A Theory of Online Learning with Autoregressive Chain-of-Thought Reasoning

Ilan Doron-Arad
MIT
ilanda@mit.edu

Idan Mehalel
The Hebrew University
idanmehalel@gmail.com

Elchanan Mossel
MIT
elmos@mit.edu

Abstract

Autoregressive generation lies at the heart of the mechanism of large language models. It can be viewed as the repeated application of a next-token generator: starting from an input string (prompt), the generator is applied for M steps, and the last generated token is taken as the final output. Joshi et al. [2025] proposed a PAC model for studying the learnability of the input-output maps arising from this process. We develop an online analogue of this framework, focusing on the mistake bound of learning the final output induced by an unknown next-token generator. We distinguish between two forms of feedback. In the End-to-End model, after each round the learner observes only the final token produced after M autoregressive steps. In the Chain-of-Thought model, the learner is additionally shown the entire M-step trajectory. Our goal is to understand how the optimal mistake bound depends on the generation horizon M, and to what extent observing intermediate tokens can reduce this dependence. Our main results show that the online theory of autoregressive learning exhibits a qualitative picture analogous to the statistical one found by Hanneke et al. [2026], but with a different scale of dependence on the generation horizon. In the End-to-End model, we prove a taxonomy of possible mistake-bound growth rates in the generation horizon M: subject to mild regularity conditions, every rate between constant and logarithmic can arise. We also show that this logarithmic ceiling is unavoidable for the online setting, in the sense that every class of finite Littlestone dimension exhibits at most logarithmic dependence on M. In the Chain-of-Thought model, we show that access to the full generated trajectory eliminates the dependence on M altogether, mirroring the picture proved in the statistical setting. We also analyze autoregressive linear threshold classes. For autoregressive linear thresholds in $\mathbb{R}^d$, we prove that the optimal mistake bound is $\Theta(d^2)$ under both End-to-End and Chain-of-Thought feedback, and for any generation length M. The methods developed for this analysis also yield new lower bounds in the statistical setting. Along the way, our results resolve several questions left open by Joshi et al. [2025]. In particular, we show that even for classes of finite Littlestone dimension, the End-to-End statistical sample complexity can depend on the generation length M.

1 Introduction

Modern large language models are not learned once from a fixed collection of independent examples and then used in isolation. They are trained on data accumulated over time, updated through successive rounds of optimization, deployed in interactive environments, and increasingly adapted to the particular users and contexts in which they operate. In such settings, the classical picture of learning from an i.i.d sample captures only part of the story. The learner observes a stream of

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

prompts, receives feedback, changes its behavior, and is then evaluated on future interactions. This is precisely the kind of situation for which online learning was designed.

At the same time, the objects being learned are fundamentally autoregressive. A language model does not usually produce an answer in a single atomic step. Rather, starting from a prompt, it generates one token at a time; each newly generated token becomes part of the context for the next prediction. After many such steps, the final token, sentence, or answer is interpreted as the model's output. Thus, even when the desired task is an end-to-end prediction problem, the mechanism producing the prediction is an iterated next-token process.

This creates a natural distinction between two forms of supervision. In one regime, the learner observes only the final answer produced after the autoregressive process has run for M steps. This corresponds to *end-to-end supervision*: the intermediate reasoning, if any, is hidden. In a stronger regime, the learner also observes the entire generated trajectory. This corresponds to *chain-of-thought supervision*: the learner sees not only what answer was produced, but also the sequence of intermediate tokens through which the generator arrived there. Intuitively, such intermediate information should be useful, as it exposes more of the underlying next-token generation mechanism.

A recent line of work initiated by Joshi et al. [2025] introduced a formal framework for studying this phenomenon in the PAC setting. In their model, the primitive object is a next-token generator $f : \Sigma^\star \to \Sigma$, where Σ is the alphabet of possible tokens. Iterating f for M steps from an input string produces an M-token autoregressive trajectory, whose last token is the end-to-end output. This framework makes it possible to ask, in precise learning-theoretic terms, how the difficulty of learning the induced input-output map depends on the generation length M, and how much chain-of-thought supervision can reduce this dependence. These two fundamental questions were studied in the PAC-learning framework of Joshi et al. [2025] by Hanneke et al. [2026].

The present paper develops the online analogue of this theory. Rather than receiving an i.i.d training sample, the learner interacts with an adversarially chosen sequence of examples. On each round, the learner is given a prompt, predicts the final token generated after M autoregressive steps, and then receives feedback. In the end-to-end model, the feedback consists only of the final token. In the chain-of-thought model, the feedback consists of the full M-step trajectory. The central quantity in this setting is no longer sample complexity, but the optimal mistake bound: how many prediction errors are unavoidable before the learner has identified enough of the hidden generator to perform well on all subsequent rounds?

There is also a second, more structural reason to study the online model. Online learnability is often a stronger and more robust notion than ordinary distributional learnability. A class that can be learned in the online mistake-bound model is not merely learnable from an i.i.d sample; it often remains learnable under additional constraints that make standard PAC learning more demanding. A prominent example is private PAC learning, which is equivalent to online learning, in the sense that a class is private PAC learnable if and only if it is online learnable Alon et al. [2022]. Thus, online learnability serves as a powerful certificate of learnability: it indicates that the class has enough combinatorial structure to be learned even in settings where the data are sequential, adversarial, or subject to privacy constraints. It is also valuable in cases where the i.i.d data assumption is unrealistic, and the training data is treated as an arbitrary stream of examples. We now turn to the formal definition of the model in study.

1.1 The autoregressive online learning model

We start by recalling the general autoregressive learning framework presented in Joshi et al. [2025] and defined below. A *next-token-generator* is a function

$$f : \Sigma^\star \to \Sigma,$$

where Σ is some finite alphabet. In this paper, we focus on the binary case, so $\Sigma = \{0, 1\}$ unless stated otherwise. The function f naturally induces an *autoregressive generation* process. Define the *apply-and-append* map $\bar{f} : \Sigma^\star \to \Sigma^\star$ of f by

$$\bar{f}(x) := x \circ f(x).$$

Denote

$$\bar{f}^{\,M} = \underbrace{\bar{f} \circ \cdots \circ \bar{f}}_{M \text{ times}}.$$

We define the M-*step chain of thought* of f on x to be the length M suffix of $\bar{f}^M(x)$, denoted by $f^{\mathsf{CoT}-M}(x)$. The corresponding *end-to-end output* of the M-step chain of thought of f on x is the last symbol of $f^{\mathsf{CoT}-M}(x)$, denoted as $f^{\mathsf{e2e}-M}(x) := f^{\mathsf{CoT}-M}(x)[-1]$. So, each class $\mathcal{F}$ of next-token-generators (also called a *base class*) naturally induces two classes of interest:

$$\mathcal{F}^{\mathsf{CoT}-M} := \{f^{\mathsf{CoT}-M} : f \in \mathcal{F}\}, \qquad \mathcal{F}^{\mathsf{e2e}-M} := \{f^{\mathsf{e2e}-M} : f \in \mathcal{F}\}.$$

In this paper, we are interested in learning the class $\mathcal{F}^{\mathsf{e2e}-M}$ online in two regimes: e2e-learning, also called end-to-end learning, and CoT-learning, also called end-to-end learning with chain-of-thought supervision. Let us start with the e2e-learning regime. We consider a game played between a learner Lrn and an adversary Adv played for T many rounds. Before the game begins, Adv chooses a *target function* $f_\star \in \mathcal{F}$. In each round $t \in [T]$, Adv first sends an instance $x_t \in \Sigma^\star$ to Lrn, and then Lrn chooses a symbol (or a *label*, in classification theory terminology) $\hat{y}_t \in \Sigma$. Finally, Adv reveals the correct end-to-end output $f_\star^{\mathsf{e2e}-M}(x_t)$, and Lrn suffers loss 1 if and only if $\hat{y}_t \neq f_\star^{\mathsf{e2e}-M}(x_t)$. Most of our results hold already for a deterministic learner, so unless stated otherwise, we limit our study to the case that Lrn is deterministic. We now define the CoT-learning regime. CoT-learning is very similar to e2e-learning; the only difference is that in CoT-learning, Adv reveals the entire chain-of-thought $f_\star^{\mathsf{CoT}-M}(x_t)$, instead of only $f_\star^{\mathsf{e2e}-M}(x_t)$.

In both regimes, Lrn's goal is to minimize its total cumulative loss throughout the game, also called the *mistake bound*. Adv's goal is to maximize this quantity. Note that e2e-learning amounts to standard online learning of the class $\mathcal{F}^{\mathsf{e2e}-M}$. In contrast, in CoT-learning Lrn receives $f_\star^{\mathsf{CoT}-M}(x_t)$ as feedback from Adv, but tested only on correctness of the final symbol of $f_\star^{\mathsf{CoT}-M}(x_t)$, that is, on $f_\star^{\mathsf{e2e}-M}(x_t)$.

The main goal in this paper is to understand how the mistake bound of autoregressive learning scales with the generation length M, and to what extent chain-of-thought supervision may help reducing the mistake bound. Throughout the paper, we assume that $M \geq 2$, as the case $M = 1$ is equivalent to the standard online learning scenario.

2 Results

This section assumes some familiarity with basic terms in learning theory, such as PAC-learning. The unfamiliar reader may read Section D, where we present all relevant technical background in a self-contained manner.

2.1 Background on the Littlestone dimension

We first give some relevant background on the Littlestone dimension, which is necessary to understand the results. The main combinatorial quantity that arises when analyzing mistake bounds of online learning a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ of functions from a *domain* $\mathcal{X}$ to a *label space* $\mathcal{Y}$ is the *Littlestone dimension* Littlestone [1988], Daniely et al. [2015] of $\mathcal{H}$, denoted as $\mathtt{L}(\mathcal{H})$. Let us informally define the Littlestone dimension. A formal definition is given in Section D.2. $\mathtt{L}(\mathcal{H})$ is the maximal depth of a perfect *Littlestone tree* that is shattered by $\mathcal{H}$. A perfect Littlestone tree for $\mathcal{H}$ is a rooted perfect binary tree whose internal vertices are labeled by instances from $\mathcal{X}$, and for each internal vertex, its two outgoing edges are labeled by two different labels from $\mathcal{Y}$. Such a tree is *shattered* by $\mathcal{H}$ if every branch b (root-to-leaf path) in the tree has a function $h_b \in \mathcal{H}$ that agrees with b. Here, "agrees" means that for all $x \in \mathcal{X}$ that labels an internal vertex in b, if the outgoing edge of the internal vertex labeled by x that lies in b is labeled with $y \in \mathcal{Y}$, then $f_b(x) = y$. A fundamental result of Littlestone [1988], Daniely et al. [2015] states that **the optimal mistake bound of learning $\mathcal{H}$ is precisely** $\mathtt{L}(\mathcal{H})$.

Since e2e-learning of $\mathcal{F}$ with generation length M amounts to standard online learning of the class $\mathcal{F}^{\mathsf{e2e}-M}$, our analysis focuses on bounding $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M})$. On the other hand, CoT-learning is not a standard online learning task: while the loss is measured with respect to the target function $f_\star^{\mathsf{e2e}-M} \in \mathcal{F}^{\mathsf{e2e}-M}$, the feedback is given by the target function $f_\star^{\mathsf{CoT}-M} \in \mathcal{F}^{\mathsf{CoT}-M}$. A useful approach towards proving upper bounds for CoT-learning exploited in Joshi et al. [2025], Hanneke et al. [2026] is to replace CoT-learning in the more difficult task of learning the class $\mathcal{F}^{\mathsf{CoT}-M}$. This task is indeed harder, since in contrast with CoT-learning, any disagreement between the learner's full chain-of-thought prediction $\hat{y} \in \Sigma^M$ counts as a mistake, even if the final End-to-End output

is correct. We use the same approach and derive mistake bounds for CoT-learning by bounding $\mathtt{L}(\mathcal{F}^{\mathsf{CoT}-M})$. Unless stated otherwise, we limit our discussion only for base classes with finite Littlestone dimension. This limitation is analogous to the limitation of finite VC-dimension in the statistical setting Joshi et al. [2025], Hanneke et al. [2026]. Further justification for this limitation is given in Section J, where we show that for non-Littlestone classes, learnability of $\mathcal{F}^{\mathsf{e2e}-M}$ can alternate between the two extremes of impossible even with CoT-supervision, and trivial even without CoT-supervision. A similar result was proved in Hanneke et al. [2026] for the statistical setting, for non-VC classes.

2.2 Learning with chain-of-thought supervision

The results in this section are proved in Section H Our first main result shows that having $\mathtt{L}(\mathcal{F}) < \infty$ and chain-of-thought supervision eliminates any dependence on the generation length M.

Theorem 2.1. *For every base class $\mathcal{F} \subset \Sigma^{\Sigma^\star}$ (even if $|\Sigma| > 2$, and even for $|\Sigma| = \infty$), and for every generation length M, we have*

$$\mathtt{L}(\mathcal{F}^{\mathsf{CoT}-M}) \leq \mathtt{L}(\mathcal{F}).$$

Notably, Theorem 2.1 holds even when $|\Sigma| > 2$. Hanneke et al. [2026] proved an analog result for the statistical setting: having $\mathrm{VC}(\mathcal{F}) < \infty$ (where $\mathrm{VC}(\mathcal{F})$ denoted the VC-dimension of $\mathcal{F}$) and chain-of-thought supervision eliminates any dependence of the PAC-sample complexity on the generation length M. It is not hard to think of autoregressively-degenerate classes (classes of functions that keeps generating the same bit forever) for which $\mathtt{L}(\mathcal{F}^{\mathsf{CoT}-M}) \geq \mathtt{L}(\mathcal{F})$ for all M, and so the bound of Theorem 2.1 cannot be improved in general, not even by a constant factor.

Theorem 2.1 was also proved independently and concurrently in Balcan et al. [2026].

Theorem 2.1 can be used to improve, for some classes, the PAC CoT-learning sample complexity bound given by Hanneke et al. [2026]. Their result shows that the PAC sample complexity of CoT-learning is proportional to $\mathrm{VC}(\mathcal{F}) \cdot \mathrm{VC}^\star(\mathcal{F})$, where $\mathrm{VC}^\star(\mathcal{F}) = \mathrm{VC}(\mathcal{F}^\star)$, where $\mathcal{F}^\star$ is the dual class of $\mathcal{F}$, which is the class obtained by "switching roles" between $\mathcal{X}$ and $\mathcal{F}$. However, $\mathrm{VC}^\star(\mathcal{F})$ can grow as $2^{\mathrm{VC}(\mathcal{F})}$. Using a standard online-to-batch conversion, we deduce the following sample complexity bound. Notably, it holds even when $|\Sigma| > 2$.

Theorem 2.2. *For every base class $\mathcal{F} \subset \Sigma^{\Sigma^\star}$ (even if $|\Sigma| > 2$, and even for $|\Sigma| = \infty$), and for all M, if $\mathtt{L}(\mathcal{F}) < \infty$ the class $\mathcal{F}^{\mathsf{CoT}-M}$ is PAC-learnable with sample complexity*

$$m(\varepsilon, \delta) = O\left(\frac{\mathtt{L}(\mathcal{F}) + \log(1/\delta)}{\varepsilon}\right).$$

So, the advantage of Theorem 2.2 over the result stated in Hanneke et al. [2026] is that it holds for non-binary base classes $\mathcal{F}$, and also that it provides a better bound for every binary class $\mathcal{F}$ for which $\mathtt{L}(\mathcal{F}) < \mathrm{VC}(\mathcal{F})\,\mathrm{VC}^\star(\mathcal{F})$, such as d-hamming balls.

Along the way, Theorem 2.2 also settles a $\log M$ gap from Joshi et al. [2025], who proved essentially the same bound as of Theorem 2.2, but with $\mathtt{L}(\mathcal{F}) \log M$ replacing the $\mathtt{L}(\mathcal{F})$ in our bound. They asked if the logarithmic dependence on M is necessary, and our bound shows that it is not.

We note here that Theorem 2.2 may also be deduced via *stable sample compression schemes*, using a result of Hanneke et al. [2026]. Concretely, they proved that if $\mathcal{F}$ has a stable sample compression scheme of size k, then $\mathcal{F}^{\mathsf{CoT}-M}$ also has a stable sample compression scheme of size k. Since every class $\mathcal{F}$ has a sample compression scheme of size $O(\mathtt{L}(\mathcal{F}))$ Nikita [2017], Theorem 2.2 is implied.

2.3 End-to-End learning

We first establish a general upper bound that bounds how $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M})$ can scale with M in the worst case.

Theorem 2.3. *For every base class $\mathcal{F}$, and for every generation length M, we have*

$$\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = O(\mathtt{L}(\mathcal{F}) \log(\mathtt{L}(\mathcal{F})M)).$$

Theorem 2.3 is proved in Section E. While it gives a strong guarantee for any base class $\mathcal{F}$ (of finite Littlestone dimension), unlike the guarantee of Theorem 2.1, it is not clear how tight Theorem 2.3

is. Indeed, it is not hard to think of classes $\mathcal{F}$ for which $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) \leq \mathtt{L}(\mathcal{F})$ for all M, and it is not immediately clear if there are classes for which $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M})$ scales as $\mathtt{L}(\mathcal{F}) \log M$, or even as some sub-logarithmic monotone function of M. This naturally raises the following question:

What are the possible growth rates of $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M})$ in terms of $\mathtt{L}(\mathcal{F})$ and M?

This question was considered by Hanneke et al. [2026] in the statistical setting. They proved that under mild regularity conditions, essentially any at-most-linear growth rate of $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M})$ with M is realized by some class $\mathcal{F}$. We prove the analogous result for the online setting, with similar mild regularity conditions to those of Hanneke et al. [2026]. Let us explicitly state those conditions. Fix a sufficiently large universal constant M_0. We say that a function $r : \mathbb{N}_+ \to \mathbb{N}_+$ is a *well-behaved sub-logarithmic growth rate* if:

1. r is sub-logarithmic: $r(M) \leq \frac{1}{4}\log(M)$ for all $M \geq M_0$.
2. r is non-decreasing : $M_1 \geq M_2 \implies r(M_1) \geq r(M_2)$ for all $M_1, M_2 \geq 1$.
3. r is sub-additive: $r(M_1 + M_2) \leq r(M_1) + r(M_2)$ for all $M_1, M_2 \geq 1$.

Note that all natural sub-logarithmic growth rates such as $\lceil\sqrt{\log M}\rceil$, $\lceil\log\log M\rceil$ etc. satisfy our definition. We now state the result for well-behaved sub-logarithmic rates.

Theorem 2.4. *For any well-behaved sub-logarithmic growth rate r there exists a base class $\mathcal{F}$ such that:*

1. $\mathtt{L}(\mathcal{F}) = 1$.
2. $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = \Theta(r(M))$ *for all* $M \geq M_0$.

The definition of $\mathcal{F}$ and the lower bound strategy of the second item are due to unpublished private communication Hanneke and Moran [2026].

Theorem 2.4 handles only sub-logarithmic rates. For logarithmic rates, we prove the following

Theorem 2.5. *For any $d \in \mathbb{N}_+$, there exists a class $\mathcal{F}$ such that $\mathtt{L}(\mathcal{F}) = O(d)$ and*

$$\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = \Omega(\mathtt{L}(\mathcal{F}) \log M)$$

for all sufficiently large M.

Theorem 2.4 and Theorem 2.5 are proved in Section F. Since $\mathrm{VC}(\mathcal{F}) \leq \mathtt{L}(\mathcal{F})$ for all $\mathcal{F}$, together with the upper bound of Theorem E.1, this establishes that for any $d \in \mathbb{N}_+$ there exists a class $\mathcal{F}$ such that $\mathtt{L}(\mathcal{F}) = O(d)$ and

$$\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}), \mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = \Theta(\mathtt{L}(\mathcal{F}) \log M)$$

for all sufficiently large M.

So, Theorem 2.4, Theorem 2.5 and Theorem 2.3 establish an essentially complete taxonomy of the possible growth rates of $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M})$ in terms of $\mathtt{L}(\mathcal{F})$ and M.

Indeed, even though Theorem 2.4 relates only to base classes with $\mathtt{L}(\mathcal{F}) = 1$, the taxonomy is still essentially complete when considering the asymptotics. Concretely, suppose that r_1, r_2 are two rates, such that r_1 is asymptotically larger than r_2. So, for any fixed d, r_1 is also asymptotically larger than $d \cdot r_2$, and r_1 can be established via a class $\mathcal{F}$ with $\mathtt{L}(\mathcal{F}) = 1$ by Theorem 2.4. Put differently, $d \cdot r_2$ is sandwiched between r_2 and r_1, and r_1, r_2 can have an arbitrarily small asymptotic difference.

Along the way, Theorem 2.5 resolves an open question on the statistical setting posed by Joshi et al. [2025]: they proved that for any class $\mathcal{F}$ it holds that $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = O(\mathtt{L}(\mathcal{F}) \log M)$, and asked if the logarithmic dependence in their bound can be improved. Theorem 2.5 shows that their bound is tight.

Theorem 2.3 implies that finite $\mathtt{L}(\mathcal{F})$ guarantees that $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M})$ grows at most logarithmically with M, and Theorem F.2 states that this is tight in the worst case. One may further ask if there exists a dimension that guarantees strictly $o(\log M)$ growth rate. In the paper Hanneke et al. [2026], the

analog result of Theorem 2.4 for the statistical setting is used to prove that no dimension characterizes sublinear growth rates. We note here that Theorem 2.4 allows one to apply essentially the same argument of Hanneke et al. [2026] to prove that in the online setting, no dimension characterizes sub-logarithmic rates.

2.3.1 Corollary for agnostic learning

Much of the online learning literature studies *agnostic online learning*, where there is no target function in the class that perfectly explains the streamed data. In this setting, instead of measuring the mistake bound, we are usually interested in the *regret bound*, which is the optimal mistake bound measured after subtracting the number of mistakes made by the best function in class. A known result of Alon et al. [2021] allow to derive a similar taxonomy for agnostic end-to-end learning.

Corollary 2.6. *Let $\mathcal{F}$ be a base class. Then for all sufficiently large M, there exists an agnostic online learner for $\mathcal{F}^{\mathsf{e2e}-M}$ with regret bound*

$$O\Big(\sqrt{T\mathtt{L}(\mathcal{F})\log M}\Big).$$

Furthermore, for any well-behaved sub-logarithmic growth rate r there exists a base class $\mathcal{F}$ such that $\mathtt{L}(\mathcal{F}) = 1$ and the optimal regret bound for agnostic online learning $\mathcal{F}^{\mathsf{e2e}-M}$ is

$$\Theta\Big(\sqrt{T \cdot r(M)}\Big)$$

for all $M \geq M_0$ and sufficiently large T.

Finally, for any $d \in \mathbb{N}_+$, there exists a class $\mathcal{F}$ such that the optimal regret for agnostic online learning of $\mathcal{F}^{\mathsf{e2e}-M}$ is

$$\Theta\Big(\sqrt{Td\log M}\Big)$$

for all $M \geq M_0$ and sufficiently large T.

Corollary 2.6 is proved in Section G.

2.4 Results for autoregressive linear classifiers

An interesting specific study case is the basic class of *autoregressive linear classifiers*. This class was presented in Joshi et al. [2025] and further studied in Hanneke et al. [2026], Joshi et al. [2026]. It is naturally defined as follows. Let $d \in \mathbb{N}$ be the dimension. For any $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ define the function $f_{\boldsymbol{w},b} : \{0,1\}^\star \to \{0,1\}$ by

$$f_{\boldsymbol{w},b}(x) = 1\left[\sum_{i=1}^{\min\{d,|\boldsymbol{x}|\}} \boldsymbol{w}[-i]x[-i] + b \geq 0\right]$$

for all $x \in \{0,1\}^\star$. That is, $f_{\boldsymbol{w},b}$ acts as the linear classifier defined by $(\boldsymbol{w}, b)$ when taking only values on the d-boolean cube, and the input point to the classifier is given by the (at most) final d bits of the input x. The class $\mathcal{F}_{d,\mathrm{lin}}$ of autoregressive linear classifiers in dimension d is defined as

$$\mathcal{F}_{d,\mathrm{lin}} := \{f_{\boldsymbol{w},b} : \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}\}.$$

2.4.1 Mistake and sample complexity bounds

The results in this section are proved in Section I.1. Joshi et al. [2025] proved that $\mathrm{VC}(\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e},M}) = O(d^2)$ for all M. This implies that the dependence of the sample complexity of CoT-learning $\mathcal{F}_{d,\mathrm{lin}}$ on d is also at most $O(d^2)$. Hanneke et al. [2026] improved the dependence on d for CoT-learning from $O(d^2)$ to $O(d)$. No lower bounds (that hold for any $M > 1$) were previously given.

In this work, we prove that the optimal mistake bound for online learning $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}}$, both with and without CoT supervision, is $\Theta(d^2)$.

Theorem 2.7. *For any dimension d and generation length M we have*

$$\mathtt{L}(\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}) = O(d^2).$$

Furthermore, for any M and any $d \geq 3$, there exists an adversary forcing $\Omega(d^2)$ many mistakes on any learner for $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$, even with CoT*-supervision.*

The technique we use for the lower bound of Theorem 2.7 allows us to derive the first lower bound for the statistical setting that holds for $M > 1$.

Theorem 2.8. *For any dimension $d \geq 3$ and generation length $M \geq 2$, any PAC-learner for $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$ has sample complexity $\Omega(d)$, even with* CoT*-supervision.*

Theorem 2.8 shows that the upper bound of Hanneke et al. [2026] for PAC-learning $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}}$ with CoT-supervision is tight.

2.4.2 Computational complexity

The results in this section are proved in Section I.2. The preceding bounds are information-theoretic. We also obtain a computational separation between e2e and CoT-learning for linear classifiers.

Theorem 2.9. *Assume Hardness of Learning TC^0.*[1] *Then there is no possibly improper online learner for $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$ that runs in time polynomial in d, M and the input length per round and makes polynomially many mistakes. In contrast, with* CoT*-supervision, there is a deterministic polynomial-time learner for $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$ with mistake bound $O(d^2 \log d)$.*

Thus, CoT-supervision removes the computational obstruction, at least up to a logarithmic factor in the mistake bound.

2.5 Non-Littlestone classes

The finite Littlestone assumption is essential. For $\mathtt{L}(\mathcal{F}) < \infty$, our upper bound gives a uniform logarithmic ceiling in M. Without this assumption, a single base class can alternate between trivial and unlearnable horizons.

Theorem 2.10. *There exists a binary base class $\mathcal{F}$, such that for every $M_i = i + 4$:*

$$i \text{ odd} \implies \mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M_i}) = 0, \qquad i \text{ even} \implies \mathcal{F}^{\mathsf{e2e}-M_i} \text{ is unlearnable, even with CoT-supervision.}$$

That is, the odd horizons are trivial already in the e2e *model, while the even horizons are unlearnable even with* CoT*-supervision.*

Thus, outside the Littlestone regime, there are cases where there is no stable growth law in M. Theorem 2.10 is proved in Section J

2.6 A stochastic autoregressive lower bound

We also show that the deterministic picture of online learning does not extend to stochastic autoregressive generators, at least in the case of no CoT-supervision. In this model, a generator maps each string in Σ^* to a distribution over the next token, so even after the input x_t and target generator are fixed, the CoT trajectory is random. For the stochastic setting, the natural benchmark is regret.

Theorem 2.11. *For every odd $M \geq 1$, there exists a stochastic two-function base class $\mathcal{G} = \{g_-, g_+\}$ whose direct next-token problem has regret $O(1)$, but for every online learner, there exists $g_\star \in \mathcal{G}$ such that learning $\mathcal{G}^{\mathsf{e2e}-M}$ against target $g_\star$ incurs expected regret $\Omega(2^M)$ for horizon $T = \Theta(4^M)$.*

Thus, stochastic autoregression can exponentially amplify the difficulty of an otherwise trivial next-token learning problem. This is in stark contrast to the upper bound for deterministic online learning given in Theorem 2.1 and Theorem 2.3. Theorem 2.11 is proved in Section K.

[1] This assumption is formally given in Assumption I.5; for more details, the reader is referred to, e.g., Joshi et al. [2025].

3 Proof sketches of the main results

In this section, we explain the main technical ideas of the proofs of our main theorems. The proof sketches of our results for linear classifiers are deferred to the appendix due to lack of space.

3.1 Proof sketch of Theorem 2.1: mistake bound for learning with CoT

Theorem 2.1 proves that with CoT-supervision, learning $\mathcal{F}^{\mathsf{e2e}-M}$ is possible with mistake bound at most $\mathtt{L}(\mathcal{F})$, for all M. The proof follows from a relatively simple SOA-style[2] argument we describe below. We design an online learner Lrn for the class $\mathcal{F}^{\mathsf{CoT}-M}$, which operates as follows. Lrn maintains a *version space* V_t of all generators $f \in \mathcal{F}$ that are consistent with the feedback given up to the end of round t: at first $V_0 = \mathcal{F}$, and at the end of any round t, Lrn sets V_t to be all functions in V_{t-1} that are consistent with the feedback of round t. In each round t, Lrn receives x_t and predicts

$$\max_{\hat{y}\in\{0,1\}^M} \mathtt{L}(V_{t-1}|_{x_t\to\hat{y}}),$$

where $V_{t-1}|_{x_t\to\hat{y}}$ denotes the restriction of V_{t-1} only to functions for which autoregressive chain-of-thought on x_t is precisely $\hat{y}$. The crucial observation is that there is at most one $\hat{y} \in \{0,1\}^M$ such that

$$\mathtt{L}(V_{t-1}|_{x_t\to\hat{y}}) = \mathtt{L}(V_{t-1}).$$

Indeed, if there are two distinct $\hat{y}, \hat{y'}$ such that

$$\mathtt{L}(V_{t-1}|_{x_t\to\hat{y}}) = \mathtt{L}(V_{t-1}|_{x_t\to\hat{y'}}) = \mathtt{L}(V_{t-1})$$

then one can construct a Littlestone tree of depth $\mathtt{L}(V_{t-1})+1$ which is shattered by V_{t-1}, which is a contradiction to the fact that $\mathtt{L}(V_{t-1})$ is the optimal mistake bound for learning V_{t-1}, as proved by Daniely et al. [2015].

Therefore, after every mistake of Lrn the Littlestone dimension of the version space decreases by at least 1, which implies an upper bound of $\mathtt{L}(V_0) = \mathtt{L}(\mathcal{F})$.

3.2 Proof sketch of Theorem 2.3: mistake bound for learning without CoT

Consider a Littlestone tree $\mathbf{T}$ of depth m that is shattered by $\mathcal{F}^{\mathsf{e2e}-M}$. It suffices to show that $m = O(\mathtt{L}(\mathcal{F})\log(\mathtt{L}(\mathcal{F})M))$. The idea is a variant of tree-covering arguments occasionally used in online learning (see Filmus et al. [2023] for example). First, we inflate $\mathbf{T}$ to represent the entire chain-of-thought for every instance labeling a node in the tree. We do so by replacing each node and its two outgoing edges by the the full M-step *generation tree* matching the instance labeling the node. An M-step generation tree for $x \in \{0,1\}^\star$, denoted by $\mathbf{T}_M(x)$, is a Littlestone tree of depth M with the root labeled by x, and for any prefix $u \in \{0,1\}^{<M}$, the node at the end of the prefix u is labeled by $x \circ u$. So the branches of $\mathbf{T}_M(x)$ represent the different possible chain-of-thoughts for x. Let $\mathbf{T}_\mathcal{F}$ be the inflated tree, which is now of depth mM. We now analyze the size of the set of branches in $\mathbf{T}_\mathcal{F}$ that are realized by some $f \in \mathcal{F}$, denoted as $B_\mathcal{F}(\mathbf{T}_\mathcal{F})$. Since every generation tree in $\mathbf{T}_\mathcal{F}$ is originated from a node in $\mathbf{T}$ and its two outgoing edges, and since $\mathbf{T}$ is shatterd by $\mathcal{F}^{\mathsf{e2e}-M}$, it is not hard to see that $B_\mathcal{F}(\mathbf{T}_\mathcal{F})$ grows by a factor of at least 2 after each M levels of $\mathbf{T}_\mathcal{F}$, which implies

$$|B_\mathcal{F}(\mathbf{T}_\mathcal{F})| \geq 2^m.$$

On the other hand, a known (for example, by Bhaskar [2021]) Sauer-Shelah-Perles lemma for trees give

$$|B_\mathcal{F}(\mathbf{T}_\mathcal{F})| \leq \binom{mM}{\leq \mathtt{L}(\mathcal{F})}.$$

By the two inequalities, we obtain

$$2^m \leq \binom{mM}{\leq \mathtt{L}(\mathcal{F})}.$$

Solving the inequality for m gives the stated bound.

[2]The *SOA* stands for *standard optimal algorithm*, which is the original algorithm given in Littlestone [1988] for online learning of binary classes.

3.3 Proof sketch of Theorem 2.4: taxonomy for sub-logarithmic rates

To prove Theorem 2.4, we construct a class $\mathcal{F}$ suggested in private communication Hanneke and Moran [2026] that has two desired properties:

1. It is simple, and in fact has $\mathtt{L}(\mathcal{F}) = 1$.
2. We can make $\mathcal{F}^{\mathsf{e2e}-M}$ as complicated as we want up to some well-behaved sub-logarithmic rate $r(M)$. In fact, given any well-behaved sub-logarithmic rate r, we have $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = \Theta(r(M))$.

To implement this, we define a fixed baseline function f, and $\mathcal{F}$ is constructed from a class of functions such that each function differs from f on precisely one instance. This fact alone suffices to prove the first item. To prove the second item, the baseline function f, as well as the other functions in the class, should be carefully chosen to fit the given well-behaved sub-logarithmic rate r. The proof, and even the definition of $\mathcal{F}$, are quite technical. Below, we explain the high level idea of the definition and proof. The function f is defined such that it is "interesting" only on instances of the form

$$x_{s,k,z} := 0^s 1 0^k 1 0^{s-k-1} z$$

where s can be any natural number, $k \in \left[2^{r(s)}\right]$, and $z \in \{0,1\}^{<r(s)}$. Leaving some technical details aside, $f(x_{s,k,z}) = k_{|z|}$, so $f(x_{s,k,z})$ in fact returns a bit of the binary representation of k, in a location encoded by the length of z. For any other instance x, define $f(x) = 0$. Note that assuming that r is sub-logarithmic is crucial already here: since k can be large as $2^{r(s)}$, assuming that r is sub-logarithmic ensures that $s - k - 1 \geq 1$, and so 0^{s-k-1}, which appears in $s_{s,k,z}$, is a well-defined binary string. Now, for every s, k as above we define a function $f_{s,k}$. This function is defined exactly as f everywhere except for the instance $0^s 1 0^k$, for which f returns 0, but we define $f_{s,k}(0^s 1 0^k) = 1$. The class $\mathcal{F}$ is defined as the class of all $f_{s,k}$. The definition of $f_{s,k}$ is one of the core ideas of the proof: when the instance $0^s 1 0^k$ is given to $f_{s,k}$ and only to $f_{s,k}$, the s-step autoregressive generation of $f_{s,k}$ for $0^s 1 0^k$ is non-degenerate, and in fact encodes the binary representation of $k \in \left[2^{r(s)}\right]$. This is the high-level idea of the lower bound: for $s := M$, any learner makes at least roughly $\log 2^{r(M)} = r(M)$ many mistakes before identifying the "correct" k, where the logarithmic lower bound comes from optimality of the halving algorithm in this case.

For the upper bound, the main idea is to design a learner for which the halving algorithm used to find the correct k is essentially the only obstacle. Let us describe this learner in high level. Until making its first mistake, the learner always predicts as $f^{\mathsf{e2e}-M}$. Once the learner makes its first mistake on some instance x, it knows the following fact:

$$f^{\mathsf{e2e}-M}_{s^\star,k^\star}(x) \neq f^{\mathsf{e2e}-M}(x),$$

where $f_{s^\star,k^\star}$ is the target function. Since $f_{s^\star,k^\star}$ differs from f only on the instance $0^{s^\star}1 0^{k^\star}$, this implies that x must be of the form $0^{s^\star}1 0^i$ for some $i \in \mathbb{N}$. Therefore, the learner now knows $s^\star$ just by looking on x, and so the learner now only needs to find $k^\star$ to know the target function $f_{s^\star,k^\star}$. If $s^\star = O(M)$, then $k^\star$ can only receive as few as

$$2^{r(s^\star)} \leq 2^{r(O(M))} \leq 2^{O(r(M))} \tag{1}$$

many values, and thus the halving algorithm finds $k^\star$ after at most $O(r(M))$ many mistakes, which is analogous to the argument of the lower bound. Note that (1) does not hold for a general rate r. The first inequality holds since r is non-decreasing, and the second inequality holds since r is sub-additive. This is the exact reason that we make those two assumptions. However, it is also possible that $s^\star \gg M$. In this case, we need to use a completely different argument. In the case that $s^\star \gg M$, the learner acts as follows. For any x not of the form $0^{s^\star}1 0^i$ it predicts as f and never errs. For x which is of the form $0^{s^\star}1 0^i$, the learner always predicts 0 until making its first mistake on such an instance, which implies that

$$f^{\mathsf{e2e}-M}_{s^\star,k^\star}(0^{s^\star}1 0^i) = 1, \tag{2}$$

where $0^{s^\star}1 0^i$ is the instance the learner errs on. However, recalling the definition of f, it is non-degenerate only for instances of the form $0^s 1 0^k 1 0^{s-k-1} z$, which in particular means that in our case $s^\star - k - 1$ zeroes must be appended before getting to the interesting part. However, since $s^\star \gg M$, we have that $s^\star - k - 1 > M$, and thus M is simply not large enough to even get to the interesting part. So, in fact, the case (2) can essentially never happen (leaving aside some technical details).

3.4 Proof sketch of Theorem 2.5: logarithmic lower bound

Theorem 2.5 states that for any (large enough) d there exists a base class $\mathcal{F}$ with $\mathtt{L}(\mathcal{F}) = O(d)$, and such that $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = \Omega(d \log M)$ for all (large enough) M. The idea is to construct for every d, M a base class $\mathcal{F} := \mathcal{F}(d, M)$ of size $\approx M^d$ which is very unbalanced for every instance x (that is, the functions of $\mathcal{F}$ demonstrate an overwhelming majority towards one of the possible labels for every x), but after M generation steps they become much more balanced for many instances. This is precisely what makes $\mathtt{L}(\mathcal{F}) = O(d)$, while $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = \Omega(d \log M)$. In more detail, our specific choice of overwhelming majority is of a $\approx (1 - 1/M)$-fraction of functions for all x. This indeed gives $\mathtt{L}(\mathcal{F}) = O(d)$, since we start from $\approx M^d$ many functions, and for each restriction of the form $\mathcal{F}_{x \to r} := \{f \in \mathcal{F} : f(x) = r\}$ for some $r \in \{0, 1\}$, the number of surviving functions decreases by a factor of M. Therefore, after at most roughly d such restrictions (that correspond to an optimal learner's mistakes), there are no surviving functions left. On the other hand, we ensure that after M many generation steps, there is a constant fraction of functions realizing each label for x, for many instances x. Since for every restriction of an instance to be classified with some label after M generation steps a constant fraction of the class stays consistent, it is implied that

$$\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = \Omega(\log(M^d)) = \Omega(d \log M).$$

It remains to construct a class having the above desired overwhelming majority property for all x. However, ensuring this desired property simultaneously for all x seems difficult to do deterministically. We thus define the class at random: every f gives to every x (in a specific desired set of instances) $r \in \{0, 1\}$ with probability $1 - 1/M$, and $1 - r$ with probability $1/M$. Figure 2 demonstrates why $\mathcal{F}^{\mathsf{e2e}-M}_{x \to 0}$ and $\mathcal{F}^{\mathsf{e2e}-M}_{x \to 1}$ both contain a constant fraction of $\mathcal{F}$.

In this intuitive explanation, one of the important facts we overlooked is that $\mathcal{F} := \mathcal{F}(d, M)$, that is, it depends on the generation length M, while we want the result to hold for all M. This is resolved by "gluing" together $\{\mathcal{F}(d, M)\}_M$ for all M values, a technique used for example in Chase and Mehalel [2024].

Acknowledgments and Disclosure of Funding

We thank Steve Hanneke and Shay Moran for their significant contribution to this work, and concretely for suggesting the class used in the proof of Theorem 2.4, as well as the strategy of the lower bound.

We also thank anonymous NeurIPS 2026 reviewers for their valuable suggestions, which we think have helped to improve the paper. In particular, one of the reviewers noticed that Theorem 2.2 can alternatively be proved via a result of Hanneke et al. [2026].

Idan Mehalel is supported by the European Research Council (ERC) under the European Union's Horizon 2022 research and innovation program (grant agreement No. 101041711), the Israel Science Foundation (grant number 2258/19), and the Simons Foundation (as part of the Collaboration on the Mathematical and Scientific Foundations of Deep Learning).

Ilan Doron-Arad is supported by grant NSF DMS-2031883 and Vannevar Bush Faculty Fellowship ONR-N00014-20-1-2826 (PI Mossel).

Elchanan Mossel is partially supported by NSF DMS-2031883, Vannevar Bush Faculty Fellowship ONR-N00014-20-1-2826, MURI N000142412742, and a Simons Investigator Award.

References

Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. Adversarial laws of large numbers and optimal regret in online classification. In *Proceedings of the 53rd annual ACM SIGACT symposium on theory of computing*, pages 447–455, 2021.

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

Awni Altabaa, Omar Montasser, and John Lafferty. Cot information: Improved sample complexity under chain-of-thought supervision. *arXiv preprint arXiv:2505.15927*, 2025.

Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180, 2010.

Benny Applebaum, Dung Bui, Geoffroy Couteau, and Nikolas Melissaris. Structured-seed local pseudorandom generators and their applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025)*, volume 353, pages 63–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025.

Maria-Florina Balcan, Avrim Blum, Kiriaki Fragkia, Zhiyuan Li, and Dravyansh Sharma. Online learnability of chain-of-thought verifiers: Soundness and completeness trade-offs. *arXiv preprint arXiv:2603.03538*, 2026.

Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, 2009.

Siddharth Bhaskar. Thicket density. *The Journal of Symbolic Logic*, 86(1):110–127, 2021.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Zachary Chase and Idan Mehalel. Deterministic apple tasting. *arXiv preprint arXiv:2410.10404*, 2024.

Lijie Chen, Binghui Peng, and Hongxun Wu. Theoretical limitations of multi-layer transformer. In *2025 IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2631–2653. IEEE, 2025.

Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf's. In *Conference on Learning Theory*, pages 815–830. PMLR, 2016.

Amit Daniely and Gal Vardi. From local pseudorandom generators to hardness of learning. In *Conference on Learning Theory*, pages 1358–1394. PMLR, 2021.

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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL `https://doi.org/10.48550/arXiv.2501.12948`.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL `https://doi.org/10.48550/arXiv.2407.21783`.

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

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory*, pages 1000–1019. PMLR, 2014.

Yuval Filmus, Steve Hanneke, Idan Mehalel, and Shay Moran. Optimal prediction using expert advice and randomized littlestone dimension. In *COLT*, volume 195 of *Proceedings of Machine Learning Research*, pages 773–836. PMLR, 2023.

Steve Hanneke. The optimal sample complexity of pac learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016.

Steve Hanneke and Shay Moran. Private communication. Unpublished, 2026. Private communication with the authors.

Steve Hanneke, Idan Mehalel, and Shay Moran. Sample complexity of autoregressive reasoning: Chain-of-thought vs. end-to-end. *arXiv preprint arXiv:2604.12013*, 2026. doi: 10.48550/arXiv.2604.12013.

Yu Huang, Zixin Wen, Aarti Singh, Yuejie Chi, and Yuxin Chen. Transformers provably learn chain-of-thought reasoning with length generalization. *arXiv preprint arXiv:2511.07378*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, and Ilge Akkaya. Openai o1 system card. *CoRR*, abs/2412.16720, 2024. doi: 10.48550/ARXIV.2412.16720.

Nirmit Joshi, Gal Vardi, Adam Block, Surbhi Goel, Zhiyuan Li, Theodor Misiakiewicz, and Nathan Srebro. A theory of learning with autoregressive chain of thought. *arXiv preprint arXiv:2503.07932*, 2025.

Nirmit Joshi, Roey Magen, Nathan Srebro, Nikolaos Tsilivis, and Gal Vardi. Learning to think from multiple thinkers. *arXiv preprint arXiv:2604.24737*, 2026.

Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought. *arXiv preprint arXiv:2410.08633*, 2024.

Adam R Klivans and Alexander A Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2–12, 2009.

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

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*, 2024.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

Nick Littlestone. From on-line to batch learning. In *Proceedings of the Second Annual Workshop on Computational Learning Theory (COLT 1989)*, pages 269–284, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc. doi: 10.5555/93335.93365. URL `https://dl.acm.org/doi/10.5555/93335.93365`.

Wolfgang Maass and György Turán. How fast can a threshold gate learn? In *Computational Learning Theory and Natural Learning Systems, Vol. I*, pages 381–414. MIT Press, Cambridge, MA, 1994.

Eran Malach. Auto-regressive next-token predictors are universal learners. *arXiv preprint arXiv:2309.06979*, 2023.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. *arXiv preprint arXiv:2310.07923*, 2023.

Zhivotovskiy Nikita. Optimal learning via local entropies and sample compression. In *Conference on Learning Theory*, pages 2023–2065. PMLR, 2017.

Maxwell I. Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models. *CoRR*, abs/2112.00114, 2021. URL `https://arxiv.org/abs/2112.00114`.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019.

Rocco A Servedio. The probably approximately correct learning model in computational learning theory. *arXiv preprint arXiv:2511.08791*, 2025.

Stefan Tiegel. Improved hardness results for learning intersections of halfspaces. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 4764–4786. PMLR, 2024.

Nikolaos Tsilivis, Eran Malach, Karen Ullrich, and Julia Kempe. How reinforcement learning after next-token prediction facilitates learning. *arXiv preprint arXiv:2510.11495*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022b.

Appendix Table of Contents

A Proof sketches of Theorem 2.7 and Theorem 2.8: bounds for linear classifiers **16**

B Future work **16**

C Additional related work **17**

D Technical preliminaries **17**

D.1 Littlestone trees 17

D.2 Class dimensions, learnability, and an SSP lemma 18

E Upper bound for end-to-end learning **20**

F The taxonomy of end-to-end learning **20**

F.1 Proof of Theorem F.1 21

F.2 Proof of Theorem F.2 24

G Agnostic taxonomy **30**

H Learning with chain-of-thought supervision **31**

I Autoregressive Linear classifiers **31**

I.1 Mistake and sample complexity bounds 32

I.2 Bounds for Efficient learning 33

J Non-Littlestone base classes **35**

K Stochastic Autoregressive Lower Bound **37**

A Proof sketches of Theorem 2.7 and Theorem 2.8: bounds for linear classifiers

The interesting part of our bounds for autoregressive linear classifiers is the following "latch" mechanism allowing to "drag" the output of the base classifier throughout the entire autoregressive process in the price of reducing the dimension by 2:

Lemma A.1 (Latch lemma)**.** *Fix $d \geq 3$ and denote $m := d - 2$. Then, for all $\boldsymbol{v} \in \mathbb{R}^m$ and $c \in \mathbb{R}$, there exists $f_{\boldsymbol{w},b} \in \mathcal{F}_{d,\mathrm{lin}}$ such that*

$$f_{\boldsymbol{w},b}^{\mathsf{e2e}-M}(z10) = f_{\boldsymbol{v},c}(z)$$

for all $z \in \{0,1\}^m$ and all $M \geq 2$.

Once we have Lemma A.1, the rest of the arguments in the proofs of Theorem 2.7 and Theorem 2.8 are either simple or known. Indeed, the upper bounds for autoregressive linear classifiers are not larger than the standard upper bounds for (standard) linear classifiers. Therefore, it essentially suffices to show that the autoregressive process cannot make the class simpler, and the class maintains its original mistake bound/sample complexity.

The idea of the proof of Lemma A.1 is quite simple. We define $\boldsymbol{w}$ to be as $\boldsymbol{v}$, except for the final two locations of $\boldsymbol{w}$, that we are free to choose. We also choose b to be as c, but shifted in some value. When choosing the above values carefully, we are able to ensure that:

1. For all $z \in \{0,1\}^m$: $f_{\boldsymbol{w},b}(z10) = f_{\boldsymbol{v},c}(z)$ (first bit of CoT is correct).
2. For any $x \in \{0,1\}^\star$ that ends with 00, we have $f_{\boldsymbol{w},b}(x) = 0$ (always drag 0).
3. For any $x \in \{0,1\}^\star$ that ends with 1, we have $f_{\boldsymbol{w},b}(x) = 1$ (always drag 1).

The above three items essentially establish the statement of the lemma. The idea is to use known lower bounds for dimension m, but add the suffix 10 to every instance used in the proof, so the dimension becomes $d := m + 2$. Now, the first item ensures that for every $z \in \{0,1\}^m$, $f_{\boldsymbol{w},b}(z10)$ outputs the correct intended output $f_{\boldsymbol{v},c}(z)$. The other two items then establish that this correct output is being dragged throughout the entire autoregressive generation process: if $f_{\boldsymbol{w},b}(z10) = 0$ then the generated new instance must end at 00, and if $f_{\boldsymbol{w},b}(z10) = 1$ it must end with 1. Thus, any lower bound strategy for $\mathcal{F}_{m,\mathrm{lin}}$ works also for $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$ for all M, even with CoT-supervision.

B Future work

Our work leaves some natural and interesting directions for future work. Below, we describe some of them.

Extend the results to the multiclass setting. Some of our results extend to the multiclass setting $|\Sigma| > 2$ "for free", such as the upper bound for CoT-learning (Theorem 2.1). This is not true, however, for all results in this paper. For example, our initial attempts to extend the upper bound for $\mathsf{e2e}$-learning (Theorem 2.2) to the multiclass setting demonstrated an extra $\log|\Sigma|$ factor, which we do not know if necessary, even for some classes.

Perhaps an even more interesting direction in the multiclass setting is to study the optimal mistake bounds with and without CoT-supervision attainable with different models of partial feedback, such as bandit feedback and feedback graphs.

The stochastic setting. Theorem 2.11 shows that in the stochastic setting, the regret bound can grow exponentially with M even for a class of size two. Further characterizing the online and PAC learnability of the stochastic setting is a natural direction for a follow-up work.

Studying specific classes. A natural direction is to move beyond worst-case base classes and analyze standard finite-Littlestone classes directly, as we did by studying autoregressive linear threshold functions. Other concrete examples for future work include Boolean classes such as conjunctions, disjunctions, decision lists, and bounded-depth decision trees, as well as finite-domain geometric classes such as intervals and unions of intervals. For such structured classes, one may hope for sharper

mistake bounds than the general $O(\mathtt{L}(\mathcal{F})\log(\mathtt{L}(\mathcal{F})M))$ guarantee, or for class-specific separations between e2e-learning and CoT-learning.

Finer characterization of CoT-learning Our results for CoT-learning are established via learning the full-trajectory class $\mathcal{F}^{\mathsf{CoT}-M}$. It is worth noting that this is in fact a more difficult task than what is actually required, which is learning the class $\mathcal{F}^{\mathsf{e2e}-M}$ with CoT-supervision. It will be interesting to develop tools that strictly improve the bounds obtained by analyzing the Littlestone dimension of $\mathcal{F}^{\mathsf{CoT}-M}$. Such tools may allow to derive the correct final classification without necessarily generating the full chain-of-thought produced by the target function.

C Additional related work

Beyond the closest autoregressive PAC works discussed above, we mention three surrounding directions studied extensively in recent years.

Chain-of-thought data in language-model training. Chain-of-thought and scratchpad-style supervision were introduced empirically as ways to expose intermediate computations in language models Nye et al. [2021], Wei et al. [2022b]. This is related to supervised fine-tuning and next-token prediction more broadly Brown et al. [2020], Wei et al. [2022a], following the standard transformer language-modeling paradigm Vaswani et al. [2017], Radford et al. [2019]. Recent RL-based post-training methods also encourage longer reasoning traces Jaech et al. [2024], DeepSeek-AI et al. [2025], but their supervision signal is typically attached to final outcomes rather than to every intermediate token. The cost and importance of curating chain-of-thought data is also emphasized in large-scale model reports Dubey et al. [2024]. Our results formalize this distinction in an online model: observing only the final generated token can increase the mistake bound with the generation length M, while full CoT feedback removes this dependence for finite-Littlestone base classes.

Theory of chain-of-thought. A growing theoretical literature studies why intermediate reasoning traces can help. This includes universality of autoregressive next-token predictors Malach [2023], sample-complexity bounds in terms of CoT information Altabaa et al. [2025], and reinforcement learning after next-token prediction Tsilivis et al. [2025]. Other works study the power and limitations of transformer architectures with chain-of-thought reasoning Merrill and Sabharwal [2023], Li et al. [2024], Chen et al. [2025], Kim and Suzuki [2024], Huang et al. [2025]. These works are mostly statistical, architectural, or optimization-theoretic. In contrast, our results are architecture-free and online: they characterize the possible mistake-bound growth rates of autoregressive learning under end-to-end feedback and show when trajectory feedback eliminates this growth.

Computational hardness. Our computational separation is related in spirit to the broader literature on hardness of distribution-free learning; see Servedio [2025] for a recent survey. Relevant examples include hardness results for intersections of halfspaces Klivans and Sherstov [2009], Tiegel [2024], and DNFs or related classes under average-case or pseudorandomness assumptions Applebaum et al. [2010], Daniely and Shalev-Shwartz [2016], Daniely and Vardi [2021], Applebaum et al. [2025]. Unlike this hardness literature, most of our paper is information-theoretic; the computational result shows that, even for autoregressive linear classifiers, end-to-end feedback can leave a computational obstruction that CoT feedback removes.

D Technical preliminaries

The online autoregressive model studied in this work was formally defined in Section 1.1. The goal of this section is to formally define additional relevant concepts from learning theory, as well as to state known and relevant classic results. Most of the definitions and notation are borrowed from Hanneke et al. [2026].

D.1 Littlestone trees

In this paper, a Littlestone tree (or just a tree) $\mathbf{T}$ is a finite, full, and rooted binary tree, accompanied with the following information:

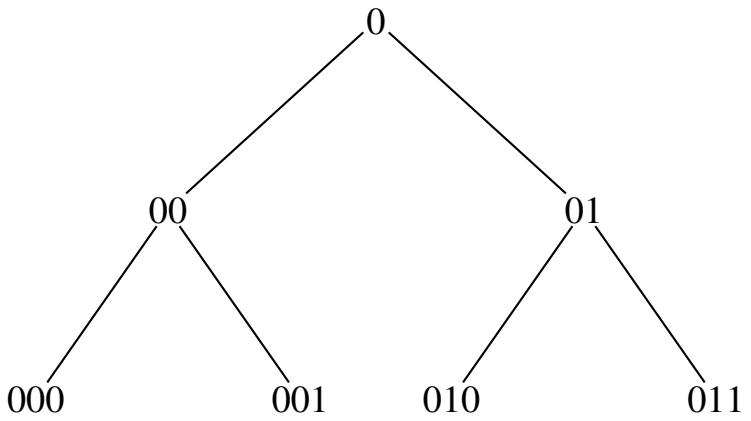

Figure 1: The tree $\mathbf{T}_M(x)$ for $M = 2, x = 0$.

1. Every internal node v is associated with an instance $x \in \mathcal{X}$, denoted as x_v.
2. For every internal node v, the left outgoing edge is labeled with 0, and the right outgoing edge with 1.

The set of vertices and edges of $\mathbf{T}$ are denoted by $v(\mathbf{T})$ and $e(\mathbf{T})$, respectively. Every path in the tree that starts at the root is called a prefix. A prefix that ends at a leaf is called a *branch*. The set of all branches in $\mathbf{T}$ is denoted $\mathcal{B}(\mathbf{T})$. A prefix can be identified by a sequence of vertices starting from the root, or by just the last vertex in the prefix, or by the associated sequence of edges. A prefix $p = v_0, \ldots, v_t$ in $\mathbf{T}$ defines a sample $(x_1, y_1), \ldots, (x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ in the natural way: $x_i = x(v_{i-1})$, and y_i is the label of the edge (v_{i-1}, v_i) for all i. We denote $\boldsymbol{x}(p) = x_1, \ldots, x_t$ and $\boldsymbol{y}(p) = y_1, \ldots y_t$. For a given vertex v, we use $p_{\mathbf{T}}(v)$ (or $p(v)$, when the tree $\mathbf{T}$ is fixed) to denote the path from the root to v. We may overload notation and use $p(v)$ to denote the sample associated with the path, when the context is clear. If there is a bijection between the set of vertices and the set of their instance labelings, we may replace v with its instance label in all notation.

Generation trees. Given a base class $\mathcal{F}$, a generation length M, and an instance x, there is a natural way to represent all possible labels that could be given to x by functions in $\mathcal{F}^{\mathsf{CoT}-M}$ as a decision tree $\mathbf{T}(x) := \mathbf{T}_M(x)$. We call this tree the *generation tree* of x of depth M, or simply the *generation tree* of x, if T is fixed. The tree is constructed as follows. The root of $\mathbf{T}(x)$ is x. The left child of x is labeled by $x0$ and likewise, the right child of x is labeled by $x1$. We continue this labeling process inductively on the children of x, for a total number of T many times. There is a bijection between the set $\{\boldsymbol{y}(b)\}$ for all branches b in $\mathbf{T}(x)$ and the label space $\{0,1\}^T$. Furthermore, note that a function $f^{\mathsf{CoT}-T} \in \mathcal{F}^{\mathsf{CoT}-T}$ realizes a branch b if and only if $f^{\mathsf{CoT}-T}(x) = y(b)$. The branch b may be referred to as the *computation path* of $f^{\mathsf{CoT}-T}$ on x. This representation will be useful in some of the analyses conducted in this work. See a visualized example of a generation tree in Figure 1.

D.2 Class dimensions, learnability, and an SSP lemma

Consider a tree $\mathbf{T}$ and a class $\mathcal{F} \subset \{0,1\}^{\mathcal{X}}$. Given a function $f \in \mathcal{F}$, we say that f *realizes* a prefix $p = v_0, \ldots, v_t$ in $\mathbf{T}$ if $f(\boldsymbol{x}(p)_i) = \boldsymbol{y}(p)_i$ for all i. The set of all functions $f \in \mathcal{F}$ who realizes p are called the *version space* of v_t (or of p, or of $(\boldsymbol{x}(p), \boldsymbol{y}(p))$) with respect to $\mathcal{F}$, which is denoted as $V_{\mathcal{F}}(v_t)$, or simply $V(v_t)$ when $\mathcal{F}$ is fixed. We say that $\mathcal{F}$ *shatters* $\mathbf{T}$, if for every leaf $\ell \in \mathbf{T}$ it holds that $V(\ell) \neq \emptyset$. The set of trees shattered by $\mathcal{F}$ is denoted by $\mathcal{T}(\mathcal{F})$. The set of branches in $\mathbf{T}$ that are realized by $\mathcal{F}$ are denoted by $B_{\mathcal{F}}(\mathbf{T})$. That is, if $\mathcal{F}$ shatters $\mathbf{T}$ then $B_{\mathcal{F}}(\mathbf{T}) = \mathbf{T}$. We now define two fundamental and useful dimensions.

Definition D.1 (Littlestone dimension)**.** The *Littlestone dimension* of $\mathcal{F}$, denoted $\mathtt{L}(\mathcal{F})$, is the maximal depth of a perfect Littlestone tree which is shattered by $\mathcal{F}$. If there is no such maximal depth, we say that $\mathtt{L}(\mathcal{F}) = \infty$.

The Littlestone dimension is known to characterize *online learnability*.

Definition D.2 (Online learnability)**.** In the online learning model, in each round t an adversary sends $x_t \in \mathcal{X}$ to the learner. The learner then predicts[3] $\hat{y}_t \in \mathcal{Y}$, and the adversary then sends the correct label $y_t \in \mathcal{Y}$. Finally, the learner suffers the 0/1 loss $1[\hat{y}_t \neq y_t]$. The adversary is restricted to be *realizable*: there must be $f \in \mathcal{F}$ so that all examples (x_t, y_t) satisfy $y_t = f(x_t)$. A learner's *mistake bound* is the maximal number of mistakes it makes in this setting, against any adversary. We say that $\mathcal{F}$ is *online learnable* with mistake bound L if there exists a learner with mistake bound L such that $L < \infty$.

We will use the following known characterization.

Theorem D.3 (Online learnability characterization Littlestone [1988])**.** *The optimal mistake bound for online learning $\mathcal{F}$ is precisely $\mathtt{L}(\mathcal{F})$. That is, there exists a learner with mistake bound $\mathtt{L}(\mathcal{F})$, and there exists an adversary who can force at least $\mathtt{L}(\mathcal{F})$ many mistakes on any learner.*

For Littlestone classes, there is a useful SSP (Sauer-Shelah-Perles) lemma for trees, proved e.g. by Bhaskar [2021].

Theorem D.4 (SSP for trees Bhaskar [2021])**.** *Let $\mathcal{F} \subset \{0,1\}^{\mathcal{X}}$ be a class, and let $\mathbf{T}$ be a Littlestone tree of depth n with vertices labeled by instances from $\mathcal{X}$. Then*

$$|B_{\mathcal{F}}(\mathbf{T})| \leq \binom{n}{\leq \mathtt{L}(\mathcal{F})} \leq n^{2\mathtt{L}(\mathcal{F})}.$$

The work of Daniely et al. [2015] further extended the definition of the Littlestone dimension to non-binary classes $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$, $|\mathcal{Y}| \geq 2$. In their definition, instead of labeling each left edge with 0 and each right edge with 1, one may choose for every two sibling edges (two edges that are the outgoing edges of a single vertex) two distinct labels from $\mathcal{Y}$ to be labeled by. When $\mathcal{Y} = \{0,1\}$, this definition is reduced to the standard definition. However, for $|\mathcal{Y}| > 2$, Daniely et al. [2015] showed that this definition still characterizes the optimal mistake bound for online learning $\mathcal{F}$.

Theorem D.5 (Multiclass Online learnability characterization Daniely et al. [2015])**.** *The optimal mistake bound for online learning $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ for any $\mathcal{Y}$ is precisely $\mathtt{L}(\mathcal{F})$.*

Some of our results have implications for autoregressive PAC-learning, so for completeness we also formally define PAC-learnability here.

Definition D.6 (PAC learnability)**.** We say that $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ (where $\mathcal{Y}$ is not necessarily binary) is *PAC-learnable* with sample complexity $m := m(\epsilon, \delta)$, if there exists a PAC-learner $\mathsf{Lrn} : (\mathcal{X} \times \mathcal{Y})^{\star} \to \mathcal{Y}^{\mathcal{X}}$, such that for every distribution D over $\mathcal{X}$, for every $\epsilon, \delta > 0$, and for every target function $f_{\star} \in \mathcal{F}$, if $\boldsymbol{x} := x_1, \ldots, x_m \sim_{iid} D$ and $(\boldsymbol{x}, \boldsymbol{y}) := ((x_i, f_{\star}(x_i))_{i=1}^m$, then

$$L_{D,f_{\star}}(f_{\mathsf{Lrn}(\boldsymbol{x},\boldsymbol{y})}) > \varepsilon$$

with probability at most δ, where $f_{\mathsf{Lrn}(\boldsymbol{x},\boldsymbol{y})} := \mathsf{Lrn}(\boldsymbol{x}, \boldsymbol{y})$ and $L_{D,f_{\star}}(f) := \Pr_{x \sim D}[f(x) \neq f_{\star}(x)]$.

Our results have implications for e2e-PAC learning, which is simply PAC-learning of $\mathcal{F}^{\mathsf{e2e}-M}$, and also for CoT-PAC learning, which is the same as e2e-PAC learning, with the difference that training examples reveal the entire chain-of-thought led to the final output. So, as in the online setting, every PAC sample complexity bound for learning $\mathcal{F}^{\mathsf{CoT}-M}$ is also A PAC sample complexity bound for CoT-learning, which is an easier task than the task of PAC-learning $\mathcal{F}^{\mathsf{CoT}-M}$.

We now define the VC-dimension, which is known to characterize the sample complexity of PAC-learning binary classes ($|\mathcal{Y}| = 2$), up to accuracy (ϵ, δ) factors Blumer et al. [1989], Ehrenfeucht et al. [1989], Hanneke [2016].

The VC-dimension is defined as the Littlestone dimension, but restricted only to trees where each branch is labeled by the same sequence of instances. A formal definition is given below.

Definition D.7 (VC-dimension)**.** The *VC dimension* of $\mathcal{F}$, denoted $\mathrm{VC}(\mathcal{F})$, is the maximal depth of a perfect tree which is shattered by $\mathcal{F}$, and also satisfies $\boldsymbol{x}(b_1) = \boldsymbol{x}(b_2)$ for every two branches b_1, b_2. If there is no such maximal depth, we say that $\mathrm{VC}(\mathcal{F}) = \infty$.

Since in the context of the VC-dimension we require that $\boldsymbol{x}(b_1) = \boldsymbol{x}(b_2)$ for every two branches b_1, b_2, instead of discussing trees, we may simply refer to $\boldsymbol{x}(b_1)$ as a *shattered set*. Note that the definition of the VC-dimension immediately implies that $\mathrm{VC}(\mathcal{F}) \leq \mathtt{L}(\mathcal{F})$ for any class $\mathcal{F}$.

[3] We only discuss deterministic learners.

As the result below states, analyzing the Littlestone dimension of a class $\mathcal{F}$ could have an advantage in the case where $|\mathcal{Y}| > 2$.

Theorem D.8 (Online-to-PAC conversion Littlestone [1989])**.** *If $\mathtt{L}(\mathcal{F}) < \infty$, then $\mathcal{F}$ is learnable with sample complexity*

$$m(\varepsilon, \delta) = O\left(\frac{\mathtt{L}(\mathcal{F}) + \log(1/\delta)}{\varepsilon}\right).$$

In the case $|\mathcal{Y}| = 2$, Theorem D.8 does not give any advantage, since Hanneke [2016] proved the same bound with $\mathtt{L}(\mathcal{F})$ replaced by $\mathrm{VC}(\mathcal{F})$, and $\mathrm{VC}(\mathcal{F}) \leq \mathtt{L}(\mathcal{F})$. However, Theorem D.8 applies also to the case $|\mathcal{Y}| > 2$, while bounds depending on the VC-dimension are not, since the VC-dimension is defined only over binary classes.

E Upper bound for end-to-end learning

Theorem E.1. *Let $\mathcal{F}$ be a base class. Then for all $M \geq 2$:*

$$\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = O(\mathtt{L}(\mathcal{F}) \log(M\mathtt{L}(\mathcal{F}))).$$

Proof. Let $\mathbf{T}$ be a Littlestone tree of depth m which is shattered by $\mathcal{F}^{\mathsf{e2e}-M}$. We will show that $m = O(\mathtt{L}(\mathcal{F}) \log(M\mathtt{L}(\mathcal{F})))$, which implies the lemma. For every node $u \in \{0,1\}^{<m}$ of $\mathbf{T}$ (where the node is given by the labels of the edge-path from the root to the node), let x_u be the instance labeling that node. Now, we inflate the tree $\mathbf{T}$ to a tree $\mathbf{T}_{\mathcal{F}}$ of depth mM that takes into account the intermediate autoregressive steps. The idea is to inflate every two sibling edges in $\mathbf{T}$ to possible paths of length M using a generation tree. Let us formally describe the recursive construction of $\mathbf{T}_{\mathcal{F}}$. We maintain a saved path $v \in \{0,1\}^{<m}$ in the original tree $\mathbf{T}$ saved in every node of $\mathbf{T}_{\mathcal{F}}$. In the root of $\mathbf{T}_{\mathcal{F}}$, the saved path is $\emptyset$ (the empty path), and every node inherits the saved path from its parent, until we explicitly update the saved path. For the base of the recursion, we construct the generation tree $\mathbf{T}_M(x_\emptyset)$. For the recursive step, we either finish construction, or paste another generation tree at every leaf of the tree constructed so far. Concretely, for every leaf of the tree constructed so far, if v is the saved path of this leaf and $|v| = m$, finish construction. Otherwise, if it is a left leaf (that is, a leaf that lies at the tip of an edge labeled with 0), paste $\mathbf{T}_M(x_{v0})$ at this leaf, and update the saved path to $v0$. Otherwise, paste $\mathbf{T}_M(x_{v1})$ at this leaf, and update the saved path to $v1$.

For a class $\mathcal{F}$ and a tree $\mathbf{T}$, let $B_{\mathcal{F}}(\mathbf{T})$ be the set of branches of $\mathbf{T}$ that are realized by $\mathcal{F}$. Now, note that

$$|B_{\mathcal{F}}(\mathbf{T}_{\mathcal{F}})| \geq 2^m. \tag{3}$$

Indeed, there is a one-to-one function G from the branches of $\mathbf{T}$ to the branches of $B_{\mathcal{F}}(\mathbf{T}_{\mathcal{F}})$. For a branch $u \in \{0,1\}^m$ of $\mathbf{T}$, let $f_u \in \mathcal{F}$ such that $f_u^{\mathsf{e2e}-M}$ realizes u. Then by construction of $\mathbf{T}_{\mathcal{F}}$, $f_u \in \mathcal{F}$ agrees with some branch in $\mathbf{T}_{\mathcal{F}}$ that its leaf has the saved path u. So, $G(u)$ returns this branch. Therefore $G(u) \in B_{\mathcal{F}}(\mathbf{T}_{\mathcal{F}})$ and G is clearly a bijection. On the other hand, Theorem D.4 implies

$$|B_{\mathcal{F}}(\mathbf{T}_{\mathcal{F}})| \leq (mM)^{2\mathtt{L}(\mathcal{F})}. \tag{4}$$

From (3) and (4), we obtain:

$$2^m \leq (mM)^{2\mathtt{L}(\mathcal{F})}.$$

Solving this inequality for m implies the stated bound. □

F The taxonomy of end-to-end learning

We have shown that for any base class $\mathcal{F}$ and any generation length $M \geq \mathtt{L}(\mathcal{F})$ we have $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = O(\mathtt{L}(\mathcal{F}) \log M)$. Below, we prove that essentially any "well-behaved" sub-logarithmic growth rate of $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M})$ between constant and logarithmic in M is realized by some Littlestone class. We further show that for all d, there exists a class $\mathcal{F}$ with $\mathtt{L}(\mathcal{F}) = O(d)$ and

$$\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) \geq \mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = \Omega(\mathtt{L}(\mathcal{F}) \log M) \tag{5}$$

for all M. This essentially completes the taxonomy of online e2e-learning of Littlestone classes. Along the way, the right-hand side of 5 resolves an open question of Joshi et al. [2025], who proved

that $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = O(\mathtt{L}(\mathcal{F})\log M)$ for all $\mathcal{F}, M$, and asked if this upper bound is tight for some classes.

Before we formally state the results, we define precisely what we mean by a "well-behaved" sub-logarithmic growth rate. Fix a sufficiently large universal constant M_0, to be used throughout the section. We say that a function $r : \mathbb{N}_+ \to \mathbb{N}_+$ is a *well-behaved sub-logarithmic growth rate* if:

1. r is sub-logarithmic: $r(M) \leq \frac{1}{4}\log(M)$ for all $M \geq M_0$.
2. r is non-decreasing : $M_1 \geq M_2 \implies r(M_1) \geq r(M_2)$ for all $M_1, M_2 \geq 1$.
3. r is sub-additive: $r(M_1 + M_2) \leq r(M_1) + r(M_2)$ for all $M_1, M_2 \geq 1$.

Note that all natural sub-logarithmic growth rates such as $\lceil\sqrt{\log M}\rceil$, $\lceil\log\log M\rceil$ etc. satisfy our definition. We can now state the result for well-behaved sub-logarithmic rates.

Theorem F.1. *For any well-behaved sub-logarithmic growth rate r there exists a base class $\mathcal{F}$ such that:*

1. $\mathtt{L}(\mathcal{F}) = 1$.
2. $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = \Theta(r(M))$ *for all* $M \geq M_0$.

The definition of $\mathcal{F}$ and the lower bound strategy of the second item are due to unpublished private communication Hanneke and Moran [2026].

For logarithmic (in contrast to sub-logarithmic) rates, we prove the following

Theorem F.2. *For any $d \in \mathbb{N}_+$, there exists a class $\mathcal{F}$ such that $\mathtt{L}(\mathcal{F}) = O(d)$ and*

$$\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = \Omega(\mathtt{L}(\mathcal{F})\log M)$$

for all $M \geq M_0$.

Together with the upper bound of Theorem E.1, this establishes that for any $d \in \mathbb{N}_+$ there exists a class $\mathcal{F}$ such that $\mathtt{L}(\mathcal{F}) = O(d)$ and

$$\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}), \mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = \Theta(\mathtt{L}(\mathcal{F})\log M)$$

for all $M \geq \max\{M_0, \mathtt{L}(\mathcal{F})\}$.

In the following sections, we prove Theorem F.1 and Theorem F.2.

F.1 Proof of Theorem F.1

Fix the growth rate r.

We first define the class $\mathcal{F}$. For every $s \geq M_0$ let

$$K_s := \{r(s), r(s)+1, \ldots, r(s)+2^{r(s)}-1\}.$$

That is, K_s is simply the set $\{0, \ldots, 2^{r(s)}-1\}$ of size $2^{r(s)}$, with $r(s)$ added to each element. We now define a "baseline" function $f : \{0,1\}^\star \to \{0,1\}$ as follows. For every $s \geq M_0$, $k \in K_s$, $j \in \{0, \ldots, r(s)-1\}$ and $z \in \{0,1\}^j$, set:

$$f(0^s 1 0^k 1 0^{s-k-1} z) = (k - r(s))_{j+1}$$

where $(k-r(s))_{j+1}$ is the $(j+1)^{st}$ bit in the binary representation of $k - r(s)$. For any other string x that does not fit the form above, we set $f(x) = 0$. Note that 0^{s-k-1} is well-defined, since

$$k + 1 \leq r(s) + 2^{r(s)} \leq \frac{1}{4}\log s + 2^{\frac{1}{4}\log s} \leq \sqrt{s} \leq s - 1, \tag{6}$$

where the final inequality holds since $s \geq M_0$. Thus $s - k - 1 \geq 1$. Now, we define for each fixed pair s and $k \in K_s$ the following single-point modification of f:

$$f_{s,k}(x) = \begin{cases} 1 & x = 0^s 1 0^k, \\ f(x) & \text{otherwise.} \end{cases}$$

We may now define the class $\mathcal{F}$ as

$$\mathcal{F} := \{f_{s,k} : s \geq M_0, k \in K_s\}.$$

We first prove the first item of Lemma F.1.

Lemma F.3. *We have* $\mathtt{L}(\mathcal{F}) = 1$.

Proof. We have $\mathtt{L}(\mathcal{F}) \geq 1$ since $\mathcal{F}$ contains more than one function. To see that $\mathtt{L}(\mathcal{F}) \leq 1$, consider the following online learner for $\mathcal{F}$. The learner predicts as f until it makes its first mistake. Once it makes a mistake, it discovers the correct function and always predicts as this function. □

We now prove the lower bound in the second item of Theorem F.1. We prove it by showing the stronger claim $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) \geq r(M)$ for all $M \geq M_0$.

Lemma F.4. *Fix* $M \geq M_0$. *Then the set*

$$A_M := \{a_i := 0^M 10^i : 1 \leq i \leq r(M)\}$$

is shattered by $\mathcal{F}^{\mathsf{e2e}-M}$.

Proof. Let $\boldsymbol{y} = y_1, \ldots y_{r(M)}$ be a labeling of A_M. It suffices to show that this labeling is realized by $\mathcal{F}^{\mathsf{e2e}-M}$. Let b be the number given in binary representation by the vector $\boldsymbol{y}$. and let $k := r(M) + b \in K_M$. We claim that for all i, $f_{M,k}^{\mathsf{e2e}-M}(a_i) = b_i$. Let's describe the M-length computation path of $f_{M,k}^{\mathsf{CoT}-M}$ on a_i:

1. In the first $k - i \geq b$ steps, $f_{M,k}$ agrees with f, which means that 0 is appended.
2. After the first $k - i \leq \sqrt{M}$ steps (the inequality is by (6)), the computation path reaches the instance $0^M 10^k$, so in step $k - i + 1$, 1 is appended, by definition of $f_{M,k}$.
3. By definition of f, 0 is appended in the following $M - k - 1$ steps.
4. So far, we completed a computation of length $k - i + 1 + M - k - 1 = M - i$, so we have i more steps. By definition of f, in the last i steps of the computation, the generated bits are $\boldsymbol{y}_{\leq i}$.

Overall, we deduce that:

$$f_{M,k}^{\mathsf{CoT}-M}(a_i) = 0^{k-i} 10^{M-k-1} y_1, \ldots, y_i,$$

which implies $f_{M,k}^{\mathsf{e2e}-M}(a_i) = y_i$. □

Lemma F.4 immediately implies the lower bound of the first item of Theorem F.1. We now turn to the upper bound.

For every $s \geq M_0$ define the "bucket"

$$B_s := \{0^s 10^i : i \geq 0\}.$$

Lemma F.5. *Fix* $M, s \geq M_0$ *and* $k \in K_s$. *For all* $x \notin B_S$:

$$f_{s,k}^{\mathsf{e2e}-M}(x) = f^{\mathsf{e2e}-M}(x).$$

Proof. Since $f_{s,k}$ and f differ only on the instance $x_{s,k} := 0^s 10^k$, it is possible that $f_{s,k}^{\mathsf{e2e}-M}(x) \neq f^{\mathsf{e2e}-M}(x)$ only if $x_{s,k}$ is an instance appearing in an intermediate step in the generation of $f^{\mathsf{CoT}-M}(x)$. This is possible only if x is a prefix of $x_{s,k}$. So, we only need to analyze the case where x is both a prefix of $x_{s,k}$ and not in the bucket B_s. So the form of x must be $x = 0^n$ for some $1 \leq n \leq s$. In this case, f appends 0 in every step, so $x_{s,k}$ never appears. □

We now prove another lemma that handles the case $s \gg M$, where B_s is the "correct" bucket. The lemma establishes that in this case, for any function in $\mathcal{F}^{\mathsf{e2e}-M}$ there is at most a single instance it labels with 1.

Lemma F.6. *Fix* $s \geq M_0$, $s \geq 10M$, $i \geq 0$ *and* $k \in K_s$. *Then*

$$f_{s,k}^{\mathsf{e2e}-M}(0^s 10^i) = 1 \iff M = k - i + 1.$$

Proof. For every point of the form in the statement of the lemma, f appends 0, Therefore, it is only possible that $f_{s,k}(0^s10^j) = 1$ in the case where $f_{s,k}$ differs from f, that is, in the case $j = k$. Therefore, if $i > k$ then the instance 0^s10^k is never reached in the autoregressive generation. Thus, we assume that $i \leq k$. We consider the three possible sub-cases.

1. If $M < k - i + 1$, then 0 the autoregressive function of $f_{s,k}$ appends 0 for at most $k - i$ many times, and thus $f_{s,k}$ is never receives 0^s10^k as input, and so $f_{s,k}^{\mathsf{e2e}-M}(0^s10^i) = 0$.

2. If $M = k - i + 1$, then $f_{s,k}^{\mathsf{e2e}-M}(0^s10^i) = 1$, since in the last step the instance 0^s10^i is given as input to $f_{s,k}$.

3. If $M > k - i + 1$, then the instance 0^s10^k is given as input to $f_{s,k}$, and then there is at least one more step before the autoregressive generation of CoT is done, since $M > k - i + 1$. The definition of f implies that the following $s - k - 1$ bits after 0^s10^k1 are 0, and only then 1 could be appended. However, we have

$$s - k - 1 \geq s - 2^{\frac{1}{2}\log s} - 1 \geq s - \sqrt{s} - 1 \geq s/2 \geq 5M > M,$$

where the first inequality is since $k \leq r(s) + 2^{r(s)}$ and the assumption $r(s) \leq \frac{1}{4}\log s$, and the third inequality holds since $s \geq M_0$. So, only zeroes are appended, and thus $f_{s,k}^{\mathsf{e2e}-M}(0^s10^i) = 0$.

□

Lemma F.7. *For all $M \geq M_0$:*

$$\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) \leq 20r(M).$$

Proof. We construct an online learner that makes at most $20r(M)$ many mistakes.

As long as the learner hasn't made a single mistake, it predicts as $f^{\mathsf{e2e}-M}$ on every instance. So, the number of mistakes made in this phase is at most 1.

After the first mistake on an instance x, the learner changes its behavior as follows. Let $f_{s^\star,k^\star}$ be the (unknown) target concept. So, we know that $f_{s^\star,k^\star}^{\mathsf{e2e}-M}(x) \neq f^{\mathsf{e2e}-M}(x)$. Therefore, Lemma F.5 implies that $x \in B_{s^\star}$. Thus $x = 0^{s^\star}10^i$ for some $i \in \mathbb{N}$, so the learner now knows $s^\star$ by just looking at x. Thus, from this point, the learner only needs to learn the class

$$\mathcal{F}_{s^\star} := \{f_{s^\star,k^\star} : k \in K_{s^\star}\}.$$

We consider the two possible cases.

First, suppose that $s^\star \leq 10M$. In this case, the learner runs the halving algorithm over $\mathcal{F}_{s^\star}$. The mistake bound in this case is

$$\log|\mathcal{F}_{s^\star}| = \log 2^{r(s)} = r(s) \leq r(10M) \leq 10r(M),$$

where the first inequality is since r is non-decreasing, and the second inequality is since r is sub-additive.

In the complementing case, we have $s^\star > 10M$. As usual, for $x \notin B_{s^\star}$, the learner predicts as f, and thus is always correct for $x \notin B_{s^\star}$. For $x \in B_{s^\star}$, that is, x of the form $0^{s^\star}10^i$, the learner always predicts 0, until making a mistake. Once it makes a mistake on the instance $0^{s^\star}10^i$ for some $i \in \mathbb{N}$, it means that $f_{s^\star,k^\star}^{\mathsf{e2e}-M}(0^{s^\star}10^i) = 1$. Lemma F.6 implies that in this case $k^\star = M + i - 1$. So the learner now knows both $s^\star$ and $k^\star$, and therefore knows the target concept $f_{s^\star,k^\star}$ and can always predict as $f_{s^\star,k^\star}$.

Summing all mistakes, we deduce that overall the learner makes at most $1 + 10r(M)$ many mistakes, which concludes the proof. □

Proof of Theorem F.1. The theorem is given by combining Lemma F.3, Lemma F.4, and Lemma F.7. □

F.2 Proof of Theorem F.2

We show that for any $d \in \mathbb{N}$, there exists a class $\mathcal{F}$ with $\mathtt{L}(\mathcal{F}) = O(d)$, and such that $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = \Omega(d \log M)$ for all sufficiently large M. Our general upper bound asserts that $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = O(d \log M)$ for all $M \geq \mathtt{L}(\mathcal{F})$, and thus $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = \Theta(d \log M)$.

Our construction solves an open question from Joshi et al. [2025]. They showed that for any class $\mathcal{F}$ and all M, it holds that $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = O(\mathtt{L}(\mathcal{F}) \log M)$, and asked if the logarithmic dependence (or, in fact, any dependence) on M is necessary. Our proof gives a positive answer to this question.

The proof is via the probabilistic method applied to a "hard" random class defined below.

F.2.1 The hard class

Fix $d, M \in \mathbb{N}_+$. Let

$$N := M^{10d}, \quad Z := \{0^i : i \in [M^2]\}.$$

be the size of the class, and Z be the set of instances on which the class is "interesting". Partition Z to

$$Z_1 := \{0^{jM : j \in [M]}\}, \quad Z_0 := Z \backslash Z_1,$$

so $|Z_1| = M, |Z_0| = M^2 - M$. We now draw N functions independently at random. For each $a \in [N]$ and $x \in \{0,1\}^\star$, define $f_a(x)$ as follows:

- If $x \notin Z$, let $f_a(x) = 0$.
- Otherwise, if $x \in Z_1$, let $f_a(x) = 1$ w.p. $1 - 1/M$, and $f_a(x) = 0$ w.p. $1/M$.
- Otherwise, if $x \in Z_0$, let $f_a(x) = 0$ w.p. $1 - 1/M$, and $f_a(x) = 1$ w.p. $1/M$.

Now, let

$$\mathcal{F}_{d,M} := \{f_a : a \in [N]\}.$$

For each $x \in Z_0$, we say that 1 is the *minority* label of x, and 0 is the *majority* label of x. We define the analog for Z_1.

We are now ready to ready to begin with the proof of Theorem F.2. We begin by proving the upper bound $\mathtt{L}(\mathcal{F}_{d,M}) = O(d)$ with high probability.

F.2.2 Upper bounding the Littlestone dimension of the base class

The intuition for this bound is that since $|\mathcal{F}| = N = M^{10d}$ and for each instance there exists a label $r \in \{0,1\}$ for which typically only a $1/M$-fraction of the version agrees with, the version space should be empty at the bottom of some branch of length $O(d)$. However, this is only true on *average*, while the Littlestone dimension is determined by the *maximal* depth of a shattered tree, so the proof is in fact more challenging than this intuition. We use the technique of Chase and Mehalel [2024], who provided such an analysis for a similar random class.

Lemma F.8. *Let $d, M \in \mathbb{N}$ larger than some large enough universal constant. With probability at least $3/4$, we have*

$$\mathtt{L}(\mathcal{F}_{d,M}) \leq 100d.$$

Proof. Denote $\mathcal{F} := \mathcal{F}_{d,M}$. We show that the probability that $\mathcal{F}$ shatters a tree of depth $100d$ is at most $1/4$. Fix a tree $\mathbf{T}$ whose internal nodes are labeled by instances from $\{0,1\}^\star$, and that is shattered by $\mathcal{F}$. If $\mathbf{T}$ contains nodes labeled with instances outside of Z it cannot be shattered, so we may assume that all nodes are labeled by instances from Z. So, every node has precisely one *minority edge*, corresponding to the minority label of the instance labeling it. Let $B_{1/2}(\mathbf{T})$ be the set of branches in $\mathbf{T}$ in which precisely $50d$ of the edges are minority edges. Note that

$$\left|B_{1/2}(\mathbf{T})\right| = \binom{100d}{50d} \geq \frac{2^{100d}}{2\sqrt{50d}} \geq \frac{2^{100d}}{20\sqrt{d}}, \tag{7}$$

where the first inequality is by the known bound $\binom{2n}{n} \geq \frac{2^{2n}}{2\sqrt{n}}$, applied with $n = 50d$. Since $\mathbf{T}$ is shattered, in particular every branch of $B_{1/2}(\mathbf{T})$ is realized by at least one function $f \in \mathcal{F}$. Fix a branch $b \in B_{1/2}(\mathbf{T})$ and $f \in \mathcal{F}$. Since the draw of f is independent for each instance

$$\Pr[f \text{ realizes } b] \leq \frac{1}{M^{50d}}. \tag{8}$$

Now, note that each function can realize at most a single branch. Therefore, if $\mathbf{T}$ is shattered, then there exists an injective mapping $\phi : B_{1/2}(\mathbf{T}) \to [N]$ such that for all $b \in B_{1/2}(\mathbf{T})$, $f_{\phi(b)}$ realizes b. That is ϕ assigns to each $b \in B_{1/2}(\mathbf{T})$ the index of the function realizing it, and the same function cannot be assigned to two different branches. Fix a mapping ϕ. Since the probability that $f_{\phi(b)}$ realizes b is independent of the other randomness, and using (8), the probability that for all $b \in B_{1/2}(\mathbf{T})$, $f_{\phi(b)}$ realizes b is at most

$$\left(\frac{1}{M^{50d}}\right)^{|B_{1/2}(\mathbf{T})|}. \tag{9}$$

Since there are at most $N^{|B_{1/2}(\mathbf{T})|} = M^{10d|B_{1/2}(\mathbf{T})|}$, using (9) and (7), the probability that there exists ϕ so that for all $b \in B_{1/2}(\mathbf{T})$, $f_{\phi(b)}$ realizes b is at most

$$M^{10d|B_{1/2}(\mathbf{T})|}\left(\frac{1}{M^{50d}}\right)^{|B_{1/2}(\mathbf{T})|} = \left(\frac{1}{M^{40d}}\right)^{|B_{1/2}(\mathbf{T})|} \leq \left(\frac{1}{M^{40d}}\right)^{\frac{2^{100d}}{20\sqrt{d}}} \leq \frac{1}{M^{2\sqrt{d}\cdot 2^{100d}}}. \tag{10}$$

So, we deduce that for a fixed tree $\mathbf{T}$, the probability that it is shattered by $\mathcal{F}$ is at most $\frac{1}{M^{2\sqrt{d}\cdot 2^{100d}}}$. The number of perfect trees of depth $100d$ labeled by instances from Z is at most $|Z|^{2^{100d}} = M^{2\cdot 2^{100d}}$. Thus, using 10, the probability that there exists a tree $\mathbf{T}$ of depth $100d$ which is shattered by $\mathcal{F}$ is at most

$$M^{2\cdot 2^{100d}}\frac{1}{M^{2\sqrt{d}\cdot 2^{100d}}} = M^{2\cdot 2^{100d}(1-\sqrt{d})} < 1/4,$$

which concludes the proof. □

We now turn to the lower bound $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) = \Omega(d \log M)$ with high probability. To formalize a choice of a random subset before it is drawn, we use standard propositional logic formalization, as defined below.

F.2.3 A framework for choosing subsets from a random class

We define a notion of *rules* for choosing a subset of a random class $\mathcal{F}$ before it is drawn, by restricting the labels of some the instances in the domain. A rule is any restriction on the labels of some of the instances, that can be represented as a propositional logic formula with the conjunction and disjunction connectives, where the examples $(x, y) \in \mathcal{X} \times \mathcal{Y}$ are the atomic variables, and the assignments are given by the functions of the class. A rule is said to be *independent* of an instance x, if x does not appear in any of the atomic variables used in the rule. For example, if $x_1, x_2, x_3, x_4 \in \mathcal{X}$, then $((x_1, 0) \wedge (x_2, 1)) \vee (x_3, 0)$ is a rule, and it is independent of x_4. The *truth value* of a variable (example) (x, y) under the assignment (function) $f \in \{0, 1\}^{\mathcal{X}}$ is naturally defined as $f(x, y) := 1[f(x) = y]$, where $f(x, y) = 1$ represents *true* and $f(x, y) = 0$ *false*. The truth value of a rule R under the function f, denoted by $f(R)$, is inductively defined in the standard way used in propositional logic. A rule R for the class $\mathcal{F}$ naturally defines a subclass of $\mathcal{F}$, denoted as $\mathcal{F}(R)$, of all functions satisfying the rule:

$$\mathcal{F}(R) = \{f \in \mathcal{F} : f(R) = 1\}.$$

Given a decision tree, any branch b in the tree naturally induces the rule $(\boldsymbol{x}(b), \boldsymbol{y}(b))$ (formally, we need to replace the commas in the sequence $(\boldsymbol{x}(b), \boldsymbol{y}(b))$ to $\wedge$ connectives, to make it a valid rule, but it is understood that $(\boldsymbol{x}(b), \boldsymbol{y}(b))$ represents this rule). Likewise, any two branches b_1, b_2 naturally induce the rule $(\boldsymbol{x}(b_1), \boldsymbol{y}(b_1)) \vee (\boldsymbol{x}(b_2), \boldsymbol{y}(b_2))$, and this can be extended to any number of branches. For a set of branches B, we let $R(B)$ be the rule induced by B.

We now use this framework and prove a standard concentration lemma for predetermined rules defined over $\mathcal{F}_{d,M}$.

F.2.4 A concentration lemma for predetermined rules

Lemma F.9. *Let $d \in \mathbb{N}$ larger than some large enough universal constant, and let $M \geq d^2$. Let $\mathcal{R}$ be a family of rules such that $|\mathcal{R}| \leq e^{M^{6d}}$. Then w.p. at least $3/4$ over the draw of $\mathcal{F}_{d,M}$, for all $x \in Z_r$ (where $r \in \{0, 1\}$) and all $R \in \mathcal{R}$, such that R is independent of x and $|\mathcal{F}_{d,M}(R)| \geq M^{8d}$, we have*

$$\frac{1}{2M}|\mathcal{F}_{d,M}(R)| \leq |\mathcal{F}_{d,M}(R \wedge (x, 1-r))| \leq \frac{3}{2M}|\mathcal{F}_{d,M}(R)|.$$

Proof. Since R is independent of x, we have

$$X := |\mathcal{F}_{d,M}(R \wedge (x, 1-r))| \sim \mathrm{Bin}(|\mathcal{F}_{d,M}(R)|, 1/M), \quad \mu := \mathbb{E}[X] = |\mathcal{F}_{d,M}(R)|/M \geq M^{8d-1}.$$

Multiplicative Chernoff bound thus gives

$$\Pr\left[X \notin \left[\frac{\mu}{2}, \frac{3\mu}{2}\right]\right] \leq 2e^{-\mu/12} \leq e^{-M^{8d-1}/12}.$$

Taking a union bound over all $x \in Z, R \in \mathcal{R}$, the failure probability that there exist a pair (x, R) not satisfying the lemma is at most

$$|Z||\mathcal{R}|e^{-M^{8d-1}/12} \leq M^2 e^{M^{6d}} e^{-M^{8d-1}/12} < 1/4,$$

where the final inequality holds for large enough d. □

F.2.5 Lower bounding the VC-dimension of the end-to-end class

We are now ready to prove the lower bound on $\mathrm{VC}(\mathcal{F}_{d,M}^{\mathsf{e2e},M})$. In light of Lemma F.9, we fix a class $\mathcal{F} := \mathcal{F}_{d,T}$ drawn as in Section F.2.1, for which the lemma holds.

Lemma F.10. *We have*

$$\mathrm{VC}(\mathcal{F}^{\mathsf{e2e},M}) = \Omega(d \log M).$$

Let $m := \left\lfloor \frac{d \log M}{600} \right\rfloor$. For each $j \in [m]$, let $x^{(j)} := 0^{(j-1)M+1}$, and denote $S := \{x^{(j)} : j \in [m]\}$. We will prove that S is shattered by $\mathcal{F}^{\mathsf{e2e},M}$.

The main tool used to prove Lemma F.10 is the concentration lemma proved above. To use it, we need to define set $\mathcal{R}$ of rules determined before drawing $\mathcal{F}_{d,M}$. We will need two different types of rules, one to control the version space, and one to control any possible specific autoregressive trajectory. Let us start with the first type. For each $j \in [m]$, consider the generation tree of depth M rooted at $x^{(j)}$. In this tree, define

- The *green branch*, given by the sequence $g^{(j)} := 0^{T-1}1$ of edge labels.
- The *red branches*, given by the sequences

$$r_q^{(j)} := 0^{q-1}10^{M-q}, \quad q \in [M-1].$$

See Figure 2 for an illustration of the green and red branches, as well as an intuitive explanation for how they can be used to yield the lower bound on $\mathrm{VC}(\mathcal{F}_{d,M}^{\mathsf{e2e},M})$.

Now, let

$$G^{(j)} := \{g^{(j)}\}, \quad R^{(j)} := \{r_q^{(j)} : q \in [M-1]\}$$

be the sets of green and red branches, respectively. We now define for every possible labeling $u := (u_1, \ldots, u_t) \in \{0,1\}^t$ of $(x^{(1)}, \ldots, x^{(t)})$ where $t \leq m$, a rule Q_u as follows:

$$Q_u := \bigwedge_{j \in [t]: u_j = 1} R(G^{(j)}) \wedge \bigwedge_{j \in [t]: u_j = 0} R(R^{(j)}).$$

That is, Q_u requires to satisfy the rule induced by the green branch $g^{(j)}$ for $x^{(j)}$ that are labeled by 1, and to satisfy the rule induced by the red branches $R^{(j)}$ for $x^{(j)}$ that are labeled by 0. This makes sense, as one can note that

- $f \models R(G^{(j)}) \implies f^{\mathsf{e2e}-M}(x^{(j)}) = 1.$
- $f \models R(\{r_q^{(j)}\}) \implies f^{\mathsf{e2e}-M}(x^{(j)}) = 0$, for any $q \in [M-1]$.

Now, let

$$\mathcal{R}_{ver} := \{Q_u : u \in \{0,1\}^t, t \in [m]\}.$$

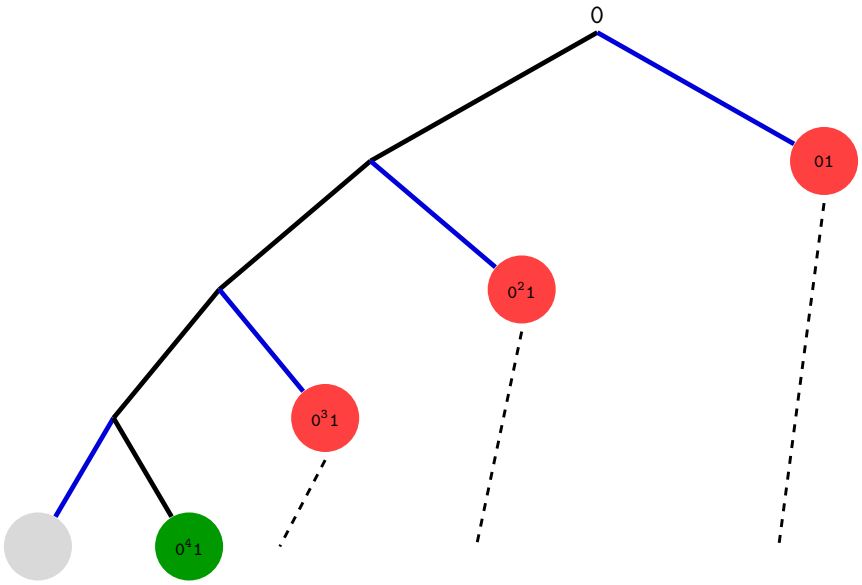

Figure 2: The generation tree $\mathbf{T}_M(x^{(j)})$, visualized for the hard class $\mathcal{F}_{d,M}$ with $M = 4$ and $j = 1$ (that is, $x^{(j)} = 0$). Note that there are $M - 1$ red nodes, where the version space of each typically contains an $O(1)/M$ fraction of the class, and thus in total a $(M-1)\frac{O(1)}{M} = O(1)$ fraction of the class is contained in the union of red vertices. The edges where the size of the class drops by a factor of $O(1)/M$ are blue. In the rest of the edges, the size of the version space only drops by a factor of at least $1 - O(1)/M$. Therefore, a fraction of $(1 - O(1)/M)^M = O(1)$ of the class is typically contained in the green vertex version space. Furthermore, for all f in the version space of the red nodes, we have $f^{\mathsf{e2e}-M}(0) = 0$, and for all f in the version space of the green node, we have $f^{\mathsf{e2e}-M}(0) = 1$. So, for each possible choice of labeling for 0, we get to keep a constant factor of the version space. Applying this idea repeatedly allows to shatter a set of size $\Omega(\log(M^d)) = \Omega(d \log M)$.

We now turn to the second type. For two instances $x, y \in \{0,1\}^\star$ and $t \leq |y|$, we define the appropriate rule for the t-prefix path that begins at x and follows the trajectory given by the first t bits of y as

$$P_t(x, y) := \bigwedge_{s=1}^{t} (xb_1 \dots b_{s-1}, b_s).$$

In the case where we consider the entire trajectory of y, that is $t = |y|$, we abbreviate $P(x, y) := P_{|y|}(x, y)$. So, $f \models P_t(x, y)$ means that when starting from the instance x, the first t bits autoregressivly generated by f are precisely $y_1, \dots, y_t$. Now we can also define a rule that captures functions from a specific version space given by a labeling $u \in \{0,1\}^{<m}$, that also follow a specific trajectory of a green or red branch starting at the next instance $x^{(|u|+1)}$. Formally, let:

$$\mathcal{R}_{bran} := \{Q_u \wedge P_t(x^{(|u|+1),b}) : u \in \{0,1\}^{<m}, t \in [T], b \in G^{(|u|+1)} \cup R^{(|u|+1)}\}.$$

Now, let

$$\mathcal{R} := \mathcal{R}_{ver} \cup \mathcal{R}_{bran}$$

To use Lemma F.9, we need to show that $\mathcal{R}$ is not too large.

Lemma F.11. *We have*

$$|\mathcal{R}| \leq M^{2d}$$

Proof. We have

$$|\mathcal{R}_{ver}| \leq m2^m,$$

since there are at most $m2^m$ options for the choice of u. directly from the definition. Note that for all $j \in [m]$, $|G^{(j)} \cup R^{(j)}| \leq M$. Thus

$$|\mathcal{R}_{bran}| \leq m2^m \cdot M \cdot M$$

since there are at most $m2^m$ options for the choice of u, at most T options for the choice of t, and at most M options for the choice of b. So, over all:

$$|\mathcal{R}| \leq |\mathcal{R}_{ver}| + |\mathcal{R}_{bran}| \leq m2^m(M^2+1) \leq d\log M \cdot M^d(M^2+1) \leq M^{2d},$$

which concludes the proof. □

We now show that when committing to a label of another instance, the version space does not decrease by too much.

Lemma F.12. *Let $u \in \{0,1\}^t$ where $t < m$, and suppose that $|\mathcal{F}(Q_u)| \geq \frac{1}{e^{300|u|}}N$. Then for any $r \in \{0,1\}$:*

$$|\mathcal{F}(Q_{ur})| \geq \frac{1}{e^{300}}|\mathcal{F}(Q_u)|.$$

Proof. To use Lemma F.9, we have to show that any rule R we use, it holds that $|\mathcal{F}(R)| \geq M^{8d}$. We will use rules of the form

$$Q_u \wedge P_t(x^{|u|+1}, b), \tag{11}$$

which are indeed in $\mathcal{R}_{bran} \subset \mathcal{R}$. where $t \in [M]$, and $b \in G^{(|u|+1)} \cup R^{(|u|+1)}$. By assumption,

$$|\mathcal{F}(Q_u)| \geq \frac{1}{e^{300|u|}}N \geq \frac{1}{M^d}M^{10d} = M^{9d},$$

so the bound of Lemma F.9 holds for Q_u. Now, from monotonicity of the inclusion relation, to prove that for each possible $P_t(x^{|u|+1}, b)$ we have $\mathcal{F}(Q_u \wedge P_t(x^{|u|+1}, b)) > M^{8d}$, it suffices to show that it holds for the worst-case $P_t(x^{|u|+1}, b)$. In the worst case, b has a single minority edge and $T-1$ majority edges. Thus, using $(1-100/M)^M \geq e^{-200}$ for large enough M, for any rule of the form of (11) we have

$$\mathcal{F}(Q_u \wedge P_t(x^{|u|+1}, b)) \geq \frac{1}{100M}(1-100/M)^M|\mathcal{F}(Q_u)| \geq \frac{1}{100M}e^{-200}M^{9d} \geq M^{9d-2},$$

where the final inequality holds for large enough M. We can now prove the statement of the lemma. Let us start with the case $r = 1$. Recall that any function f realizing the green branch $g^{(|u|+1)}$ satisfies $f^{\mathsf{e2e}-M}(x^{(|u|+1)}) = 1$. So, it suffices to lower bound the number of such functions. Since the green branch has only majority edges, we obtain:

$$|\mathcal{F}(Q_{u1})| \geq (1-100/M)^M|\mathcal{F}(Q_u)| \geq e^{-200}|\mathcal{F}(Q_u)|.$$

Likewise, any function f realizing any of the red branches $r_q^{(|u|+1)}$ for some $q \in [M-1]$ satisfies $f^{\mathsf{e2e}-M}(x^{(|u|+1)}) = 0$. So, it suffices to lower bound the number of such functions. Any red branch has a single minority edge, and there are $M-1$ red branches. Furthermore, a function can realize at most one branch, so the set of functions realizing different red branches are disjoint. Thus overall:

$$|\mathcal{F}(Q_{u0})| \geq (M-1)\frac{1}{100M}(1-100/M)^M|\mathcal{F}(Q_u)| \geq e^{-300}|\mathcal{F}(Q_u)|,$$

where the final inequality holds for large enough M. □

We are now ready to prove Lemma F.10.

Proof of Lemma F.10. We prove the lemma by showing that $|\mathcal{F}_(Q_u)| > 0$ for all u, which means that any labeling of S is realized by some function in $\mathcal{F}$. This establishes that $\mathrm{VC}(\mathcal{F}_{d,M})^{\mathsf{e2e},M} \geq m = \Omega(d\log M)$. The claim follows almost directly from Lemma F.12 by proving the following stronger claim:

$$|\mathcal{F}(Q_u)| \geq \frac{1}{e^{300|u|}}N \tag{12}$$

for all $u \in \{0,1\}^{\leq m}$. Indeed, having (12) we are done:

$$\frac{1}{e^{300|u|}}N \geq \frac{1}{e^{300\frac{d\log M}{600}}}M^{10d} \geq M^{9d}.$$

So, it remains to prove (12). We prove it by induction on $|u|$. To handle the base case $|u| = 0$, let us use the notation $\mathcal{F}(Q_0) = \mathcal{F}$. That is Q_0 means just $\mathcal{F}$ with no further restrictions. So indeed:

$$|\mathcal{F}(Q_u)| = |\mathcal{F}| = N = \frac{1}{e^{300\cdot 0}}N = \frac{1}{e^{300\cdot|u|}}N,$$

so the base case holds. For the induction step, Let $u' = ur$ where $|u'| \geq 0$ and $r \in \{0,1\}$. By induction hypothesis, we may assume that $|\mathcal{F}(Q_u)| \geq \frac{1}{e^{300|u|}}N$ So:

$$|\mathcal{F}(Q_{u'})| = |\mathcal{F}(Q_{ur})| \geq \frac{1}{e^{300}}\frac{1}{e^{300|u|}}N = \frac{1}{e^{300|u'|}}N$$

where the inequality is by the induction hypothesis and Lemma F.12. This concludes the proof. □

Recall that $\mathcal{F} := \mathcal{F}(d, M)$. That is, so far we assumed that the number of autoregressive iterations M is fixed. The only remaining step is to prove that we can construct from $\mathcal{F}(d, M)$ a single class that uniformly handles all M values, without significantly increasing it's Littlestone dimension. We prove this final step below.

F.2.6 A gluing lemma

We prove a *gluing lemma* that glues together the classes $\{\mathcal{F}_{d,M}\}_{M\in\mathbb{N}_+}$ without increasing the Littlestone dimension by more than 1. Our lemma is similar to the gluing lemma of Chase and Mehalel [2024].

Lemma F.13. *Let $c > 0$ be a universal constant, and fix $d \in \mathbb{N}$ larger than some universal constant. Let $f_d : \mathbb{N} \to \mathbb{N}$ be a function. Suppose that for all large enough M there exists a class $\mathcal{F}_{d,M}$ such that:*

1. $\mathtt{L}(\mathcal{F}_{d,M}) \leq c \cdot d$.
2. $\mathrm{VC}(\mathcal{F}_{d,M}^{\mathsf{e2e}-M}) \geq f_d(M)$
3. *Let*
$$X_M := \{x \in \{0,1\}^\star : \exists f \in \mathcal{F}_{d,M}, f(x) = 1\}.$$
Then $|X_M| \leq M^3$.

Then, there exists a class $\hat{\mathcal{F}}$ such that $\mathtt{L}(\hat{\mathcal{F}}) \leq c \cdot d + 1$ and for all large enough M it holds that $\mathrm{VC}(\mathcal{F}^{\mathsf{e2e}-M}) \geq f_d(M)$.

Proof. For every $M \geq 1$ and for every $f \in \mathcal{F}_{d,M}$, define the shifted function $\hat{f}$ of f as

$$\hat{f}(x') = \begin{cases} f(x) & \exists x \in \{0,1\}^\star : x' = 0^{2^{M^4}}x, \\ 0 & \text{otherwise,} \end{cases}$$

for all $x' \in \{0,1\}^\star$. Now, let

$$\hat{\mathcal{F}}_{d,M} := \{\hat{f} : f \in \mathcal{F}_{d,M}\}.$$

Denote the analog of X_M for $\hat{\mathcal{F}}_{d,M}$ as

$$\hat{X}_M := \{x \in \{0,1\}^\star : \exists \hat{f} \in \hat{\mathcal{F}}_{d,M}, \hat{f}(x) = 1\}.$$

We show that $M_1 > M_2 \implies \hat{X}_{M_1} \cap \hat{X}_{M_2} = \emptyset$. Indeed, the length of every instance in $\hat{X}_{M_2}$ is at most $2^{M_2^4} + M_2^3$. However, the length of every instance in $\hat{X}_{M_1}$ is at least

$$2^{M_1^4} \geq 2^{(M_2+1)^4} \geq 2^{M_2^4}2^{M_2^3} > 2^{M_2^4} + M_2^3,$$

and so indeed $\hat{X}_{M_1} \cap \hat{X}_{M_2} = \emptyset$. Now, define

$$\hat{\mathcal{F}} := \bigcup_M \hat{\mathcal{F}}_{d,M}.$$

where the union is taken over all large enough M. First, since $\hat{\mathcal{F}}$ contains $\hat{\mathcal{F}}_{d,M}$ for all large enough M, it is clear that for all large enough M:

$$\mathrm{VC}(\hat{\mathcal{F}}^{\mathsf{e2e}-M}) \geq \mathrm{VC}(\hat{\mathcal{F}}^{\mathsf{e2e}-M}_{d,M}) \geq f_d(M),$$

as desired. It remains to prove that $\mathtt{L}(\hat{\mathcal{F}}) \leq c \cdot d + 1$. To see why this is true, consider an online learner for $\hat{\mathcal{F}}$ that always predicts 0, until it makes it first mistake. Once it makes a mistake, it can infer that the target function gives 1 to the instance it erred for. Since $\hat{X}_{M_1} \cap \hat{X}_{M_2} = \emptyset$ for any two distinct M_1, M_2, this reveals the identity of the class $\hat{\mathcal{F}}_{d,M}$ from which the target function is taken. Now, the learner learns $\hat{\mathcal{F}}_{d,M}$, on which it will make at most $c \cdot d$ many mistakes by assumption. This concludes the proof. □

We may now prove the main theorem.

Proof of Theorem F.2. The theorem follows from combining Lemma F.10 and Lemma F.13. □

G Agnostic taxonomy

Consider the following known result.

Theorem G.1 (Ben-David et al. [2009], Alon et al. [2021])**.** *Let $\mathcal{F}$ be a binary function class.*[4] *Then the optimal regret bound for agnostic online learning of $\mathcal{F}$ is*

$$\Theta\Big(\sqrt{\mathtt{L}(\mathcal{F})T}\Big).$$

Using these bounds and our taxonomy of possible Littlestone dimension of $\mathcal{F}^{\mathsf{e2e}-M}$, we may infer an essentially complete taxonomy for agnostic learning as well. First, we deduce a general upper bound.

Corollary G.2. *Let $\mathcal{F}$ be a base class. Then for all sufficiently large M, there exists an agnostic online learner for $\mathcal{F}^{\mathsf{e2e}-M}$ with regret bound*

$$O\Big(\sqrt{T\mathtt{L}(\mathcal{F})\log M}\Big).$$

Below, we show that all sub-logarithmic well-behaved rates between $\sqrt{T}$ and $\sqrt{T\log M}$ are attainable. Note that due to Theorem G.1, going below order of $\sqrt{T}$ regret is possible only in the case that the autoregressive process degenerates $\mathcal{F}^{\mathsf{e2e}-M}$ to a single function. Therefore, this essentially establishes a complete taxonomy for agnostic online e2e-learning.

Corollary G.3. *For any well-behaved sub-logarithmic growth rate r there exists a base class $\mathcal{F}$ such that $\mathtt{L}(\mathcal{F}) = 1$ and the optimal regret bound for agnostic online learning $\mathcal{F}^{\mathsf{e2e}-M}$ is*

$$\Theta\Big(\sqrt{T \cdot r(M)}\Big)$$

for all $M \geq M_0$ and sufficiently large T.

A similar result holds also for logarithmic rates.

Corollary G.4. *For any $d \in \mathbb{N}_+$, there exists a class $\mathcal{F}$ such that the optimal regret for agnostic online learning of $\mathcal{F}^{\mathsf{e2e}-M}$ is*

$$\Theta\Big(\sqrt{Td\log M}\Big)$$

for all $M \geq M_0$ and sufficiently large T.

[4] The upper bound of Alon et al. [2021] requires that $\mathcal{F}$ satisfies some mild measurability constraints, which hold for all classes relevant to the scope of this paper, since our domain $\mathcal{X} = \{0,1\}^\star$ is countable.

H Learning with chain-of-thought supervision

In contrast to e2e-learning, we show that similarly to the statistical setting (see Hanneke et al. [2026]), the optimal mistake bound of learning a Littlestone class is independent of M.

Theorem H.1. *For every class $\mathcal{F}$, and for every M, we have $\mathtt{L}(\mathcal{F}^{\mathsf{CoT}-M}) \leq \mathtt{L}(\mathcal{F})$.*

Proof. We use an SOA-style learner who realizes the stated mistake bound, described below. The learner maintains a valid version space $V := V_t$ of $\mathcal{F}$, which in the first round is $V_1 = \mathcal{F}$. Let $x := x_t$ be the instance given by the adversary in round t. The learner constructs the generation tree $\mathbf{T}_M(x)$. For a branch b, let V_b be the version space V restricted to b. That is, those are all functions in V that agree with the branch b. We argue that there is at most one branch $b^\star$ so that $\mathtt{L}(V_{b^\star}) = \mathtt{L}(V)$. Indeed, suppose that there are two such branches b_0, b_1. Let v be the internal node where these branches first split, and let v_0, v_1 be the left and right children of v, where b_0 goes through v_0, and b_1 goes through v_1. So, the version space V restricted to both b_0 up to v_0 and b_1 up to v_1 has Littlestone dimension $\mathtt{L}(V)$. Therefore, we can construct a tree of depth $\mathtt{L}(V)+1$ which is rooted at v and is shattered by V, which is a contradiction. So, we determine the learner's prediction in round t to be the label associated with the edges of the branch

$$b^\star := \operatorname*{arg\,max}_{b \in B(\mathbf{T}_M(x))} \mathtt{L}(V_b).$$

Now, if the learner errs and the correct label is $b \neq b^\star$, then the learner updates $V_{t+1} = V_b$, and $\mathtt{L}(V_{t+1}) < \mathtt{L}(V_t)$. Telescoping over all rounds where the learner errs, at most $\mathtt{L}(\mathcal{F})$ many mistakes are made before $\mathtt{L}(V_t) = 0$, and then a single consistent function remains, and the learner's predictions are trivially given by this function from this round. □

Note that in contrast with the e2e setting, here we cannot infer an agnostic regret bound from the realizable mistake bound. Indeed, since the learner of Theorem H.1 learns the entire CoT, every change in the CoT counts as a mistake. Thus, it is possible that for some function in $\mathcal{F}$ the e2e-loss is extremely good, while for all functions in $\mathcal{F}$ the CoT-loss is very bad. In such case, the learner of Theorem H.1 is measured with respect to the function with the best CoT-loss, its regret might be very bad when measured against the function with the best e2e-loss. It is an interesting direction for future research to prove that this obstacle can or cannot be circumvented by some learner.

Using an online-to-batch conversion (Theorem D.8), Theorem H.1 immediately allows to derive a PAC sample complexity bound for learning $\mathcal{F}^{\mathsf{CoT}-M}$.

Theorem H.2 (Upper bound in terms of $\mathtt{L}(\mathcal{F})$)**.** *For every base class $\mathcal{F} \subset \Sigma^{\Sigma^\star}$ (even if $|\Sigma| > 2$, and even for $|\Sigma| = \infty$), and for all M, if $\mathtt{L}(\mathcal{F}) < \infty$ then $\mathcal{F}^{\mathsf{CoT}-M}$ is learnable with sample complexity*

$$m(\varepsilon, \delta) = O\left(\frac{\mathtt{L}(\mathcal{F}) + \log(1/\delta)}{\varepsilon}\right).$$

I Autoregressive Linear classifiers

In this section, we study the base class of autoregressive linear classifiers in $\mathbb{R}^d$. It is defined to only take into account the final d entries. Formally, define

$$\mathcal{F}_{d,\mathrm{lin}} := \{f_{\boldsymbol{w},b} : \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}\}$$

where the function $f_{\boldsymbol{w},b}$ is given as

$$f_{\boldsymbol{w},b}(x) = 1\left[\sum_{i=1}^{\min\{d,|\boldsymbol{x}|\}} \boldsymbol{w}[-i]x[-i] + b \geq 0\right]$$

for all $x \in \Sigma^\star$.

I.1 Mistake and sample complexity bounds

$\mathcal{F}_{d,\mathrm{lin}}$ is essentially the class of linear classifiers over the d-cube. This immediately gives the following bound

Proposition I.1. *For all M:*

$$\mathtt{L}(\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}) = O(d^2)$$

Proof. Trivial: The argument of [Joshi et al., 2025, Lemma 4.1] shows that $|\mathcal{F}_{d,\mathrm{lin}}| \leq 2^{O(d^2)}$. Since $|\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}| \leq |\mathcal{F}_{d,\mathrm{lin}}|$, the standard halving algorithm implies the stated bound. □

The bound above is based on a simple size argument. It is natural to ask if it is tight. Below, we give a positive answer to this question, that holds even for CoT-learning.

The idea is to implement a "latch" mechanism (like in digital systems) that "drags" the output bit of the base function throughout the entire CoT, and then just apply the best known bound for $\mathcal{F}_{d,\mathrm{lin}}$.

Theorem I.2. *There exists an adversary forcing $\Omega(d^2)$ on any learner for $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$ for all $d \geq 3$ and $M \geq 2$, even with CoT-supervision.*

The key is the following lemma, that provides the latch mechanism, in the price of reducing the dimension by 2.

Lemma I.3 (Latch lemma)**.** *Fix $d \geq 3$ and denote $m := d - 2$. Then, for all $\boldsymbol{v} \in \mathbb{R}^m$ and $c \in \mathbb{R}$, there exists $f_{\boldsymbol{w},b} \in \mathcal{F}_{d,\mathrm{lin}}$ such that*

$$f_{\boldsymbol{w},b}^{\mathsf{e2e}-M}(z10) = f_{\boldsymbol{v},c}(z)$$

for all $z \in \{0,1\}^m$ and all $M \geq 2$.

Proof. Define

$$L := \min_{z\in\{0,1\}^m} \langle \boldsymbol{v}, z\rangle + c, \quad U := \max_{z\in\{0,1\}^m} \langle \boldsymbol{v}, z\rangle + c$$

and let $B := \max\{0, U\} + 1, A := B - L + 1$. We now define $f_{\boldsymbol{w},b} \in \mathcal{F}_{d,\mathrm{lin}}$ that realizes the theorem. Let $s(x) := (u, p, q)$ be the last d bits of an input $x \in \{0,1\}^\star$ for $f_{\boldsymbol{w},b}$, where $|u| = d - 2, |p| = |q| = 1$. Let

$$\boldsymbol{w} := (\boldsymbol{v}, B, 2A), \quad b := c - B.$$

Therefore, for any input x of length at least d we have:

$$f_{\boldsymbol{w},b}(x) = 1[\langle \boldsymbol{v}, u\rangle + Bp + 2Aq + c - B \geq 0].$$

Now, we show that $f_{\boldsymbol{w},b}$ satisfies three desired properties:

1. For all $z \in \{0,1\}^m$: $f_{\boldsymbol{w},b}(z10) = f_{\boldsymbol{v},c}(z)$ (first bit of CoT is correct).
2. For any $x \in \{0,1\}^\star$ with $p = q = 0$, we have $f_{\boldsymbol{w},b}(x) = 0$ (always drag 0).
3. For any $x \in \{0,1\}^\star$ with $q = 1$, we have $f_{\boldsymbol{w},b}(x) = 1$ (always drag 1).

The three items establish the statement of the lemma: The first item states that $f_{\boldsymbol{w},b}(z10)$ returns the "correct" bit $f_{\boldsymbol{v},c}(z)$. Then, since $f_{\boldsymbol{w},b}^{\mathsf{e2e}-M}$ appends the returned correct bit, precisely one of the following two states is reached: if $f_{\boldsymbol{v},c}(z) = 0$, then the last two bits become 0, and the second item establishes that 0 is dragged through all M iterations. Otherwise, the last bit becomes 1, and then the third item guarantees that 1 is dragged through all M iterations. So it remains to prove the three items.

First item. Let $z \in \{0,1\}^m$, and denote $x = z10$. So

$$\langle \boldsymbol{w}, s(x)\rangle + b = \langle \boldsymbol{v}, z\rangle + B + c - B = \langle \boldsymbol{v}, z\rangle + c.$$

Therefore,

$$f_{\boldsymbol{w},b}(x) = 1[\langle \boldsymbol{w}, s(x)\rangle + b \geq 0] = 1[\langle \boldsymbol{v}, z\rangle + c \geq 0] = f_{\boldsymbol{v},c}(z).$$

Second item. Let x such that $p = q = 0$. Then

$$\langle \boldsymbol{w}, s(x) \rangle + b = \langle \boldsymbol{v}, u \rangle + c - B \leq U - B < 0,$$

where the first inequality is by definition of U, and the second inequality is by definition of B. Therefore, $f_{\boldsymbol{w},b}(x) = 0$ by definition.

Third item. Let x such that $q = 1$. Then

$$\begin{aligned}
\langle \boldsymbol{w}, s(x) \rangle + b &= \langle \boldsymbol{v}, u \rangle + Bp + 2A + c - B \\
&\geq \langle \boldsymbol{v}, u \rangle + c + 2A - B \\
&\geq L + 2A - B \\
&= L + 2(B - L + 1) - B \\
&= B - L + 2 \\
&> 0.
\end{aligned}$$

The equalities are by definitions. The first inequality is since $B > 0$ and thus $Bp \geq 0$. The second inequality is by definition of L. The last inequality is since $B > U \geq L$. So, we have $f_{\boldsymbol{w},b}(x) = 1$ by definition. □

We may now prove Theorem I.2.

Proof of Theorem I.2. By [Feldman and Xiao, 2014, Theorem B.2], we have $\mathtt{L}(\mathcal{F}_{m,\mathrm{lin}}) = \Theta(m^2)$ for all $m \geq 1$. As the latch lemma shows, for every function $f_m \in \mathcal{F}_{m,\mathrm{lin}}$ there is a function $f_d \in \mathcal{F}_{d,\mathrm{lin}}$ where $d = m + 2$, such that for all $z \in \{0,1\}^m$ and all $M \geq 1$ we have $f_d^{\mathsf{e2e}-M}(z10) = f_m(z)$. Therefore, the same adversarial strategy realizing the $\Omega(m^2)$ lower bound of Feldman and Xiao [2014] can be implemented by $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$, replacing any instance $z \in \{0,1\}^m$ with $z10$, and any $f_m \in \mathcal{F}_{m,\mathrm{lin}}$ with the appropriate $f_d \in \mathcal{F}_{d,\mathrm{lin}}$. Therefore,

$$\mathtt{L}(\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}) = \Theta(m^2) = \Theta(d^2)$$

for all M. To see that the same lower bound holds even with CoT-supervision, note that the CoT is just M copies of the same bit, and so CoT-supervision is equivalent to only receiving the e2e output. □

The same technique allows to prove a lower bound for the statistical setting as well.

Theorem I.4. *For all $d \geq 3$ and $M \geq 2$, any PAC-learner for $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$ has sample complexity $\Omega(d)$, even with* CoT*-supervision.*

Proof. We prove that $\mathrm{VC}(\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}}) \geq m$ for $m := d - 2$. Since all functions we will use to shatter a set of size m are implemented by the latch mechanism of Lemma I.3, those functions always copy the same final bit, and thus CoT-supervision is equivalent to only receiving the e2e output. Therefore, proving $\mathrm{VC}(\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}}) \geq m$ suffices for showing that m is a lower bound on the sample complexity, even with CoT-supervision. The proof now follows essentially the same lines as the proof of Theorem I.2, only replacing the bound $\mathtt{L}(\mathcal{F}_{m,\mathrm{lin}}) = \Theta(m^2)$ used in the proof of Theorem I.2 to the known bound $\mathrm{VC}(\mathcal{F}_{m,\mathrm{lin}}) \geq m$. □

I.2 Bounds for Efficient learning

We first show that efficient online e2e-learning of autoregressive linear thresholds is impossible, conditioned on the following assumption; we refer the reader to, e.g., Joshi et al. [2025] for more details.

Assumption I.5 (Hardness of Learning TC^0)**.** *There exist a constant $L > 0$ and a polynomial $p(n)$ such that threshold circuits of depth L and size $p(n)$ over n binary inputs are not weakly PAC-learnable in $\mathrm{poly}(n)$ time, i.e., for some constants $\varepsilon < \frac{1}{2}$ and $\delta < 1$, there is no polynomial-time learner that achieves error at most ε with confidence at least $1 - \delta$.*

Based on the above assumption, we show that learning end-to-end autoregressive linear classifiers is impossible. The proof is based on a standard batch-to-online argument and using Theorem 4.4 in Joshi et al. [2025].

Theorem I.6 (Conditional hardness of efficient online e2e-learning)**.** *Assume Hardness Assumption I.5. Then there is no possibly improper online learner such that, for every d, M, n, on every realizable online sequence of examples from $\{0,1\}^n$ labeled by some target in $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$, the learner runs in time polynomial in d, M, n per round and makes at most polynomially many mistakes in d, M, n.*

Proof. Suppose such a learner exists. By the standard reduction from realizable online learning with polynomial mistake bound to realizable PAC learning, this yields a polynomial-time realizable PAC learner for $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$ over $\{0,1\}^n$. This contradicts [Joshi et al., 2025, Theorem 4.4]. □

In contrast, with CoT-supervision there is an efficient online learner matching the information-theoretic guarantee up to a logarithmic factor. We use the classical deterministic online learner for Boolean linear thresholds. We first give a reduction

Lemma I.7 (Online reduction with CoT-supervision)**.** *Let $\mathcal{F} \subseteq \{f : \{0,1\}^\star \to \{0,1\}\}$ be a base class. Assume there is a deterministic online learner $\mathcal{A}$ for $\mathcal{F}$ such that:*

1. *on every realizable sequence it makes at most K mistakes;*
2. *on every length-L sequence, its total computation time is polynomial in L and the total input length.*

Then for every $M \geq 1$ there is a deterministic online learner $\mathcal{B}$ for $\mathcal{F}^{\mathsf{e2e}-M}$ with CoT-supervision such that:

1. *on every realizable CoT-supervised sequence it makes at most K mistakes;*
2. *on every length-n sequence, its total computation time is polynomial in n, M, and the total input length.*

Proof. Fix a realizable CoT-supervised sequence $(x_t, z_t)_{t=1}^n$, where $z_t = (z_{t,1}, \ldots, z_{t,M}) = f_\star^{\mathsf{CoT}-M}(x_t)$ for some $f_\star \in \mathcal{F}$. For each round s, let $u_{s,0} := x_s$, and for $j \in [M]$ let $u_{s,j} := u_{s,j-1} z_{s,j}$, where for a binary string u and a bit $b \in \{0,1\}$, ub denotes the string obtained by appending b to u. Learner $\mathcal{B}$ maintains the ordered transcript

$$S_{t-1} := \big((u_{1,0}, z_{1,1}), \ldots, (u_{1,M-1}, z_{1,M}), \ldots, (u_{t-1,0}, z_{t-1,1}), \ldots, (u_{t-1,M-1}, z_{t-1,M})\big).$$

On round t, learner $\mathcal{B}$ sets $\hat{u}_{t,0} := x_t$. For $j = 1, \ldots, M$, suppose $\hat{z}_{t,1}, \ldots, \hat{z}_{t,j-1}$ have already been defined and $\hat{u}_{t,j-1}$ has been formed. Learner $\mathcal{B}$ runs $\mathcal{A}$ from scratch on the sequence

$$S_{t-1}, (\hat{u}_{t,0}, \hat{z}_{t,1}), \ldots, (\hat{u}_{t,j-2}, \hat{z}_{t,j-1}),$$

where for $j = 1$ this means just S_{t-1}, and lets $\hat{z}_{t,j}$ be the prediction on the next example $\hat{u}_{t,j-1}$; then it sets $\hat{u}_{t,j} := \hat{u}_{t,j-1} \hat{z}_{t,j}$. Finally, it predicts $\hat{z}_{t,M}$. After receiving the true chain-of-thought $z_t = (z_{t,1}, \ldots, z_{t,M})$, it updates the transcript to

$$S_t := \big(S_{t-1}, (u_{t,0}, z_{t,1}), \ldots, (u_{t,M-1}, z_{t,M})\big).$$

Now consider the expanded sequence

$$((u_{t,j-1}, z_{t,j}))_{1 \leq t \leq n,\ 1 \leq j \leq M}.$$

It is realizable by $f_\star$, since $z_{t,j} = f_\star(u_{t,j-1})$ for all t, j. If $\mathcal{B}$ makes a mistake on round t, let j be the first index with $\hat{z}_{t,j} \neq z_{t,j}$. Then $\hat{u}_{t,i} = u_{t,i}$ for all $i < j$, so the transcript used by $\mathcal{B}$ to compute $\hat{z}_{t,j}$ is exactly the prefix of the expanded realizable sequence seen by $\mathcal{A}$ just before processing $(u_{t,j-1}, z_{t,j})$. Hence $\mathcal{A}$ makes a mistake on that example. Therefore every final-answer mistake of $\mathcal{B}$ is charged to a distinct mistake of $\mathcal{A}$, and so $\mathcal{B}$ makes at most K mistakes.

Finally, the expanded sequence has length at most nM, and its total input length is

$$\sum_{t=1}^{n}\sum_{j=1}^{M}|u_{t,j-1}| = \sum_{t=1}^{n}\sum_{j=1}^{M}(|x_t|+j-1) \le M\sum_{t=1}^{n}|x_t| + nM^2,$$

which is polynomial in n, M, and the total input length of the original sequence. The same bound holds for every simulated prefix sequence used by $\mathcal{B}$, since $|\hat{u}_{t,j}| = |x_t| + j$ for all t, j. Since $\mathcal{B}$ performs at most nM simulations, and each simulation runs $\mathcal{A}$ from scratch on a sequence of length at most nM and polynomially bounded total input length, the total computation time of $\mathcal{B}$ is polynomial in n, M, and the total input length of the original sequence. □

To obtain our efficient algorithm, we use the following classic result for the base class.

Lemma I.8 (Maass–Turán Maass and Turán [1994])**.** *There exists a deterministic online learner for Boolean linear thresholds over $\{0,1\}^d$ such that:*

1. *on every realizable sequence it makes at most $O(d^2 \log d)$ mistakes;*
2. *on every length-m sequence, its total computation time is polynomial in m and d.*

Theorem I.9 (Efficient online CoT-learning for autoregressive linear thresholds)**.** *Fix a horizon n. There exists a deterministic online learner for $\mathcal{F}_{d,\mathrm{lin}}^{\mathsf{e2e}-M}$ with CoT-supervision such that, for every sequence $(x_t, z_t)_{t=1}^n$ satisfying $z_t = f_\star^{\mathsf{CoT}-M}(x_t)$ for some $f_\star \in \mathcal{F}_{d,\mathrm{lin}}$, the learner makes at most $O(d^2 \log d)$ mistakes and its total computation time is polynomial in n, d, M, and $\sum_{t=1}^n |x_t|$.*

Proof. Let $\phi : \{0,1\}^\star \to \{0,1\}^d$ be the suffix map returning the last d bits, padded with leading zeroes if needed. By definition of $\mathcal{F}_{d,\mathrm{lin}}$, for every $f \in \mathcal{F}_{d,\mathrm{lin}}$ there is a Boolean linear threshold $g_f : \{0,1\}^d \to \{0,1\}$ such that $f(x) = g_f(\phi(x))$ for all $x \in \{0,1\}^\star$.

Let $\mathcal{A}$ be the learner from Lemma I.8. We define a learner $\mathcal{A}'$ for $\mathcal{F}_{d,\mathrm{lin}}$ as follows: on an example x, learner $\mathcal{A}'$ predicts whatever $\mathcal{A}$ predicts on $\phi(x)$; after receiving the label y, learner $\mathcal{A}'$ updates $\mathcal{A}$ on the labeled example $(\phi(x), y)$. Since $f(x) = g_f(\phi(x))$, every realizable sequence for $\mathcal{F}_{d,\mathrm{lin}}$ maps to a realizable sequence for Boolean linear thresholds over $\{0,1\}^d$. Hence $\mathcal{A}'$ makes at most $O(d^2 \log d)$ mistakes on every realizable sequence. Its total computation time on any length-m sequence is polynomial in m, d, and the total input length, since computing all suffixes $\phi(x)$ contributes only linear overhead.

Applying Lemma I.7 to the base class $\mathcal{F}_{d,\mathrm{lin}}$ and the learner $\mathcal{A}'$ yields the claim. □

J Non-Littlestone base classes

Theorem J.1. *There exists a base class $\mathcal{F}$ over the binary alphabet $\Sigma = \{0,1\}$ such that, for the pairwise distinct sequence*

$$M_i := i + 4 \qquad (i \in \mathbb{N}),$$

the following holds:

$$i \text{ odd} \implies \mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M_i}) = 0, \qquad i \text{ even} \implies \mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M_i}) = \infty.$$

Moreover, for every odd i, online learning with no CoT-supervision is trivial, while for every even i, online learning with CoT-supervision is impossible.

Proof. For every $m \ge 2$ and $n \ge 0$, define the string

$$u_{m,n} := 0^m 1 0^n 1.$$

For every binary array $\alpha = (\alpha_{m,n})_{m\ge 2,\, n\ge 0} \in \{0,1\}^{\{2,3,\ldots\}\times(\mathbb{N}\cup\{0\})}$, define a next-token generator

$$f_\alpha : \{0,1\}^\star \to \{0,1\}$$

as follows:

$$f_\alpha(x) = \begin{cases} \alpha_{m,n} & \text{if } x = u_{m,n} \text{ for some } m \geq 2,\ n \geq 0, \\ 0 & \text{if } x = u_{m,n} b 0^t \text{ for some } m \geq 2,\ n \geq 0,\ b \in \{0,1\},\ 0 \leq t < 2m-2, \\ b & \text{if } x = u_{m,n} b 0^{2m-2} \text{ for some } m \geq 2,\ n \geq 0,\ b \in \{0,1\}, \\ 0 & \text{otherwise.} \end{cases}$$

Let

$$\mathcal{F} := \{f_\alpha : \alpha \in \{0,1\}^{\{2,3,\ldots\} \times (\mathbb{N} \cup \{0\})}\}.$$

We first analyze the even horizons. Fix $m \geq 2$, $n \geq 0$, and α. Let $b := \alpha_{m,n}$. Starting from the input $u_{m,n}$, the generated bits are

$$b,\ \underbrace{0, \ldots, 0}_{2m-2 \text{ times}}, b.$$

Therefore,

$$f_\alpha^{\mathsf{e2e}-2m}(u_{m,n}) = \alpha_{m,n}.$$

Hence, for every fixed $m \geq 2$, the set

$$X_m := \{u_{m,n} : n \geq 0\}$$

shows that $\mathcal{F}^{\mathsf{e2e}-2m}$ has infinite Littlestone dimension. Indeed, let $d \in \mathbb{N}$, and consider the complete binary tree of depth d in which every node at depth j is labeled by $u_{m,j-1}$. Given any branch-label sequence $y_1, \ldots, y_d \in \{0,1\}$, choose α so that

$$\alpha_{m,j-1} = y_j \qquad \text{for all } j = 1, \ldots, d,$$

and define the remaining coordinates of α arbitrarily. Then

$$f_\alpha^{\mathsf{e2e}-2m}(u_{m,j-1}) = y_j \qquad \text{for all } j = 1, \ldots, d.$$

Thus this tree is shattered by $\mathcal{F}^{\mathsf{e2e}-2m}$. Since d is arbitrary,

$$\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-2m}) = \infty \qquad \text{for every } m \geq 2.$$

Moreover, this lower bound is unchanged under CoT-supervision. Indeed, for every queried instance $u_{m,n}$, if $b := \alpha_{m,n}$, then the revealed $2m$-step chain is

$$f_\alpha^{\mathsf{CoT}-2m}(u_{m,n}) = b 0^{2m-2} b.$$

In particular, the revealed CoT is completely determined by the final label

$$f_\alpha^{\mathsf{e2e}-2m}(u_{m,n}) = b.$$

Hence, on the adversarial sequence above, a learner with CoT-supervision receives no more information than an e2e-learner, so the same lower bound applies.

We now turn to odd horizons. Fix an odd $M \geq 5$. We claim that all functions in $\mathcal{F}^{\mathsf{e2e}-M}$ coincide, and therefore $\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = 0$.

Fix $x \in \{0,1\}^\star$ and two arrays α, β. We show that

$$f_\alpha^{\mathsf{e2e}-M}(x) = f_\beta^{\mathsf{e2e}-M}(x).$$

We will use the following simple observation: for every $m \geq 2$, $n \geq 0$, $b \in \{0,1\}$, and $r \geq 0$, the string

$$u_{m,n} b 0^{2m-2} b 0^r$$

is of none of the special forms appearing in the definition of f_α. Indeed, if it were of the form $u_{m',n'}$ or $u_{m',n'} c 0^t$, then necessarily $m' = m$ and $n' = n$, since the first two occurrences of 1 already determine the prefix $u_{m,n}$. But then the remaining suffix $b 0^{2m-2} b 0^r$ cannot be empty, and also cannot equal $c 0^t$: if $b = 1$ then it contains two 1's, while if $b = 0$ it equals 0^{2m+r}, which would force $c = 0$ and $t = 2m + r - 1 > 2m - 2$. Thus no such string is special.

We now distinguish three cases.

Case 1: $x = u_{m,n}$ for some $m \geq 2$, $n \geq 0$. Let $b := \alpha_{m,n}$. Under f_α, the first generated bit is b, the next $2m-2$ generated bits are 0, and the $2m$-th generated bit is again b. After these $2m$ steps, the current string is

$$u_{m,n} b 0^{2m-2} b.$$

By the observation above, from this point on, only the default rule applies, so all subsequent generated bits are 0. Therefore the only possibly α-dependent generated bits occur at steps 1 and $2m$. Since $M \geq 5$ is odd it holds that $M \neq 1$ and $M \neq 2m$, and hence $f_\alpha^{\mathsf{e2e}-M}(x)$ is independent of α. In particular,

$$f_\alpha^{\mathsf{e2e}-M}(x) = f_\beta^{\mathsf{e2e}-M}(x).$$

Case 2: $x = u_{m,n} b 0^t$ for some $m \geq 2$, $n \geq 0$, $b \in \{0,1\}$, $0 \leq t \leq 2m-2$. Starting from x, the next $2m-2-t$ generated bits are 0, then the next generated bit is b, and after that the current string has the form

$$u_{m,n} b 0^{2m-2} b.$$

By the observation above, from this point on, all subsequent generated bits are 0. Therefore, the entire future trajectory from x is determined by the string x itself, and in particular is independent of α. Hence

$$f_\alpha^{\mathsf{e2e}-M}(x) = f_\beta^{\mathsf{e2e}-M}(x).$$

Case 3: x is of neither of the above two forms. Then $f_\alpha(x) = f_\beta(x) = 0$. As long as the autoregressive trajectory stays outside the set

$$\{u_{m,n} : m \geq 2,\ n \geq 0\} \cup \{u_{m,n} b 0^t : m \geq 2,\ n \geq 0,\ b \in \{0,1\},\ 0 \leq t \leq 2m-2\},$$

both functions continue to output 0. If the trajectory never enters this set, then clearly

$$f_\alpha^{\mathsf{e2e}-M}(x) = f_\beta^{\mathsf{e2e}-M}(x) = 0.$$

Otherwise, let $s \geq 1$ be the first time such a string is reached. Since all bits generated before time s are 0, the reached string cannot be of the form $u_{m,n}$, because every $u_{m,n}$ ends with 1. Hence at time s the trajectory reaches a string covered by Case 2, and from that point on the continuation is independent of α. Therefore

$$f_\alpha^{\mathsf{e2e}-M}(x) = f_\beta^{\mathsf{e2e}-M}(x).$$

We have proved that for every odd $M \geq 5$, all functions in $\mathcal{F}^{\mathsf{e2e}-M}$ are identical. Hence

$$|\mathcal{F}^{\mathsf{e2e}-M}| = 1, \qquad \text{so} \qquad \mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M}) = 0.$$

Therefore, for every odd $M \geq 5$, online learning with CoT-supervision is also possible, since CoT-supervision is only more informative than $\mathsf{e2e}$-supervision.

Finally, for the pairwise distinct sequence $M_i = i + 4$, if i is even then M_i is even and at least 6, so

$$\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M_i}) = \infty,$$

and moreover, the same impossibility holds even for online learning with CoT-supervision. If i is odd then M_i is odd and at least 5, so

$$\mathtt{L}(\mathcal{F}^{\mathsf{e2e}-M_i}) = 0 < \infty,$$

and therefore online learning with CoT-supervision is also possible. □

K Stochastic Autoregressive Lower Bound

We now consider a stochastic variant of the autoregressive model.

Definition K.1 (Stochastic autoregressive next-token generators)**.** A *stochastic base class* is a class

$$\mathcal{G} \subseteq \{g : \{0,1\}^\star \to \Delta(\{0,1\})\},$$

where $\Delta(\{0,1\})$ denotes the set of distributions over $\{0,1\}$. For $g \in \mathcal{G}$ and $x \in \{0,1\}^\star$, define $\bar{g}(x)$ to be the random string obtained by sampling $Y \sim g(x)$ and outputting $\mathrm{append}(x, Y)$. For $M \in \mathbb{N}$, define $g^{\mathsf{CoT}-M}$ and $g^{\mathsf{e2e}-M}$ exactly as in the deterministic case, except that these are now induced distributions obtained by composing $\bar{g}$ for M many steps. As before, write $\mathcal{G}^{\mathsf{e2e}-M} := \{g^{\mathsf{e2e}-M} : g \in \mathcal{G}\}$.

In the associated online prediction problem, on round t the learner receives an instance $x_t \in \{0,1\}^\star$, predicts a bit $\hat{y}_t \in \{0,1\}$, and then observes a random label Y_t, sampled from the relevant target distribution. In the direct next-token problem we have $Y_t \sim g_\star(x_t)$, while in the e2e problem we have $Y_t \sim g_\star^{\mathsf{e2e}-M}(x_t)$, for some unknown $g_\star \in \mathcal{G}$. For a learner $\mathcal{A}$, target $g_\star$, and instance sequence $(x_t)_{t=1}^T$, define the expected regret relative to the optimal prediction

$$\mathrm{Reg}_T(\mathcal{A}, g_\star, (x_t)) := \mathbb{E}\left[\sum_{t=1}^T \mathbb{1}[\hat{y}_t \neq Y_t]\right] - \sum_{t=1}^T \min_{a\in\{0,1\}} \Pr[a \neq Y_t].$$

Theorem K.2. *For every odd $M \geq 1$ there exists a stochastic base class $\mathcal{G} = \{g_-, g_+\}$ such that:*

1. *there exists an online learner for the direct next-token problem on $\mathcal{G}$ whose expected regret is $O(1)$ on every instance sequence;*
2. *for every (possibly randomized) online learner $\mathcal{A}$ for $\mathcal{G}^{\mathsf{e2e}-M}$, there exists a target $g_\star \in \mathcal{G}$ such that, even on the constant sequence $x_t = \emptyset$, its expected regret is $\Omega(2^M)$ at some horizon $T = \Theta(4^M)$.*

In particular, the e2e regret can be $\Omega\left(2^M\right)$, much larger than $\log M$ which is the upper bound growth of e2e-learning with a fixed Littlestone dimension base class (given in Theorem E.1) in the standard (deterministic functions) model.

Proof. For $x \in \{0,1\}^\star$, let $b(x)$ denote the last bit of x, with the convention $b(\emptyset) = 0$. For $\sigma \in \{-1,+1\}$ define

$$g_\sigma(x) = \text{law of } b(x) \oplus Z, \qquad Z \sim \mathrm{Ber}\left(\frac{1}{2} + \frac{\sigma}{4}\right).$$

Thus g_- flips the current last bit with probability $1/4$, whereas g_+ flips it with probability $3/4$.

We first prove the first item in the theorem, that learning the direct next token is trivial. Fix a target g_σ and an arbitrary instance sequence $(x_t)_{t=1}^T$. Define $Z_t := Y_t \oplus b(x_t)$. Then $Z_1, Z_2, \ldots$ are i.i.d. $\mathrm{Ber}(\frac{1}{2} + \frac{\sigma}{4})$, independently of the instance sequence. Consider the learner that predicts arbitrarily on round 1, and on round $t \geq 2$ computes the empirical mean $\bar{Z}_{t-1} := \frac{1}{t-1}\sum_{s=1}^{t-1} Z_s$, sets $\hat{\sigma}_t = +1$ if $\bar{Z}_{t-1} \geq \frac{1}{2}$ and $\hat{\sigma}_t = -1$ otherwise, and then predicts $b(x_t)$ if $\hat{\sigma}_t = -1$ and $1 - b(x_t)$ if $\hat{\sigma}_t = +1$. If $\hat{\sigma}_t = \sigma$, this is the Bayes-optimal prediction for $g_\sigma(x_t)$; otherwise the regret at round t is exactly $3/4 - 1/4 = 1/2$. Hence,

$$\mathrm{Reg}_T \leq \frac{1}{2} + \frac{1}{2}\sum_{t=2}^T \Pr[\hat{\sigma}_t \neq \sigma].$$

Since the true mean is either $1/4$ or $3/4$, Hoeffding's inequality gives $\Pr[\hat{\sigma}_t \neq \sigma] \leq e^{-(t-1)/8}$, and therefore $\mathrm{Reg}_T = O(1)$.

We now prove the second item. If $M = 1$, then $g_\sigma^{\mathsf{e2e}-1} = g_\sigma$. On the constant sequence $x_t = \emptyset$, the two targets induce $\mathrm{Ber}(1/4)$ and $\mathrm{Ber}(3/4)$. On round 1, if the learner predicts 1 with probability p, then its expected regret is $p/2$ under g_- and $(1-p)/2$ under g_+. Thus one of the two targets yields regret at least $1/4$ at horizon $T = 1$, which is $\Omega(2^M)$ for $M = 1$. So it remains to consider odd $M \geq 3$.

Fix such an odd M, and again consider the constant sequence $x_t = \emptyset$. Let q_m^σ be the probability that after m autoregressive steps the current last bit equals 1 under g_σ, starting from $\emptyset$. The process is a two-state Markov chain, and

$$q_{m+1}^\sigma = \left(\frac{1}{2} + \frac{\sigma}{4}\right) + \left(1 - 2\left(\frac{1}{2} + \frac{\sigma}{4}\right)\right) q_m^\sigma, \qquad q_0^\sigma = 0.$$

Solving this recursion gives $q_m^\sigma = \frac{1-(-\sigma/2)^m}{2}$. Since M is odd, $q_M^\sigma = \frac{1}{2} + \sigma 2^{-M-1}$. Let $\delta_M := 2^{-M-1}$. Hence on input $\emptyset$, the final answer under $g_-^{\mathsf{e2e}-M}$ is $\mathrm{Ber}(\frac{1}{2} - \delta_M)$, while under $g_+^{\mathsf{e2e}-M}$ it is $\mathrm{Ber}(\frac{1}{2} + \delta_M)$. The Bayes-optimal predictions are therefore 0 and 1 respectively, and using the wrong one incurs regret $2\delta_M$ per round.

Fix any randomized learner $\mathcal{A}$. Its prior-averaged expected regret under the uniform prior on $\{g_-, g_+\}$ is the expectation, over the learner's internal randomness, of the corresponding quantity for the induced deterministic learner. It therefore suffices to prove the lower bound for deterministic learners.

Fix such a deterministic learner. On round t, its prediction is a function of the first $t-1$ observed labels. Let P_-^{t-1} and P_+^{t-1} denote the distributions of these $t-1$ labels under $\mathrm{Ber}(\frac{1}{2}-\delta_M)$ and $\mathrm{Ber}(\frac{1}{2}+\delta_M)$ respectively, and let A_t be the event that the learner predicts 1 on round t. Since the Bayes-optimal prediction is 0 under g_- and 1 under g_+, the prior-averaged probability of predicting the wrong Bayes-optimal bit at round t is

$$\frac{1}{2}P_-^{t-1}(A_t) + \frac{1}{2}P_+^{t-1}(A_t^c).$$

By the Bretagnolle–Huber inequality [Lattimore and Szepesvári, 2020, Theorem 14.2],

$$P_-^{t-1}(A_t) + P_+^{t-1}(A_t^c) \geq \frac{1}{2}\exp\bigl(-\mathrm{KL}(P_-^{t-1}, P_+^{t-1})\bigr),$$

where $\mathrm{KL}(P, Q)$ denotes the Kullback–Leibler divergence from P to Q. Therefore the prior-averaged probability of predicting the wrong Bayes-optimal bit is at least

$$\frac{1}{4}\exp\bigl(-\mathrm{KL}(P_-^{t-1}, P_+^{t-1})\bigr).$$

It remains to bound the divergence. Since the labels are i.i.d.,

$$\mathrm{KL}(P_-^{t-1}, P_+^{t-1}) = (t-1)\,\mathrm{KL}\left(\mathrm{Ber}\left(\frac{1}{2}-\delta_M\right), \mathrm{Ber}\left(\frac{1}{2}+\delta_M\right)\right).$$

Moreover,

$$\mathrm{KL}\left(\mathrm{Ber}\left(\frac{1}{2}-\delta\right), \mathrm{Ber}\left(\frac{1}{2}+\delta\right)\right) = 2\delta\log\left(\frac{1+2\delta}{1-2\delta}\right) \leq 16\delta^2 \qquad (\delta \leq 1/4),$$

where the inequality uses $\log\left(\frac{1+u}{1-u}\right) \leq 4u$ for $u \in [0, 1/2]$. Hence

$$\mathrm{KL}(P_-^{t-1}, P_+^{t-1}) \leq 16(t-1)\delta_M^2.$$

Thus, whenever $t \leq 1/(32\delta_M^2)$, we have $\mathrm{KL}(P_-^{t-1}, P_+^{t-1}) \leq 1/2$, and so the prior-averaged probability of predicting the wrong Bayes-optimal bit is at least $\frac{1}{4}e^{-1/2} = \Omega(1)$. Since each such mistake contributes regret $2\delta_M$, the prior-averaged regret at round t is $\Omega(\delta_M)$.

Setting $T := \left\lfloor \frac{1}{32\delta_M^2} \right\rfloor$, we have $T = \Theta(4^M)$. Summing over rounds $t = 1, \ldots, T$, the prior-averaged regret up to time T is $\Omega(T\delta_M) = \Omega(1/\delta_M) = \Omega(2^M)$. Therefore at least one of the two targets g_-, g_+ yields regret $\Omega(2^M)$ at horizon $T = \Theta(4^M)$. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: All claims are proved.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: All assumptions are declared.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: All assumptions are declared, all claims are proved.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [N/A]

   Justification: Theory paper.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [N/A]

Justification: Theory paper.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [N/A]

   Justification: Theory paper.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [N/A]

   Justification: Theory paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [N/A]

   Justification: Theory paper.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Theory paper.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: Theory paper.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Theory paper.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [N/A]

    Justification: Theory paper.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: Theory paper.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: Theory paper.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

    Justification: Theory paper.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: No core impact.

Guidelines:

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