^{+x} is well defined. The *location* of x in S is

$$\texttt{loc}_S(x) := \max\{i \mid x_i < x\}.$$

If $x < x_1$ then define $\texttt{loc}_S(x) := 0$.

Recall, given a compatible test point $x \in \mathcal{X}$

$$\mathcal{A}_{S,x}(y_x) = \Pr_{h \sim \mathcal{A}(S)}[y_x \notin h],$$

which is the loss of $\mathcal{A}(S)$ on the point (x, y_x).

The reduction to algorithms with comparison-based loss is achieved using a Ramsey theorem for trees. This theorem guarantees that for any given algorithm and a sufficiently deep Littlestone tree, there exists a deep enough subtree in which the algorithm has a comparison-based loss with respect to this subtree.

Definition 10 (Subtree) *Let T be a b-ary tree. Define a* subtree *of T by induction on its depth d. All vertices of T are subtrees of T of depth $d = 0$. For $d \geq 1$ a subtree of depth d is obtained from an internal vertex v of T and b subtrees of depth $d-1$ each rooted at a different child of v.*

Theorem 11 (Ramsey theorem on trees. Theorem C in Fioravanti et al. (2024)) *For all d, b, c, m there exists*

$$n \leq \mathsf{twr}_{(m)}(5 \cdot b^{m-2} d c^{b^{m-1}} \log c)$$

such that, for every coloring of m-chains in the complete b-ary tree of depth n with c colors, there exists a b^{m-1}-chromatic complete subtree of depth d (i.e. its m-chains are colored with at most b^{m-1} colors). Furthermore, the obtained subtree is type-monochromatic*, in the sense that if two m-chains are of the same type then they are colored with the same color.*

We note that Fioravanti et al. (2024) stated Theorem 11 only for binary trees ($b = 2$), but the same proof applies to general b-ary trees, with minor necessary modifications. For completeness, we provide the proof for general arity in Appendix D.

Definition 12 (Approximately Comparison-Based Loss) *Let T be a $(k+1)$-ary mistake tree. A (randomized) k-list algorithm $\mathcal{A}$, defined over input samples of size m, has γ-comparison-based loss on T if the following holds. There exist numbers $p_{\boldsymbol{t},i} \in [0,1]$ for $\boldsymbol{t} \in \{0,1\}^{m+1}$ and $i \in \{0,\ldots,m\}$ such that for every input sample S of size m realizable by T, and for every x compatible with S,*

$$|\mathcal{A}_{S,x}(y_x) - p_{\boldsymbol{t},i}| \leq \gamma,$$

where $\boldsymbol{t} = \boldsymbol{t}(S^{+x})$ is the type of the sample S^{+x}, y_x is the label that satisfies $S^{+x} = S \cup (x, y_x)$, and $i = \mathtt{loc}_S(x)$ is the location of x in S.

Referring to the label y_x as the correct label for x, this property of $\mathcal{A}$ essentially states that the probability of $\mathcal{A}$ making an error on a fresh example x depends only on the type of the input sequence S and the relative position of x in T with respect to S, up to a γ-approximation factor.

As a consequence of the Ramsey theorem for trees, it turns out that every algorithm can be reduced to an algorithm that has approximately comparison-based loss.

Lemma 13 (Every Algorithm has an Approx. Comparison-based Loss on a Large Subtree) *Let $\mathcal{A}$ be a (possibly randomized) k-list algorithm that is defined over input samples of size m over a domain $\mathcal{X}$, and let T be a $(k+1)$-ary mistake tree of depth n whose vertices are labeled by instances from $\mathcal{X}$. Then, there exists a subtree T' of T of depth $\frac{\log_{(m+1)}(n)}{2^{a(k+1)^{m+1} m \log m}}$, where $a < 24$ is a universal numerical constant, such that $\mathcal{A}$ has $\left(\frac{1}{100m}\right)$-comparison-based loss on T'.*

The upcoming lemma, in conjunction with Theorem 13 and Theorem 8, implies Theorem 2, as we will prove shortly.

Lemma 14 *[Sample Complexity for Privately Learning Trees] Let T be a $(k+1)$-ary mistake tree of depth n and let $\mathcal{A}$ be a k-list algorithm defined over input samples of size m. Assume that*

1. *$\mathcal{A}$ is $(\epsilon, \delta(m))$-differentially private for some $\epsilon \leq \log\left(\frac{400k^2+1}{400k^2}\right)$, and $\delta(m) \leq \frac{1}{200k^2m^2}$.*
2. *$\mathcal{A}$ has $\left(\frac{1}{100m}\right)$-comparison-based loss on T.*
3. *$\mathcal{A}$ is an (α, β)-accurate empirical k-list learner for T [9], where $\alpha = \beta = \frac{k!}{10^4(k+1)^{k+2}}$.*

Then, $m = \Omega(\log^\star n)$.

9. Here, by empirical learner for T, we mean an empirical learner with respect to input samples that are realizable by (a branch of) T.

B.1. Proof of Theorem 2

Theorem 2 follows from Theorems 8, 13 and 14.

Proof Let $\mathcal{H}$ be a concept class over an arbitrary label domain $\mathcal{Y}$, and assume that $\mathcal{H}$ shatters a k-Littlestone tree T of depth d. Let $\mathcal{A}$ be any $(\epsilon, \delta(m))$-differentially private k-list learner for $\mathcal{H}$, with $\epsilon, \delta(m)$ as in Theorem 2. By Theorem 8, we can further assume that $\mathcal{A}$ is an (α, β)-accurate empirical learner for $\mathcal{H}$, for $\alpha = \beta = \frac{k!}{10^4(k+1)^{k+2}}$, as the sample complexity of a private empirical learner increases only by a multiplicative constant factor. By Theorem 13, there exists a subtree T' of T, of depth $\frac{\log_{(m+1)}(d)}{2^{a(k+1)^{m+1}m\log m}}$ for some universal numerical constant $a < 24$, such that $\mathcal{A}$ has $\left(\frac{1}{100m}\right)$-comparison-based loss on T'. Finally, by Theorem 14 we conclude that

$$m \geq \Omega\left(\log^\star\left(\frac{\log_{(m+1)}(d)}{2^{a(k+1)^{m+1}m\log m}}\right)\right).$$

Let $t = \log^\star(d)$ and suppose $m \leq \frac{t}{16\log(k+1)}$ (else $m = \Omega(\log^\star d)$ as we treat k as constant, and we are done). We claim that $\log^\star\left(\frac{\log_{(m+1)}(d)}{2^{a(k+1)^{m+1}m\log m}}\right) = \Omega(\log^\star d)$, and therefore $m \geq \Omega(\log^\star(d))$, which concludes the proof. Note that by the definition of the $\log^\star$ function, $\mathsf{twr}_{(t)}(1) < d \leq \mathsf{twr}_{(t+1)}(1)$. The claim follows from the following calculation:

$$\begin{aligned}
\log^\star\left(\frac{\log_{(m+1)}(d)}{2^{a(k+1)^{m+1}m\log m}}\right) &= 1 + \log^\star\left(\log_{(m+2)}(d) - a(k+1)^{m+1}m\log m\right) \\
&\qquad\qquad \text{(definition of } \log^\star\text{)} \\
&\geq 1 + \log^\star\left(\mathsf{twr}_{(t-(m+2))}(1) - a(k+1)^{m+1}m\log m\right) \\
&\qquad\qquad (d > \mathsf{twr}_{(t)}(1)) \\
&\geq 1 + \log^\star\left(\mathsf{twr}_{(t/2)}(1) - a(k+1)^{m+1}m\log m\right) \\
&\qquad\qquad \text{(holds for } t \geq 5 \text{ since } m \leq \tfrac{t}{16\log(k+1)}\text{)} \\
&\geq 1 + \log^\star\left(\mathsf{twr}_{(t/2)}(1) - 2^{t/2}\right) \\
&\qquad\qquad (\forall m:\ 24(k+1)^{m+1}m\log m \leq (k+1)^{8m} \leq 2^{t/2}) \\
&= 1 + \log^\star\left(\mathsf{twr}_{(t/2)}(1)\right) \quad \text{(holds for } t \geq 10\text{, see justification below)} \\
&= t/2. \qquad\qquad \text{(definition of } \log^\star\text{)}
\end{aligned}$$

Therefore, for large enough d, $\log^\star\left(\frac{\log_{(m+1)}(d)}{2^{a2^{m\log(k+1)}m\log m}}\right) \geq \frac{1}{2}\log^\star d$, as desired. It is left to justify the second-to-last equality. It is enough to show that $\mathsf{twr}_{(x)}(1) - 2^x > \mathsf{twr}_{(x-1)}(1)$ for $x \geq 5$. And indeed $\mathsf{twr}_{(x)}(1) - \mathsf{twr}_{(x-1)}(1) \geq \frac{1}{2}\mathsf{twr}_{(x)}(1) \geq 2^x$ for $x \geq 5$. ■

Therefore, it is left to prove Theorems 13 and 14.

B.2. Proof of Theorem 13

Proof Define a coloring of $(m+2)$-chains of T as follows. Let $C = \{x_1 < \ldots < x_{m+2}\}$ be an $(m+2)$-chain in T. Recall that C corresponds to a sample $S = \big((x_1, y_1), \ldots, (x_{m+1}, y_{m+1})\big)$ of size $m+1$ where $(y_1, \ldots, y_{m+1}) = \boldsymbol{t}(C)$ is the type of C. For each $i \in \{1, \ldots, m+1\}$, let S^{-i} denote the sample $S \setminus (x_i, y_i)$. Set $q_i(C)$ to be the fraction of the form $\frac{r}{100m}$ that is closest

to $A_{S^{-i},x_i}(y_{x_i})$ (in case of ties pick the smallest such fraction). The color assigned to C is the list $(q_1(C),\ldots,q_{m+1}(C))$. [10]

Therefore, the total number of colors is at most $c \coloneqq (100m+1)^{m+1}$. By Ramsey theorem for trees (Theorem 11) there exists a subtree T' that is type-monochromatic with respect to the above coloring of depth

$$\begin{aligned} d \geq & \frac{\log_{(m+1)}(n)}{5\cdot(k+1)^m c^{(k+1)^{m+1}}\log c} \\ = & \frac{\log_{(m+1)}(n)}{5\cdot(k+1)^m(100m+1)^{(m+1)(k+1)^{m+1}}(m+1)\log(100m+1)} \\ = & \frac{\log_{(m+1)}(n)}{2^{\log 5+m\log(k+1)+\log(100m+1)(m+1)(k+1)^{m+1}+\log(m+1)+\log\log(100m+1)}} \\ \geq & \frac{\log_{(m+1)}(n)}{2^{a(k+1)^{m+1}m\log m}}, \end{aligned}$$

where $1 < a < 24$ is a universal numerical constant. For every possible type $\boldsymbol{t} \in \{0,1\}^{m+1}$ and $i \in \{0,\ldots,m\}$, set $p_{\boldsymbol{t},i}$ to be $q_{i+1}(C)$ where C is any $\boldsymbol{t}$-typed $(m+2)$-chain in T'. Note that $p_{\boldsymbol{t},i}$ is well defined since T' is type-monochromatic. It is straightforward to verify that $\mathcal{A}$ has $\left(\frac{1}{100m}\right)$-comparison-based loss on T' with respect to $\{p_{\boldsymbol{t},i}\}$, as wanted. ■

B.3. Proof of Theorem 14

The proof of Theorem 14 follows the same procedure as the reduction from the interior point problem given by Fioravanti et al. (2024). The reduction is carried out by constructing an algorithm $\tilde{\mathcal{A}}$ designed to solve the interior point problem. For the sake of the upcoming analysis of the algorithm, we assume that the input points to the interior point problem are not too close to each other, as justified by the following lemma.

Lemma 15 *Let $\mathcal{A} : [n]^m \to [n]$ be an (ϵ,δ)-differentially private algorithm, and let $C(n) \leq \log^2 n$. Assume that for every input sequence $x_1,\ldots,x_m$ such that $min_{i\neq j}|x_i - x_j| \geq C(n)$,*

$$\Pr[\min x_i \leq \mathcal{A}(x_1,\ldots,x_m) \leq \max x_i] \geq \frac{3}{4}.$$

Then, $m \geq \Omega(\log^\star n)$.

Proof By rescaling, $\mathcal{A}$ solves the interior point problem on a domain of size $\frac{n}{C(n)}$. Therefore, by Theorem 9, the sample complexity satisfies $m \geq \Omega\left(\log^\star\left(\frac{n}{C(n)}\right)\right) = \Omega\left(\log^\star\left(\frac{n}{\log^2 n}\right)\right) = \Omega(\log^\star n)$. Indeed, the second equality holds since by the definition of the $\log^\star$ function, $\mathrm{twr}_{(t+2)}(1) \geq 2^{\frac{n}{\log^2 n}}$, where $t = \log^\star\left(\frac{n}{\log^2 n}\right)$. Furthermore, $2^{\frac{n}{\log^2 n}} \geq n$ for large enough n. Therefore, again by definition, $t+1 \geq \log^\star n$ for large enough n, which implies that $\log^\star\left(\frac{n}{\log^2 n}\right) \geq \frac{1}{2}\log^\star n$ for large enough n. ■

10. Notice that the color assigned to C depends only weakly on the last vertex x_{m+2} via the label y_{m+1}. We find it more convenient and less cumbersome to increase the size of the chain by one rather than keeping track of the labels.

Before describing the algorithm, we will introduce notation which will be used in this proof. Given a branch B in a tree and an example z on B, we denote $b_z = y_z$ where $(z, y_z) \in B$.

Definition 16 (Almost-correct interval) *Let B be a branch in a $(k+1)$-ary mistake tree T, and let h be a k-labeled hypothesis over the domain of T. Denote by $Z = (z_1 < \ldots < z_l)$ a sequence of consecutive points of length l in B. We say that Z is an almost-correct interval with respect to h, if*

$$\sum_{i=1}^{l} \mathbb{1}[b_{z_i} \notin h(z_i)] \leq \frac{1}{2(k+1)} \cdot l.$$

Note that the guaranty of accumulated loss less than $\frac{1}{2(k+1)} \cdot l$ is better than a random guess, which has an accumulated loss of $\frac{1}{k+1} \cdot l$. In other words, having an almost-correct interval implies that h follows the branch with a relatively high probability (compared to a random guess).

Reduction from interior point problem. Let T be a $(k+1)$-ary mistake tree of depth n, and $\mathcal{A}$ be a k-list algorithm as in Theorem 14. Let $d_1 \ldots d_m \in [n]$ be natural numbers, the input to the interior point problem. For convenience, assume they are ordered $d_1 \leq \ldots \leq d_m$. Additionally, assume that $d_{i+1} - d_i > \log^2 n$ for all $1 \leq i < m$. Define the algorithm $\tilde{\mathcal{A}}$ as follows.

Algorithm 1 $\tilde{\mathcal{A}}$ (Reduction from IPP, Fioravanti et al. (2024))

Input: $d_1 \ldots, d_m$.
- Sample uniformly at random a branch $B \sim \texttt{Branches}(T)$.
- $S \leftarrow ((x_1, y_1), \ldots (x_m, y_m))$, where x_i is the point of depth d_i in B, and $y_i = b_{x_i}$.
- Sample $h \sim \mathcal{A}(S)$.
- Search for almost-correct intervals $(z_1 \ldots z_l)$ with respect to h, of length $l = \lfloor \log^2 n \rfloor$, where $n = \texttt{depth}(T)$.

Output: Output $\max \left\{ \texttt{depth}(z_1) \mid Z = (z_1 \ldots z_l) \text{ is an almost-correct interval of length } l = \lfloor \log^2 n \rfloor \right\}$.
In other words, output the depth of the first point of the deepest almost-correct interval. If there are no almost-correct intervals of length $l = \lfloor \log^2 n \rfloor$, return n.

From now on, when we refer to Z as an almost-correct interval, we mean that it is an almost-correct interval with respect to $\mathcal{A}(S)$, and that its length is $l = \lfloor \log^2 n \rfloor$.

Proposition 17 *Let $\mathcal{A}$ be a k-list learner as in Theorem 14. Then, $\tilde{\mathcal{A}}$ is $(\epsilon, \delta(m))$-differentially private, and with probability at least $\frac{3}{4}$ its output lies between d_1 and d_m.*

Theorem 14 is a direct corollary of Theorems 15 and 17. In order to prove Theorem 17, we first need to collect some lemmas.

Let B and S be the sampled branch and sequence, respectively, as described in Algorithm 1. The key idea of the proof is to show that the probability of having an almost-correct interval below S (i.e. $x_m < z_1$) is low, while the probability of having an almost-correct interval within S (i.e. $x_1 < z_1 < z_l < x_m$) is high. This ensures that, with high probability, the output of $\tilde{\mathcal{A}}$ is an interior point as desired.

Lemma 18 (Almost-correct intervals below S are unlikely) *Let $S = ((x_1, y_1), \dots (x_m, y_m))$ be the sequence from the description of Algorithm 1. Then, the probability that there is an almost-correct interval $(z_1 < \dots < z_l)$ in the branch B with $x_m < z_1$, is at most $n \cdot \exp\left(-\frac{1}{8(k+1)}\lfloor \log^2 n \rfloor\right)$.*

Proof Let $l = \lfloor \log^2 n \rfloor$ and let $Z = (z_1 < \dots < z_l)$ be a sequence of consecutive examples on B. Assume that Z starts below S on T, i.e. $x_m < z_1$. We can conceptualize the randomness of the reduction algorithm $\tilde{\mathcal{A}}$ in the following manner: Initially, a fair coin is independently tossed $d_m + 1$ times (recall that d_m is the largest input). These coin tosses determine the first $d_m + 1$ turns of the branch B, which in turn determines the sample S, the input for $\mathcal{A}$. Subsequently, the coin is independently tossed $n - d_m - 1$ more times, completing the determination of the branch B. This illustrates that, for every example z that is below S, $\mathcal{A}_S(z)$ is independent of b_z, and $\Pr[b_z \notin h(z)] = \frac{1}{k+1}$. Therefore, $\mathbb{E}\left[\sum_{i=1}^{l} \mathbb{1}[b_{z_i} \notin h(z_i)]\right] = \frac{1}{k+1} l$, and by applying a standard Chernoff bound,

$$\Pr_{B, h \sim \mathcal{A}(S)}\left[\sum_{i=1}^{l} \mathbb{1}[b_{z_i} \notin h(z_i)] \leq \frac{1}{2(k+1)} l\right] \leq \exp\left(-\frac{l}{8(k+1)}\right).$$

By a simple union bound, we derive that the probability that there is an almost-correct interval below S is at most $n \cdot \exp\left(-\frac{l}{8(k+1)}\right) = n \cdot \exp\left(-\frac{1}{8(k+1)}\lfloor \log^2 n \rfloor\right)$.

■

Lemma 19 (Almost-correct intervals within S are likely) *Let $S = ((x_1, y_1), \dots (x_m, y_m))$ be the sequence from the description of Algorithm 1. Then, the probability that there is no almost-correct interval $(z_1 < \dots < z_l)$ on the branch B with $x_1 < z_1 < z_l < x_m$, is at most*

$$\exp\left(-\frac{\mu_k^2}{6(k+1)} m\right) + 2(k+1) \cdot \eta,$$

where $\mu_k = \frac{(k+1)!}{(k+1)^{k+1}}$ and $\eta = 2k(e^{\epsilon} - 1 + \delta(m)) + \frac{2}{100m} + \frac{20}{10^4(k+1)}$.

The proof of Lemma 19 consists of three parts. First, we show that with high probability, there exist $k+1$ consecutive examples in S, $x_i, \dots, x_{i+k}$, with $k+1$ distinct labels (i.e. all $k+1$ possible labels), and with small error of $\mathcal{A}$ on each of them. This argument relies on the fact that $\mathcal{A}$ is an empirical learner for T. In the second step, we use the properties of privacy and comparison-based error of $\mathcal{A}$ to show that on each point $x_i < x < x_{i+k}$ on the branch B, the error of $\mathcal{A}$ remains small. Finally, applying Markov's inequality, we conclude that for any interval of length l between x_i and x_{i+k}, the probability to not be an almost-correct interval is small.

We call an example x on the branch B correct with parameter ξ (or ξ-correct), if $\mathcal{A}_{S,x}(b_x) < \xi$. We further say that a set of examples in B is correct with parameter ξ (or ξ-correct), if any example in the set is ξ-correct.

Lemma 20 *Let $S = ((x_1, y_1), \dots (x_m, y_m))$ be the sequence from the description of Algorithm 1, and let $\xi = \frac{20}{10^4(k+1)}$. Then, with probability $\geq 1 - \exp\left(-\frac{\mu_k^2}{6(k+1)} m\right)$, there are $k+1$ consecutive points in S, $x_i, \dots, x_{i+k}$, for some $i \in \{1, \dots, m-k\}$, such that the two following properties hold.*

1. $\{y_i, \ldots, y_{i+k}\} = \{1, \ldots, k+1\}$,
2. $\{x_i, \ldots, x_{i+k}\}$ *is* ξ*-correct, with* $\xi = \frac{20}{10^4(k+1)}$,

where $\mu_k = \frac{(k+1)!}{(k+1)^{k+1}}$.

Proof For simplicity, we divide the examples in S into $\lfloor m/(k+1) \rfloor$ disjoint parts, each contains $k+1$ consecutive examples from S, and show that with probability of at least $1 - \exp\left(-\frac{\mu_k^2}{6(k+1)}m\right)$, one of this parts satisfies the two properties. For this goal, we define for each $1 \leq i \leq m-k$, a random variable X_i as follows. X_i gets the value 1 if $x_i, \ldots, x_{i+k}$ have $k+1$ distinct labels and otherwise, it gets 0. Since the branch B is chosen uniformly at random, the expectation of X_i is $\mu_k = \frac{(k+1)!}{(k+1)^{k+1}}$. Now look at all random variables X_i with $i = 1 + c(k+1)$ for some integer c. The number of such variables is $\lfloor m/(k+1) \rfloor$, and they are IID. Let $X = \sum_{j=0}^{\lfloor m/(k+1) \rfloor - 1} X_{1+j(k+1)}$ denote their sum. By Chernoff,

$$\begin{aligned}
\Pr\left[X \leq \mathbb{E}[X] - \left\lfloor \frac{m}{k+1} \right\rfloor \cdot \frac{\mu_k}{2}\right] &\leq \exp\left(-2\left(\frac{\mu_k}{2}\right)^2 \left\lfloor \frac{m}{k+1} \right\rfloor\right) \\
&\leq \exp\left(-\frac{\mu_k^2}{2}\frac{m}{3(k+1)}\right) \\
&= \exp\left(-\frac{\mu_k^2}{6(k+1)}m\right).
\end{aligned}$$

On the other hand,

$$\Pr\left[X \geq \mathbb{E}[X] - \left\lfloor \frac{m}{k+1} \right\rfloor \cdot \frac{\mu_k}{2}\right] = \Pr\left[X \geq \left\lfloor \frac{m}{k+1} \right\rfloor \cdot \frac{\mu_k}{2}\right] \leq \Pr\left[X \geq \frac{\mu_k}{6(k+1)}m\right].$$

Hence, with probability of at least $1 - \exp\left(-\frac{\mu_k^2}{6(k+1)}m\right)$, the number of $k+1$ consecutive examples in S with $k+1$ distinct labels is at least $\frac{\mu_k}{6(k+1)}m$.

Consider the case when there exist at least $\frac{\mu_k}{6(k+1)}m$ of $k+1$ consecutive examples in S with $k+1$ distinct labels, and assume towards contradiction that the second property of ξ-correct does not hold for any such $k+1$ consecutive examples. $\mathcal{A}$ is an (α, β)-empirical list learner and therefore,

$$\frac{1}{m}\sum_{i=1}^{m} \mathcal{A}_{S,x_i}(y_i) \leq \alpha + \beta = \frac{2\mu_k}{10^4(k+1)^2}.$$

But by the assumption above,

$$\frac{1}{m}\sum_{i=1}^{m} \mathcal{A}_{S,x_i}(y_i) \geq \frac{\mu_k}{6(k+1)}\xi = \frac{10\mu_k}{3 \cdot 10^4(k+1)^2},$$

which leads to a contradiction. Therefore, with probability of at least $1 - \exp\left(-\frac{\mu_k^2}{6(k+1)}m\right)$, there are $k+1$ consecutive examples in S satisfying the two properties as wanted. ■

Lemma 21 *Let $S = ((x_1, y_1), \dots (x_m, y_m))$ be the sequence from the description of Algorithm 1, and suppose that $x_i, \dots, x_{i+k}$ are $k+1$ consecutive examples in S satisfying the two properties from Lemma 20. Let x be an example in B such that $x_i \leq x \leq x_{i+k}$. Then,*

$$\mathcal{A}_{S,x}(b_x) \leq \eta,$$

where $\eta = 2k(e^\epsilon - 1 + \delta(m)) + \frac{2}{100m} + \frac{20}{10^4(k+1)}$.

Proof If $x \in \{x_i, \dots, x_{i+k}\}$, then x is correct with parameter $\frac{20}{10^4(k+1)} < \eta$ and we are done. Assume that $x \notin \{x_i, \dots, x_{i+k}\}$. Since $x_i, \dots, x_{i+k}$ have distinct labels, there is $x' \in \{x_i, \dots, x_{i+k}\}$ with the same label y as x. Without loss of generality, assume that x' is below x, and change the sequence S as follows. Take all the examples in S that are between x and x', including x', and replace them with examples on B that are outside the interval (x, x'), to get a new sample $\tilde{S}$ of length m. Notice that by this construction,

$$\texttt{loc}_{\tilde{S}}(x) = \texttt{loc}_{\tilde{S}}(x').$$

Further, x and x' have the same label y, so that the sequences $\tilde{S}^{+x}$ and $\tilde{S}^{+x'}$ have the same type. The algorithm $\mathcal{A}$ has $\left(\frac{1}{100m}\right)$-comparison-based loss on T, so we can deduce that,

$$|\mathcal{A}_{\tilde{S},x}(y) - \mathcal{A}_{\tilde{S},x'}(y)| < \frac{2}{100m}.$$

Note that for every $\epsilon, \delta \geq 0$, if $a, b \in [0,1]$ satisfy $a \overset{\epsilon,\delta}{\approx} b$, then $|a - b| \leq e^\epsilon - 1 + \delta$. Indeed,

$$a - b \leq (e^\epsilon - 1) \cdot b + \delta \leq e^\epsilon - 1 + \delta,$$
$$b - a \leq (e^\epsilon - 1) \cdot a + \delta \leq e^\epsilon - 1 + \delta.$$

Since $\mathcal{A}$ is (ϵ, δ)-differentially private, and the sequence $\tilde{S}$ differs from the sequence S by at most k entries, we can transform from S to $\tilde{S}$ by at most k replacements of a single example at each time. Therefore,

$$|\mathcal{A}_{S,x}(y) - \mathcal{A}_{\tilde{S},x}(y)| \leq k(e^\epsilon - 1 + \delta),$$
$$|\mathcal{A}_{S,x'}(y) - \mathcal{A}_{\tilde{S},x'}(y)| \leq k(e^\epsilon - 1 + \delta).$$

Combining all together we get:

$$\begin{aligned}\mathcal{A}_{S,x}(y) &\leq |\mathcal{A}_{S,x}(y) - \mathcal{A}_{\tilde{S},x}(y)| + |\mathcal{A}_{\tilde{S},x}(y) - \mathcal{A}_{\tilde{S},x'}(y)| + |\mathcal{A}_{\tilde{S},x'}(y) - \mathcal{A}_{S,x'}(y)| + \mathcal{A}_{S,x'}(y) \\ &\leq 2k(e^\epsilon - 1 + \delta) + \frac{2}{100m} + \frac{20}{10^4(k+1)}.\end{aligned}$$

■

Proof [Proof of Theorem 19] Assume that $x_i, \dots, x_{i+k}$ are $k+1$ consecutive examples in S satisfying the two properties from Theorem 20. Let $z_1 < \dots < z_l$ be an interval such that $x_i < z_1 < \dots < z_l < x_{i+k}$. By Theorem 21,

$$\mathop{\mathbb{E}}_{B, h \sim \mathcal{A}(S)} \left[\sum_{i=1}^{l} \mathbb{1}[b_{z_i} \notin h(z_i)] \right] \leq \eta \cdot l.$$

By Markov,

$$\Pr\left[\sum_{i=1}^{l} \mathbb{1}[b_{z_i} \notin h(z_i)] \geq \frac{1}{2(k+1)} \cdot l\right] \leq 2(k+1) \cdot \eta.$$

By Theorem 20, the probability for having such $x_i, \ldots, x_{i+k}$ examples in at least $1 - \exp\left(-\frac{\mu_k^2}{6(k+1)}m\right)$. All in all, we conclude that the probability of not having an almost-correct interval inside S is less than $\exp\left(-\frac{\mu_k^2}{6(k+1)}m\right) + 2(k+1) \cdot \eta$. ■

We turn to prove Theorem 17 and show that $\tilde{\mathcal{A}}$ is $(\epsilon, \delta(m))$-differentially private, and with probability at least $3/4$ its output is an interior point.

Proof [Proof of Theorem 17]

Privacy: Let D and D' be neighboring datasets. Consider the output distributions $\tilde{\mathcal{A}}(D)$ and $\tilde{\mathcal{A}}(D')$. We couple these distributions by selecting the same random branch B in the first step of the algorithm. Hence, the samples S and S' that are input to $\mathcal{A}$ are also neighbors. Since $\mathcal{A}$ is (ϵ, δ)-DP, it follows that the distributions $\mathcal{A}(S)$ and $\mathcal{A}(S')$ are (ϵ, δ)-indistinguishable. Since the outputs $\tilde{\mathcal{A}}(D)$ and $\tilde{\mathcal{A}}(D')$ are functions of $\mathcal{A}(S)$ and $\mathcal{A}(S')$, by post-processing (Theorem 7), it follows that $\tilde{\mathcal{A}}(D)$ and $\tilde{\mathcal{A}}(D')$ are also (ϵ, δ)-indistinguishable. Hence, $\tilde{\mathcal{A}}$ is (ϵ, δ)-DP.

Utility: By Theorem 18, the probability that there is an almost-correct interval below S is at most

$$n \cdot \exp\left(-\frac{1}{8(k+1)}\lfloor \log^2 n \rfloor\right).$$

By Theorem 19, the probability that there is no almost-correct interval within S, is at most

$$\exp\left(-\frac{\mu_k^2}{6(k+1)}m\right) + 2(k+1) \cdot \eta,$$

where $\mu_k = \frac{(k+1)!}{(k+1)^{k+1}}$ and $\eta = 2k(e^\epsilon - 1 + \delta(m)) + \frac{2}{100m} + \frac{20}{10^4(k+1)}$. Hence, the probability that $\tilde{\mathcal{A}}$ does not output an interior point is as most

$$\begin{aligned}
&n \cdot \exp\left(-\frac{\lfloor \log^2 n \rfloor}{8(k+1)}\right) + \exp\left(-\frac{\mu_k^2}{6(k+1)}m\right) + 2(k+1) \cdot \eta \\
=&n \cdot \exp\left(-\frac{\lfloor \log^2 n \rfloor}{8(k+1)}\right) + \exp\left(-\frac{\mu_k^2}{6(k+1)}m\right) + 4k(k+1)(e^\epsilon - 1 + \delta(m)) + \frac{4(k+1)}{100m} + 40 \cdot 10^{-4}.
\end{aligned}$$

Each one of the five summands is smaller than $1/20$, by the choice of $\epsilon = \log\left(\frac{400k^2+1}{400k^2}\right), \delta(m) = \frac{1}{200k^2m^2}$, and for large enough [11] m and n. ■

Appendix C. Private k-List-Learnability Implies Finite k-Monotone Dimension

In this section, we prove Theorem 4.

11. Note that, without loss of generality, we may assume that $m \geq 1/\alpha$ since $\mathcal{A}$ is an (α, β)-accurate learner for T. Additionally, the assumption that n is large enough is concealed in the big Ω notation.

C.1. Proof of Theorem 4

Let $\mathcal{M}_k(\mathcal{X})$ denote the class of all monotone functions with $k+1$ labels over $\mathcal{X}$. To prove Theorem 4, it suffices to establish the following lemma. This is because (i) any class with k-monotone dimension at least n contains a copy of $\mathcal{M}_k([n])$ (ii) any k-list learner can be converted to k-list learner for which its prediction list is among the $k+1$ labels from the definition of the k-monotone dimension (while maintaining utility and privacy), by a simple post-processing step: if the learner outputs a hypothesis that predicts a label outside this set, it is replaced with one of the $k+1$ labels.

Lemma 22 *Let $k \geq 1$. Let $\mathcal{A}$ be an $(1/200k(k+), 1/200k(k+1))$-accurate k-list learning algorithm for the class $\mathcal{M}_k([n])$ with sample complexity m, satisfying $(\epsilon, \delta(m))$-differential privacy for $\epsilon = 0.1$ and $\delta(m) \leq \frac{1}{6(200km)^4 \log^2(200km)}$. Then the following bound holds:*

$$m = \Omega(\log^\star n).$$

To prove Theorem 22, we begin by introducing some notations and definitions that will be useful throughout the proof.

Introducing notations. Given a linearly ordered domain $\mathcal{X}$, a sequence $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big)$ is *ordered* if $x_1 < x_2 < \ldots < x_m$. Given an ordered sequence S and a test point $x \in \mathcal{X}$, the *location* of x in S is

$$\texttt{loc}_S(x) \coloneqq \max\{i \mid x_i < x\},$$

and if $x \leq x_1$ then define $\texttt{loc}_S(x) \coloneqq 0$. An ordered sequence $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big)$ is *increasing* if $y_1 \leq \ldots \leq y_m$. An ordered sequence is *balanced* if it is increasing, and every label out of $\mathcal{Y} = \{0, \ldots, \ell - 1\}$ appears the same amount of times. I.e., say $|S| = t \cdot \ell$ then $y_{i \cdot t+1} = \ldots = y_{(i+1)\cdot t} = i$, for $i = 0, \ldots, \ell - 1$.

Recall, given an input sample S and a test point $x \in \mathcal{X}$, we denote

$$\mathcal{A}_{S,x}(y) = \Pr_{h \sim \mathcal{A}(S)}[y \notin h] \text{ , where } y \in \mathcal{Y}.$$

Next, we adapt the notion of *comparison-based* algorithms, as introduced in Alon et al. (2019); Fioravanti et al. (2024), to the setting of k-list learners. Roughly speaking, a k-list algorithm $\mathcal{A}$ is comparison-based if the prediction of $\mathcal{A}$ on a test point x depends only on the labels of the sorted input sample S and the position of x inside of S. I.e. the algorithm makes all its decisions only based on how the elements of the input sample and the test point compare to each other, and not on their absolute values/locations.

Definition 23 (Approximately comparison-based on balanced samples) *Let $\mathcal{X}$ be a linearly ordered domain, let $\mathcal{Y} = \{0, \ldots, \ell - 1\}$ be the label space, and let $\gamma > 0$. A (randomized) k-list learner $\mathcal{A}$, defined over input samples of size m, is γ-comparison-based (CB) with respect to $\mathcal{X}$ if the following holds. There exist vectors $\boldsymbol{p}^{(0)}, \ldots, \boldsymbol{p}^{(m)} \in [0,1]^\ell$ such that for every increasing balanced input sequence $S \in (\mathcal{X} \times \mathcal{Y})^m$, and every test point $x \in \mathcal{X}$,*

$$\|\mathcal{A}_{S,x} - \boldsymbol{p}^{(i)}\|_\infty \leq \gamma,$$

where $i = \texttt{loc}_S(x)$.

The following two lemmas are key to proving Theorem 22.

Lemma 24 (Every algorithm is CB on a large subset) *Let $\mathcal{A}$ be a k-list learner that is defined over input samples of size m, over a linearly ordered domain $\mathcal{X}$ with $|\mathcal{X}| = n$, and a label space $\mathcal{Y} = \{0, \ldots, \ell - 1\}$. Then, there exist $\mathcal{X}' \subset \mathcal{X}$ such that $\mathcal{A}$ is $\left(\frac{1}{100km}\right)$-comparison-based on $\mathcal{X}'$ and*

$$|\mathcal{X}'| \geq \frac{\log_{(m)}(n)}{2^{O(\ell \cdot m \log km)}},$$

where the big O notation hides a universal constant value.

Lemma 25 (Lower-bounding the sample complexity of CB algorithms) *Let $\mathcal{A}$ be a k-list learner that is defined over input samples of size m, over a linearly ordered domain $\mathcal{X}$, and a label space $\mathcal{Y} = \{0, 1, \ldots, k\}$. Assume that*

1. *$\mathcal{A}$ is $(\epsilon, \delta(m))$ differentially private for $\epsilon = 0.1, \delta(m) \leq \frac{1}{6(200km)^4 \log^2(200km)}$.*
2. *$\mathcal{A}$ is $\left(\frac{1}{100km}\right)$-comparison-based on $\mathcal{X}$.*
3. *$\mathcal{A}$ is (α, β)-accurate empirical k-list learner for the class $\mathcal{M}_k(\mathcal{X})$, with $\alpha = \frac{1}{200k(k+1)}, \beta = \frac{1}{200k(k+1)}$.*

Then, $|\mathcal{X}| \leq 2^{O((km)^2 \log^2(km))}$, where the big O notation hides a universal constant value.

Proof [Proof of Theorem 22] First, by Theorem 8 we can assume that $\mathcal{A}$ empirically learns the class of step-functions over $[n]$. Then, by Theorems 24 and 25 there exists $\mathcal{X}' \subset [n]$ such that $\mathcal{A}$ is comparison based on $\mathcal{X}'$, and

$$\frac{\log_{(m)}(n)}{2^{O(km \log(km))}} \leq |\mathcal{X}'| \leq 2^{O((km)^2 \log^2(km))},$$

therefore we have $\log_{(m)}(n) \leq 2^{c \cdot (km)^2 \log^2(km)}$ for some constant c. By taking iterated logarithm $t = \log^\star(2^{c \cdot (km)^2 \log^2(km)}) = \log^\star(km) + O(1)$ times of both sides, we obtain $\log_{(m+t)}(n) \leq 1$, and therefore $\log^\star(n) \leq m + t = m + \log^\star(km) + O(1)$. As we treat k as a constant, it implies that $m \geq \Omega(\log^\star n)$ as required. ■

Therefore, it is left to prove Theorems 24 and 25.

C.1.1. PROOF OF THEOREM 24

The key to proving Theorem 24 is Ramsey theorem.

Theorem 26 (Ramsey Erdös and Rado (1952)) *Let $s > t \geq 2$ and q be integers, and let*

$$N \geq \mathsf{twr}_{(t)}(3sq \log q).$$

Then for every coloring of subsets of size t of a universe of size N using q colors, there is a homogeneous subset of size s.

Proof [Proof of Theorem 24] Define a coloring of $(m+1)$-subsets of $\mathcal{X}$ as follows. Let $X = \{x_1 < x_2 < \ldots < x_{m+1}\}$ be an $(m+1)$-subset of $\mathcal{X}$. For each $i \leq m+1$, denote $X^{-i} = X \setminus \{x_i\}$ and S^{-i} the balanced increasing sample on X^{-i}.

Next, for every $i \leq m+1$ observe $\mathcal{A}_{S^{-i},x_i} = \big(\mathcal{A}_{S^{-i},x_i}(0), \ldots, \mathcal{A}_{S^{-i},x_i}(\ell-1)\big)$, where $\mathcal{A}_{S^{-i},x_i}(j) = \Pr_{h\sim\mathcal{A}(S^{-i})}[j \notin h(x_i)]$. Set p^i_j to be the fraction of the form $\frac{t}{100km}$ closest to $\mathcal{A}_{S^{-i},x_i}(j)$ (in case of ties pick the smaller one). Now set $\boldsymbol{p}^{(i)} = (p^i_0, \ldots, p^i_{\ell-1})$. The color assigned to X is the list $(\boldsymbol{p}^{(1)}, \ldots, \boldsymbol{p}^{(m+1)})$.

Therefore, the total number of colors is $(100km+1)^{\ell(m+1)}$.[12] By applying Theorem 26 with $t \coloneqq m+1, q \coloneqq (100m+1)^{\ell(m+1)}$, and $N \coloneqq n$ there is an homogeneous subset $\mathcal{X}' \subseteq \mathcal{X}$ of size

$$|\mathcal{X}'| \geq \frac{\log_{(m)}(n)}{3(100km+1)^{\ell(m+1)}\ell(m+1)\log(100km+1)} = \frac{\log_{(m)}(n)}{2^{O(\ell\cdot m\log(km))}}$$

such that all $(m+1)$-subsets of $\mathcal{X}'$ have the same color. One can verify that $\mathcal{A}$ is $\big(\frac{1}{100km}\big)$-comparison-based on $\mathcal{X}'$. ∎

C.1.2. PROOF OF THEOREM 25

To prove Theorem 25 we first need to to establish the following claims.

Proposition 27 *Let $\mathcal{A}$ be as in Theorem 25 and let $\boldsymbol{p}^{(0)}, \ldots, \boldsymbol{p}^{(m)} \in [0,1]^{k+1}$ be the vectors promised by the definition of comparison-based algorithms. Then, there exists $0 < i \leq m$ such that*

$$\|\boldsymbol{p}^{(i)} - \boldsymbol{p}^{(i-1)}\|_\infty \geq \frac{3}{100km}.$$

Proof [Proof of Theorem 27] The proof uses the assumption that $\mathcal{A}$ empirically k-list learns the class of monotone functions with $k+1$ labels over $\mathcal{X}$, $\mathcal{M}_k(\mathcal{X})$. Let $S = \big((x_1,y_1),\ldots,(x_m,y_m)\big)$ be an increasing balanced realizable sequence. Recall, $\mathcal{A}_{S,x}(j) = \Pr_{h\sim\mathcal{A}(S)}[j \notin h(x)]$ for $j = 0,1,\ldots,k$, and $\sum_{j=0}^k \mathcal{A}_{S,x}(j) = 1$. The expected empirical loss of $\mathcal{A}$ on S is at most $\alpha+\beta$, since $\mathcal{A}$ is (α,β)-accurate empirical learner. Therefore,

$$\begin{aligned}
1-(\alpha+\beta) &\leq \mathbb{E}_{h\sim\mathcal{A}(S)}[1-\mathrm{L}_S(h)] \\
&= \sum_{j=0}^{k} \frac{1}{m}\left(\sum_{i=\frac{m}{k+1}\cdot j+1}^{\frac{m}{k+1}\cdot(j+1)} \Pr_{h\sim\mathcal{A}(S)}[j\in h(x_i)]\right) \\
&= \sum_{j=0}^{k} \frac{1}{m}\left(\sum_{i=\frac{m}{k+1}\cdot j+1}^{\frac{m}{k+1}\cdot(j+1)} (1-\mathcal{A}_{S,x_i}(j))\right)
\end{aligned}$$

By bounding each time k different summands from above by $\frac{k}{k+1}$ we have for every $j = 0,\ldots,k$

$$\frac{1}{m}\left(\sum_{i=\frac{m}{k+1}\cdot j+1}^{\frac{m}{k+1}\cdot(j+1)} (1-\mathcal{A}_{S,x_i}(j))\right) \geq \frac{1}{k+1} - (\alpha+\beta).$$

12. The number of colors can be reduced to $(100km+1)^{(\ell-1)(m+1)}$, because $p^i_{\ell-1}$ is a function of $p^i_0,\ldots,p^i_{\ell-2}$.

Therefore, by a simple averaging argument, there exist $i_j \in \left[\frac{mj}{k+1}+1, \frac{m(j+1)}{k+1}\right]$ for $j = 0, \ldots, k$ such that

$$1 - \mathcal{A}_{S,x_{i_j}}(j) \geq 1 - (k+1)\cdot(\alpha+\beta). \tag{1}$$

From now, on we will focus on $j = k$. Notice that

$$1 - \mathcal{A}_{S,x_{i_k}}(k) = \sum_{j' \neq k} \mathcal{A}_{S,x_{i_k}}(j'),$$

and therefore there exist $j' \neq k$ such that

$$\mathcal{A}_{S,x_{i_k}}(j') \geq \frac{1}{k} - \frac{k+1}{k}\cdot(\alpha+\beta).$$

On the other hand, $\mathcal{A}_{S,x_{i_{j'}}}(j') \leq (k+1)\cdot(\alpha+\beta)$ (by Equation (1)), and therefore

$$|\mathcal{A}_{S,x_{i_k}}(j') - \mathcal{A}_{S,x_{i_{j'}}}(j')| \geq \frac{1}{k} - \frac{k+1}{k}\cdot(\alpha+\beta) - (k+1)\cdot(\alpha+\beta) = \frac{1}{k} - \frac{(k+1)^2}{k}(\alpha+\beta).$$

Next, consider S' the sample where we replace $x_{i_{j'}}$ with x' satisfying $x_{i_{j'}-1} < x' < x_{i_{j'}}$, and S'' the sample where we replace x_{i_k} with x'' satisfying $x_{i_k-1} < x'' < x_{i_k}$. Note that $\texttt{loc}_{S'}(x_{i_{j'}}) = i_{j'}$ and $\texttt{loc}_{S''}(x_{i_k}) = i_k$. By privacy,

$$\mathcal{A}_{S',x_{i_{j'}}}(j') \leq e^{\epsilon}\cdot\mathcal{A}_{S,x_{i_{j'}}}(j') + \delta \leq e^{\epsilon}\cdot(k+1)(\alpha+\beta) + \delta < \frac{1}{4k},$$

(Holds for $\epsilon = 0.1$, $\alpha = \beta = \frac{1}{200k(k+1)}$, $\delta < \frac{1}{100k}$)

$$\mathcal{A}_{S'',x_{i_k}}(j') \geq (\mathcal{A}_{S,x_{i_k}}(j') - \delta)\cdot e^{-\epsilon} \geq \left(\frac{1}{k} - \frac{k+1}{k}\cdot(\alpha+\beta) - \delta\right)e^{-\epsilon} > \frac{1}{2k}.$$

(Holds for $\epsilon = 0.1$, $\alpha = \beta = \frac{1}{200k(k+1)}$, $\delta < \frac{1}{100k}$)

Now, since $\mathcal{A}$ is $\left(\frac{1}{100km}\right)$-comparison-based we have

$$\boldsymbol{p}_{j'}^{(i_j)} < \frac{1}{4k} + \frac{1}{100km},$$
$$\boldsymbol{p}_{j'}^{(i_k)} > \frac{1}{2k} - \frac{1}{100km}.$$

Therefore, there exists $i_{j'} \leq i \leq i_k$ such that

$$|\boldsymbol{p}_{j'}^{(i-1)} - \boldsymbol{p}_{j'}^{(i)}| > \frac{1/4k}{m} - \frac{1}{50km^2} \geq \frac{3}{100km},$$

which completes the proof.

■

Proposition 28 *Let $\mathcal{A}$ be as in Theorem 25 and let $n = |\mathcal{X}| - m$. Then, there exist distributions $\mathcal{P}_1, \ldots, \mathcal{P}_n$, a number $c \in [0,1]$, and events $E_1, \ldots, E_n$, such that the following holds:*

(i) for every i, j, $\mathcal{P}_i$ and $\mathcal{P}_j$ are $(\epsilon, \delta(m))$-indistinguishable for $\epsilon = 0.1$ and $\delta(m) \leq \frac{1}{6(200km)^4 \log^2(200km)}$; and

(ii) for every i, j,

$$\mathcal{P}_i(E_j) = \begin{cases} \leq c - \gamma & j < i \\ \geq c + \gamma & j > i, \end{cases}$$

where $\gamma = \frac{1}{200km}$.

Proof [Proof of Theorem 28] Let i be the index promised in Theorem 27. I.e. $\|\boldsymbol{p}^{(i)} - \boldsymbol{p}^{(i-1)}\|_\infty \geq \frac{3}{100km}$. W.l.o.g assume $|\boldsymbol{p}_0^{(i)} - \boldsymbol{p}_0^{(i-1)}| \geq \frac{3}{100km}$. Take $S = ((x_1, y_1), \ldots, (x_m, y_m))$ to be an increasing balanced sample such that the interval $I = \{x : x_{i-1} < x < x_i\}$ is of size $|\mathcal{X}| - m$. Now, for every $x \in I$ denote by S_x the sample obtained by replacing x_i with x in S (with the same labeling). Notice that since $\mathcal{A}$ is $(\frac{1}{100km})$-comparison-based, for every $x, x' \in I$

$$\mathcal{A}_{S_x, x'}(0) = \Pr_{h \sim \mathcal{A}(S_x)}[0 \notin h(x')] = \begin{cases} \leq \boldsymbol{p}_0^{(i-1)} + \frac{1}{100km} & x' < x \\ \geq \boldsymbol{p}_0^{(i)} - \frac{1}{100km} & x' > x. \end{cases}$$

Set $c = \frac{1}{2} \cdot (\boldsymbol{p}_0^{(i)} + \boldsymbol{p}_0^{(i-1)})$. Therefore,

$$\mathcal{A}_{S_x, x'}(0) = \Pr_{h \sim \mathcal{A}(S_x)}[0 \notin h(x')] = \begin{cases} \leq c - \frac{1}{200km} & x' < x \\ \geq c + \frac{1}{200km} & x' > x. \end{cases} \tag{2}$$

For every $x \in I$, we define $\mathcal{P}_x$ as the distribution $\mathcal{A}(S_x)$, and we define E_x as the event

$$\left\{ h \in \binom{\mathcal{Y}}{k}^{\mathcal{X}} : 0 \notin h(x) \right\}.$$

Note that for every $x, x' \in I$, the datasets S_x and $S_{x'}$ are neighboring. Since $\mathcal{A}$ is $(\epsilon, \delta(m))$-differentially private, the distributions $\mathcal{P}_x$ and $\mathcal{P}_{x'}$ are $(\epsilon, \delta(m))$-indistinguishable. Thus, the proof is complete.

■

Proposition 29 (Packing) *Let $c \in [0, 1], \gamma \leq \frac{1}{2}$, and let $\mathcal{P}_1, \ldots, \mathcal{P}_n$ be probability measures such that for all i, j, $\mathcal{P}_i$ and $\mathcal{P}_j$ are (ϵ, δ)-indistinguishable, for $\epsilon \leq 0.1$ and $\delta \leq \frac{1}{6\gamma^{-4} \log^2\left(\frac{1}{\gamma}\right)}$. Assume there exist events $E_1, \ldots, E_n$ such that for every i, j*

$$\mathcal{P}_i(E_j) = \begin{cases} \leq c - \gamma & j < i \\ \geq c + \gamma & j > i. \end{cases} \tag{3}$$

Then, $n \leq 2^{\gamma^{-2} \log^2(\gamma^{-1})}$.

Proof [Proof of Theorem 29] Set $T = \frac{1}{\gamma^2}\log^2\left(\frac{1}{\gamma}\right) - 1$, $D = \frac{1}{\gamma^2}\ln T$. Assume towards contradiction that $n > 2^{T+1}$. We consider the following binary search[13] over probability measures (over the same σ-algebra as the $\mathcal{P}_j$'s), defined using the events E_j's. In the first step of preforming the search on P, we consider $P(E_{n/2})$. If $P(E_{n/2}) \leq c$, then we continue recursively with $E_{n/2+1}, \ldots, E_n$. Otherwise, we continue recursively with $E_1, \ldots E_{n/2}$. We perform this search for T steps. After T steps, the result of the search is defined to be the index j of the event E_j that is supposed to be queried on the $T+1$ step. For example, for $T = 1$, if $P(E_{n/2}) \leq c$, then the result of the search is $3n/4$, and if $P(E_{n/2}) > c$, the result is $n/4$. Observe that

(i) by the assumption, there are exactly 2^T different search outcomes, and

(ii) for every possible outcome i, the result of applying the search over $\mathcal{P}_i$ is exactly i.

Let $\bar{X} = (X_1, \ldots, X_D) \sim \mathcal{P}_i^D$ be D IID samples from $\mathcal{P}_i$. Denote the empirical measure[14] induced by $\bar{X}$ by $\mathcal{P}_{i,\bar{X}}$. Next, consider performing the binary search over the empirical measure $\mathcal{P}_{i,\bar{X}}$, where i is a possible search outcome. We claim that with high probability, the result of the search will be exactly i. Quantitatively, denote by F_i the event

$$F_i = \{\bar{X} : \text{performing binary search on the empirical measure defined by } \bar{X} \text{ yields } i\}.$$

By a standard application of Chernoff and union bound,

$$\begin{aligned}
\mathcal{P}_i^D(F_i) &\geq 1 - T \cdot \Pr_{\bar{X}\sim\mathcal{P}_i^D}[|\mathcal{P}_{i,\bar{X}}(E_k) - \mathcal{P}_i(E_k)| \geq \gamma] \\
&\geq 1 - T\exp\left(-2\gamma^2 D\right) \\
&= 1 - \frac{1}{T} > \frac{2}{3},
\end{aligned}$$

where k is some index along the query branch.

On the other hand, since $\mathcal{P}_i^D$ and $\mathcal{P}_j^D$ are $(D\epsilon, D\delta)$-indistinguishable, we have

$$\begin{aligned}
\mathcal{P}_j^D(F_i) &\geq \exp\left(-D\epsilon\right) \cdot (\mathcal{P}_i^D(F_i) - D\delta) \\
&\geq \exp\left(-D\epsilon\right) \cdot \left(\frac{2}{3} - D\delta\right) \\
&\geq \frac{1}{2}\exp\left(-D\epsilon\right),
\end{aligned}$$

where the last inequality holds by the choice of δ. Recall, there are 2^T different search outcomes (Observation (i)), and therefore the events F_i's are mutually disjoint. Therefore,

$$\begin{aligned}
1 &\geq \mathcal{P}_j^D(\cup_i F_i) \\
&= \sum \mathcal{P}_j^D(F_i) \\
&\geq 2^T \cdot \frac{1}{2}\exp\left(-D\epsilon\right) = 2^{T-1}\exp\left(-D\epsilon\right),
\end{aligned}$$

13. A binary search over a domain A is modeled as a pair $(\mathcal{T}, \{p_v\})$, where $\mathcal{T}$ is a binary tree and $p_v : A \to \pm 1$ are predicates (assigned to internal vertices of $\mathcal{T}$). The result of a search over an item $a \in A$ is the leaf defined by the root-to-leaf walk on the tree according to the answers $\{p_v(a)\}$.

14. Given D independent samples $X_1, \ldots, X_D$ from a distribution P, the *empirical measure* is defined as $P_D(E) = \sum_{j=1}^{D} \delta_{X_j}(E)$, where δ_X is the Dirac measure.

however, this is a contradiction due to the choice of T, D, ϵ.

■

Finally, Theorem 25 is a direct result of Theorems 27 to 29.

Appendix D. Ramsey Theorem for b-ary Trees

In this section, we present the proof of Theorem 11, as given in Fioravanti et al. (2024), with the necessary modifications to extend it to general b-ary trees.

Denote by $\mathsf{R}_m(d; k, b)$ the smallest n that satisfies the condition in Theorem 11, i.e. $\mathsf{R}_m(d; k, b)$ is the smallest n such that for every coloring of m-chains with k colors, there exists a type-monochromatic subtree of depth d. The proof of Theorem 11 consists of two parts: the first part shows that $\mathsf{R}_m(d; k, b)$ is well defined (i.e. $\mathsf{R}_m(d; k, b) < \infty$), while the second part proves the quantitative upper bound stated in theorem. We start by citing the pigeonhole principle for trees, which serves as the base case for the inductive proof for coloring m-chains for any value of m.

Proposition 30 (Pigeonhole Principle for Trees, Alon et al. (2022)) *Let $d \in \mathbb{N}$ and let T be a complete b-ary tree of depth n. Then, for every coloring of its vertices with k colors, the following hold: If T has depth $n \geq dk$, it admits a subtree S of depth d;*

Proof [Proof of Theorem 11] We prove the theorem by induction on m. The case $m = 1$ follows immediately from Theorem 30 since a 1-chain is simply a vertex, hence a 1-chain coloring is a vertex coloring, and the promised monochromatic subtree is in particular type-monochromatic.

Assume that the statement holds for $m-1$, and denote $t = \mathsf{R}_{m-1}(d; k^b, b)$. Let T be a complete b-ary tree of depth n, where n is sufficiently large (the size of n will be determined later on in **??**). Let χ be an m-chain coloring of T using k colors. We introduce a recursive procedure constructing a subtree of T of depth t, denoted $T^\star$. Then, we define an $(m-1)$-chain coloring $\chi^\star$ of $T^\star$, such that any $\chi^\star$-type-monochromatic subtree of $T^\star$ is in fact type-monochromatic with respect to χ. Finally, we apply the induction hypothesis on $T^\star$ and $\chi^\star$, allowing us to obtain the desired type-monochromatic subtree. Throughout the proof we denote the vertices of $T^\star$ by u_σ, where for a string $\sigma \in [b]^i$, $i \in \{0, \ldots, t-1\}$, and $r \in [b]$, $u_\emptyset$ is the root of $T^\star$ and $u_{\sigma r}$ is the r'th child of u_σ.

We define by induction a sequence of trees S_σ and vertices u_σ as follows. For every $\sigma \in [b]^{\leq (m-2)}$, set u_σ to be the vertex in T represented by the sequence σ (the root of T is represented by the empty sequence). Next, for every $\sigma \in [b]^{m-2}$ set S_σ to be the subtree of T rooted at u_σ. Assume the subtree S_σ has been defined and u_σ is the root of S_σ, where $\sigma \in [b]^i$ is a b-ary sequence of length $i \geq m-2$. We define subtrees $S_{\sigma r}$ and vertices $u_{\sigma r}$ where $r \in [b]$, as follows.

1. Consider the r'th-subtree of S_σ, that is the subtree of S_σ emanating from the r'th child of the root u_σ. Define an equivalence relation on the vertices of the r'th-subtree of S_σ as follows:

$$x \equiv y$$
$$\iff$$
$$\forall A \subset \{u_{\sigma(0)}, u_{\sigma(1)}, \ldots, u_{\sigma(i-1)}\}, |A| = m-2 \ : \ \chi(A \cup \{u_\sigma, x\}) = \chi(A \cup \{u_\sigma, y\}),$$

where $\sigma(j)$ is the prefix of σ of length j (with $\sigma(0)$ being the empty sequence). Note that indeed every choice of $m-2$ vertices from the set $\{u_{\sigma(0)}, u_{\sigma(1)}, \ldots, u_{\sigma(i-1)}\}$, together with u_σ and a vertex from the r'th-subtree of S_σ, form an m-chain in T.

Observe that an equivalence class is determined by a sequence of $\binom{i}{m-2}$ colors, therefore there are at most $k^{\binom{i}{m-2}}$ such equivalence classes.

2. Apply the pigeonhole principle for trees (Theorem 30) on the r'th-subtree of S_σ, where the colors of the vertices are the equivalence classes defined in the previous step. Set $S_{\sigma r}$ to be the promised monochromatic subtree, and set $u_{\sigma r}$ to be its root.

We choose n to be sufficiently large so this procedure may be continued until $T^\star = \{u_\sigma\}_{\sigma \in [b]^{\leq t}}$ have been defined. See **??** for a more detailed discussion. Note that for every b-ary sequence σ of length i, and every $r \in [b]$,

$$\texttt{depth}(S_{\sigma r}) \geq \left\lfloor \frac{\texttt{depth}(S_\sigma) - 1}{k^{\binom{i}{m-2}}} \right\rfloor. \tag{4}$$

Next, define an $(m-1)$-chain coloring of $T^\star$, denoted $\chi^\star$, as follows.

$$\forall\ (m-1)\text{-chain } C \text{ in } T^\star\ :\ \chi^\star(C) = \big(\chi(C \cup \{u_{\sigma_0}\}), \chi(C \cup \{u_{\sigma_1}\}), \ldots, \chi(C \cup \{u_{\sigma_{b-1}}\})\big),$$

where u_{σ_r} is any vertex from $T^\star$ that belongs to the r'th subtree emanating from the last vertex of the chain C. (If the last vertex in C is at level t, meaning it is a leaf of $T^\star$, we just pick an arbitrary color for C out of the k^b possible colors.) Note that $\chi^\star$ is well-defined. Indeed, take an $(m-1)$-chain C in $T^\star$. If x, y are vertices that belong to the r'th subtree emanating from the last vertex of the chain C, then by construction $x \equiv y$, meaning $\chi(C \cup \{x\}) = \chi(C \cup \{y\})$.

Finally, by the induction hypothesis applied on $T^\star$ and $\chi^\star$, and by the choice of $t = \mathsf{R}_{m-1}(d; k^b, b)$, there exists a type-monochromatic subtree of $T^\star$ of depth d. In fact, this subtree is also type-monochromatic with respect to the original m-chain coloring χ. Indeed, if C and C' are two m-chains with the same type, then the first $m-1$ vertices of C and C' form an $(m-1)$-chains with the same type that have the same color with respect to $\chi^\star$. So by the definition of $\chi^\star$ it is affirmed that $\chi(C) = \chi(C')$. Therefore, we proved the finiteness of the Ramsey number $\mathsf{R}_m(d; k, b)$. It is left to obtain from the recursive procedure described here the upper bound for $\mathsf{R}_m(d; k, b)$ that is stated in the theorem. We provide a detailed calculation in Appendix D.1. ■

D.1. Quantitative Bound for Ramsey Number

Here we give an explicit calculation for the sufficient depth n that is required for the proof. Subsequently, we derive an upper bound for $\mathsf{R}_m(d; k, b)$.

The procedure described in the proof of Theorem 11 can be continued until step $t = \mathsf{R}_{m-1}(d; k^b, b)$ if for every sequence σ of length t, $\texttt{depth}(S_\sigma) \geq 0$. Consider a sequence σ of length t. To ease the notation, for every step $i \in \{m-2, \ldots, t\}$, $\texttt{depth}(S_{\sigma(i)})$ is denoted by d_i. Recall, by Equation (4) the following holds.

$$\begin{cases} d_{m-2} & = n - (m-2), \\ d_{i+1} & \geq \left\lfloor \frac{d_i - 1}{k^{\binom{i}{m-2}}} \right\rfloor. \end{cases} \tag{5}$$

Observe that if $d_i \geq 2k^{\binom{i}{m-2}}$ then

$$d_{i+1} \geq \left\lfloor \frac{d_i - 1}{k^{\binom{i}{m-2}}} \right\rfloor \geq \frac{d_i}{k^{\binom{i}{m-2}}} - 1 \geq \frac{d_i}{2k^{\binom{i}{m-2}}}. \tag{6}$$

If $d_i < 2k^{\binom{i}{m-2}}$, then

$$\left\lfloor \frac{d_i - 1}{k^{\binom{i}{m-2}}} \right\rfloor \in \{0, 1\} ,$$

meaning that the procedure terminates, or continues for one more last step, therefore we can assume that the bound in Equation (6) holds in every step i. By induction, using the recurrence relation in Equations (5) and (6),

$$d_i \geq \frac{n - (m-2)}{2^{i-(m-2)} k^{\sum_{j=m-2}^{i-1} \binom{j}{m-2}}} = \frac{n - (m-2)}{2^{i-(m-2)} k^{\binom{i}{m-1}}} , \tag{7}$$

because $\sum_{j=m-2}^{i-1} \binom{j}{m-2} = \binom{i}{m-1}$ (the left-hand-side counts the number $(m-1)$-subsets $S \subseteq [i]$, by partitioning them according to the largest element). For the procedure to continue t steps we require $d_t \geq 1$. Together with Equation (7) we deduce the following bound:

$$n \geq 2^{t-(m-2)} k^{\binom{t}{m-1}} + (m-2).$$

Notice that for every $m, k \geq 2$ and $t \geq m-2$,

$$2^{t-(m-2)} k^{\binom{t}{m-1}} + (m-2) \leq 2^{t-(m-2)} k^{t^{m-1}} + (m-2) \leq k^{2t^{m-1}}.$$

Therefore, choosing $n = k^{2t^{m-1}}$ is sufficient. Recall that $t = \mathsf{R}_{m-1}(d; k^b, b)$, therefore the following recursive relation is obtained.

$$\mathsf{R}_m(d; k, b) \leq k^{2\mathsf{R}_{m-1}(d;k^b,b)^{m-1}}. \tag{8}$$

From now on we will use the Knuth notation $a \uparrow b$ in place of a^b to ease the calculations, and recall that the Knuth's operator is right-associative, i.e. $a \uparrow b \uparrow c = a \uparrow (b \uparrow c)$. By applying Equation (8) repeatedly we obtain the following bound

$$\begin{aligned}
&\mathsf{R}_m(d; k, b) \\
&\leq k^2 \uparrow \mathsf{R}_{m-1}(d; k^2, b) \uparrow (m-1) \\
&\leq k^2 \uparrow (k^{2\cdot b} \uparrow (m-1)) \uparrow \mathsf{R}_{m-2}(d; k^{b^2}, b) \uparrow (m-2) \\
&\leq k^2 \uparrow (k^{2\cdot b} \uparrow (m-1)) \uparrow (k^{2\cdot b^2} \uparrow (m-2)) \uparrow \mathsf{R}_{m-3}(d; k^{b^3}, b) \uparrow (m-3) \\
&\leq \dots \\
&\leq k^2 \uparrow (k^{2\cdot b} \uparrow (m-1)) \uparrow (k^{2\cdot b^2} \uparrow (m-2)) \uparrow \dots \uparrow (k^{2\cdot b^{m-2}} \uparrow 2) \uparrow \mathsf{R}_1(d; k^{b^{m-1}}, b) \uparrow 1 \\
&= k^2 \uparrow (k^{2\cdot b} \uparrow (m-1)) \uparrow (k^{2\cdot b^2} \uparrow (m-2)) \uparrow \dots \uparrow (k^{2\cdot b^{m-2}} \uparrow 2) \uparrow dk^{b^{m-1}},
\end{aligned}$$

where $\mathsf{R}_1(d; k, b) = dk$ by the pigeonhole principle (Theorem 30). Denote

$$R_i = \begin{cases} dk^{b^{m-1}} \coloneqq R & \text{if } i = 1; \\ k^{2\cdot b^{m-i} \cdot i \cdot R_{i-1}} & \text{if } 1 < i < m; \\ k^{2\cdot R_{m-1}} & \text{if } i = m. \end{cases}$$

Using this notation, $\mathsf{R}_m(d; k, b) \leq R_m$.

Proposition 31 *For* $2 \leq i \leq m$,

$$R_i \leq \mathsf{twr}_{(i)}(c_i \cdot b^{m-2} R \log k),$$

where[15]

$$c_i = 4 + \sum_{j=3}^{i} \frac{\max\{1, \log_{(j-2)}(2 \cdot b^{m-j} j \log k)\}}{b^{m-2} R \log k}.$$

Proof [Proof of Theorem 31] Proof by induction on i. For $i = 2$,

$$R_2 = k^{4 \cdot b^{m-2} \cdot R} = \mathsf{twr}_{(2)}(c_2 \cdot b^{m-2} R \log k).$$

For $2 < i < m$,

$$\begin{aligned}
R_i &= k^{2 \cdot b^{m-i} \cdot i \cdot R_{i-1}} \\
&= \mathsf{twr}_{(2)}\left[2 \cdot b^{m-i} i \log k \cdot R_{i-1}\right] \\
&\leq \mathsf{twr}_{(2)}\left[2 \cdot b^{m-i} \cdot i \log k \cdot \mathsf{twr}_{(i-1)}(c_{i-1} \cdot b^{m-2} R \log k)\right] && \text{(by induction.)} \\
&= \mathsf{twr}_{(2)}\left[\mathsf{twr}_{(i-1)} \log_{(i-2)}(2 \cdot b^{m-i} i \log k) \cdot \mathsf{twr}_{(i-1)}(c_{i-1} \cdot b^{m-2} R \log k)\right] \\
&\leq \mathsf{twr}_{(2)}\left[\mathsf{twr}_{(i-1)}\left(\max\{1, \log_{(i-2)}(2 \cdot b^{m-i} i \log k)\}\right) \cdot \mathsf{twr}_{(i-1)}\left(c_{i-1} \cdot b^{m-2} R \log k\right)\right] \\
&\leq \mathsf{twr}_{(2)}\left[\mathsf{twr}_{(i-1)}(\max\{1, \log_{(i-2)}(2 \cdot b^{m-i} i \log k)\}) + c_{i-1} \cdot b^{m-2} R \log k)\right] && (\star) \\
&= \mathsf{twr}_{(2)}\left[\mathsf{twr}_{(i-1)}(c_i \cdot b^{m-2} R \log k)\right] && \text{(definition of } c_i\text{.)} \\
&= \mathsf{twr}_{(i)}(c_i \cdot b^{m-2} R \log k),
\end{aligned}$$

where the inequality marked with $(\star)$ holds since

$$\mathsf{twr}_{(n)}(x) \cdot \mathsf{twr}_{(n)}(y) \leq \mathsf{twr}_{(n)}(x \cdot y)$$

for $x, y \geq 1, n \geq 2$.

The case $i = m$ follows because $R_m = k^{2 \cdot R_{m-1}} \leq k^{2m \cdot R_{m-1}}$, and $k^{2m \cdot R_{m-1}} \leq \mathsf{twr}_{(m)}(c_m \cdot b^{m-2} R \log k)$, by using the above calculation for $2 < i < m$ one more time for $i = m$.

■

Corollary 32 (Upper Bound For Ramsey Number for Chains) *For every integers* $m \geq 2, d \geq m, k \geq 2, b \geq 2$,

$$\mathsf{R}_m(d; k, b) \leq \mathsf{twr}_{(m)}(5 \cdot b^{m-2} d k^{b^{m-1}} \log k).$$

Proof It suffices to bound

$$c_m = 4 + \sum_{j=3}^{m} \frac{\max\{1, \log_{(j-2)}(2 \cdot b^{m-j} j \log k)\}}{b^{m-2} R \log k}.$$

15. We use the convention $\sum_{k=n_1}^{n_2} f(k) = 0$ if $n_2 < n_1$, hence $c_2 = 4$.

Note that for every $3 \leq j \leq m$,

$$\log_{(j-2)}(2 \cdot b^{m-j} j \log k) \leq 2 \cdot b^{m-j} \log k \leq 2 \cdot b^{m-2} \log k.$$

Therefore,

$$c_m \leq 4 + \sum_{j=3}^{m} \frac{2 \cdot b^{m-2} \log k}{b^{m-2} R \log k} = 4 + \frac{2(m-2)}{dk^{b^{m-1}}} \leq 5.$$

■