Proceedings of Machine Learning Research vol 291:1–37, 2025 38th Annual Conference on Learning Theory

Generation through the lens of learning theory

Jiaxun Li[*] JASONLI@UMICH.EDU
Vinod Raman[*] VKRAMAN@UMICH.EDU
Ambuj Tewari TEWARIA@UMICH.EDU
Department of Statistics, University of Michigan

Editors: Nika Haghtalab and Ankur Moitra

Abstract

We study generation through the lens of learning theory. First, we formalize generation as a sequential two-player game between an adversary and a generator, which generalizes the notion of "language generation in the limit" from Kleinberg and Mullainathan (2024). Then, we extend the notion of "generation in the limit" to two new settings, which we call "uniform" and "non-uniform" generation. We provide a characterization of hypothesis classes that are uniformly and non-uniformly generatable. As is standard in learning theory, our characterizations are in terms of the finiteness of a new combinatorial dimension termed the Closure dimension. By doing so, we are able to compare generatability with predictability (captured via PAC and online learnability) and show that these two properties of hypothesis classes are *incomparable* – there are classes that are generatable but not predictable and vice versa. Finally, we extend our results to capture *prompted* generation and give a complete characterization of which classes are prompt generatable, generalizing some of the work by Kleinberg and Mullainathan (2024).

Keywords: Learning Theory, Generative Machine Learning

1. Introduction

Over the past 50 years, predictive machine learning has been a cornerstone for both theorists and practitioners. Predictive tasks like classification and regression have been extensively studied, in both theory and practice, due to their applications to face recognition, pedestrian detection, fraud detection, protein structure prediction, etc. Recently, however, a new paradigm of machine learning has emerged: *generation*. Unlike predictive models, which focus on making accurate predictions of the true label given examples, generative models aim to *create* new examples based on observed data. For example, in language modeling, the goal might be to generate coherent text in response to a prompt, while in drug development, one might want to generate candidate molecules. In fact, generative models have already been applied to these tasks and others (Zhao et al., 2023; Jumper et al., 2021).

The vast potential of generative machine learning has spurred a surge of research across diverse fields like natural language processing (Wolf et al., 2020), computer vision (Khan et al., 2022), and molecular sciences (Vanhaelen et al., 2020). Despite this widespread adoption, the learning-theoretic foundations of generative machine learning lags behind its predictive counterpart. While prediction has been extensively studied by learning theorists through frameworks like PAC and online learning (Shalev-Shwartz and Ben-David, 2014; Mohri et al., 2012; Cesa-Bianchi and Lugosi, 2006), generative machine learning has, for the most, part remained elusive. One reason for this is that generation is fundamentally an *unsupervised* task. Unlike classification or regression, where

[*] Equal contribution

there is a true label or response to guide the model, generation lacks a clear notion of correctness. This makes it challenging to define a loss function – the primary tool used in predictive tasks to quantify the quality of a model. Another reason is that existing theory in generative machine learning often places strong distributional/probabilistic assumptions on the data generation process (Murphy, 2023). Such assumptions are typically absent in the game-theoretic, worst-case flavor of learning theory in settings like PAC and online learning.

In light of this difficulty, we take a step back and view the task of generation in its simplest form – one sees a sequence of examples, and is tasked with producing new, valid examples. Using this perspective:

(1) We formalize generation as a sequential two-player game between an adversary and a generator. In this game, the adversary picks a binary hypothesis $h \in \mathcal{H}$ and an infinite sequence of positive examples $x_1, x_2, \dots$ such that $h(x_i) = 1$. In each round $t \in \mathbb{N}$, the adversary reveals x_t to the generator, who is then tasked with producing an example $\hat{x}_t$ such that $\hat{x}_t \notin \{x_1, \dots, x_t\}$ (i.e., $\hat{x}_t$ is new) and $h(\hat{x}_t) = 1$ (i.e., $\hat{x}_t$ is valid). The goal of the generator is to "eventually" produce new and valid examples in h. This formulation generalizes the notion of "language generation in the limit" from Kleinberg and Mullainathan (2024), enabling our results to apply to other generative tasks like image and molecule generation.

(2) We go beyond the notion of "generation in the limit" from Kleinberg and Mullainathan (2024) by introducing two stronger paradigms of generation called "uniform" and "non-uniform" generation. These models differ in how one quantifies the time after which the generator must perfectly produce new, valid examples. While Kleinberg and Mullainathan (2024) show that finite hypothesis classes are uniformly generatable, they leave the full characterization of uniform generatability open. We close this gap and provide a complete characterization of which hypothesis classes are uniformly generatable in terms of a new combinatorial dimension we call the Closure dimension.

Theorem *(Informal) A class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is* uniformly generatable *if and only if $\mathrm{C}(\mathcal{H}) < \infty$, where $\mathrm{C}(\mathcal{H})$ is the* Closure dimension *of $\mathcal{H}$.*

In addition, we use the Closure dimension to fully characterize which classes are non-uniformly generatable.

Theorem *(Informal) A class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is* non-uniformly generatable *if and only if there exists a non-decreasing sequence of classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots$ such that $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}_n$ and $\mathrm{C}(\mathcal{H}_n) < \infty$ for every $n \in \mathbb{N}$.*

In fact, while Kleinberg and Mullainathan (2024) show that all countable classes are generatable in the limit, our characterization of non-uniform generatability shows that all countable classes are non-uniformly generatable. This provides an improvement as non-uniform generation is a strictly harder than generation in the limit. With respect to generatability in the limit, we provide an alternate sufficiency condition in terms of the Closure dimension which, in conjunction with countableness, significantly expands the collection of classes that are known to be generatable in the limit.

Theorem *(Informal) A class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is* generatable in the limit *if there exists a finite sequence of classes $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_n$ such that $\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{H}_i$ and $\mathrm{C}(\mathcal{H}_i) < \infty$ for all $i \in [n]$.*

In addition to the above theorem, we give two other sufficiency conditions for generatability in the limit in terms of what we call the "Eventually Unbounded Closure" property. We leave the complete characterization of generatability in the limit as an important future question (see Section 5).

(3) We uncover fundamental differences between *generation* and *prediction* for countable classes, the latter of which we measure through PAC and online learnability. In particular, we find that these two tasks are incomparable – there exist hypothesis classes for which generation is possible but prediction is not, and vice versa.

(4) We extend our results to capture a notion of *prompted* generation, generalizing some of the results from Section 7 of Kleinberg and Mullainathan (2024). By extending the Closure dimension to the Prompted Closure dimension, we prove identical characterizations of prompted uniform and non-uniform generatability as in the informal theorems in Contribution (2).

Our results extend the study of generation beyond language modeling, but are mainly information-theoretic in nature. That said, for all our algorithms, we point out natural computational primitives that can make our algorithms computable.

1.1. Related Works

Language Identification and Generation. The literature on generative machine learning is too vast to be surveyed in complete detail. Thus, we refer the reader to the books and surveys by Jebara (2012), Harshvardhan et al. (2020), and Murphy (2023). The works most related to our work are those regarding language identification and generation in the limit (Gold, 1967; Angluin, 1979, 1980). Instead of an example space $\mathcal{X}$ and a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, these works consider a countable set U of strings and a countable language family $C = \{L_1, L_2, \dots\}$, where $L_i \subset U$ for all $i \in \mathbb{N}$.

In the Gold-Angluin model, an adversary first picks a language $K \in C$, and begins to enumerate the strings one by the one to the player in rounds $t = 1, 2, \dots$. After observing the string w_t in round $t \in \mathbb{N}$, the player guesses a language L_t in C with the hope that $L_t = K$. Crucially, the player gets no feedback at all. The player has identified K in the limit, if there exists a finite time step $t^\star \in \mathbb{N}$ such that for all $s \geq t$, we have that $L_s = K$.

In full generality, Gold (1967) showed that language identification in the limit is impossible – there are simple language families C, like those produced by finite automata, for which no algorithm can perform language identification in the limit. Following this work, Angluin (1979, 1980) provide a precise characterization of which language families C is language identification in the limit possible. This characterization further emphasized the impossibility of language identification in the limit by ruling out the vast majority of language families.

Very recently, and inspired by large language models, Kleinberg and Mullainathan (2024) study the problem of language *generation* in the limit. In this problem, the adversary also picks a language $K \in C$, and begins to enumerate the strings one by the one to the player in rounds $t = 1, 2, \dots$. However, now, after observing the string w_t in round $t \in \mathbb{N}$, the player guesses a string $\hat{w}_t \in U$ with the hope that $\hat{w}_t \in K \setminus \{w_1, \dots, w_t\}$. Once again, the player gets no feedback at all. The player has generated from K in the limit, if there exists a finite time step $t \in \mathbb{N}$ such that for all $s \geq t$, we have that $\hat{w}_s \in K \setminus \{w_1, \dots, w_s\}$. Remarkably, Kleinberg and Mullainathan (2024) prove a strikingly different result – while Gold-Angluin show that identification in the limit is impossible for most language families, Kleinberg and Mullainathan (2024) show that generation in the limit is possible for *every* countable language family C. This shows that language identification and generation are drastically different in the limit.

Concurrently and independently from our work, Kalavasis et al. (2024b) study generation in the stochastic setting, where the positive examples revealed to the generator are sampled i.i.d. from some unknown distribution. In this model, they study the trade-offs between generating with breadth and generating with consistency and resolve the open question posed by Kleinberg and Mullainathan (2024) for a large family of language models. In addition, Kalavasis et al. (2024b) quantify the error rates for generation with breadth/consistency according to the universal rates framework of Bousquet et al. (2021).

Language Generation with Breadth. One underlying problem in generation is the tension between "validity" (i.e., produce valid outputs without hallucinating) and "breadth"(i.e., provide outputs to capture the richness of the language) (Kleinberg and Mullainathan, 2024). Several follow-up studies have proposed and studied different definitions of breadth in the language generation model. Charikar and Pabbaraju (2024) introduce the notion of "exhaustive generation." Kalavasis et al. (2024b) and Kalavasis et al. (2024a) introduce three notions: "generation with exact breadth", "generation with approximate breadth" and "unambiguous generation." Although these notions differ slightly, they all essentially require the generator's outputs cover nearly, or exactly the entirely of the true language in the limit. The overall conclusion from these works is that this kind of breadth requires strong assumptions on the language family. In fact, Kalavasis et al. (2024a) shows that "generation with exact breadth" is even as hard as language identification (see Appendix A and B for precise definitions). In response to this hardness, Kleinberg and Wei (2025) consider a weaker notion of breadth that only requires the generator's output to cover a proportion of the chosen language. Here, they show that the tension between validity and breadth is not as severe.

Other Related Work. In a previous version of this work, we posed whether all countable class are non-uniformly generatable (see Definition 4) as an open question. In a follow-up work Charikar and Pabbaraju (2024), independently of us, resolve this affirmatively. In addition to this positive result, Charikar and Pabbaraju (2024) also show that non-uniform generation is not possible using only membership queries. This is in contrast to generatability in the limit, where Kleinberg and Mullainathan (2024) show that every countable classes is generatable in the limit using only membership queries. Finally, Charikar and Pabbaraju (2024) also characterize which classes are uniformly generatable when some feedback is available.

There are several subsequent works building on the present paper: Peale et al. (2025) introduce and study representative generation, in which the generator not only needs to generate new, valid instances, but also needs to output a distribution that is representative of the data seen so far; Raman and Raman (2025) consider the noisy setting where the adversary can provide negative examples and the generator is unaware of which examples are noisy. Both works provide analogous definitions and characterizations of uniform and non-uniform generation in their respective settings.

2. Preliminaries

Let $\mathcal{X}$ denote a *countable* example space (e.g., text, molecules, images) and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ denote a binary hypothesis class (e.g., class of all vision transformers). Let $\mathcal{X}^{\star}$ denote the set of all finite subsets of $\mathcal{X}$. In the context of language modeling, one can think about $\mathcal{X}$ as the collection of all valid strings, and each hypothesis $h \in \mathcal{H}$ as a language (i.e., a subset of strings). For a hypothesis

$h \in \mathcal{H}$, let $\text{supp}(h) := \{x \in \mathcal{X} : h(x) = 1\}$, that is, its collection of positive examples. For any $h \in \mathcal{H}$, an *enumeration* of $\text{supp}(h)$ is any infinite sequence $x_1, x_2, \ldots$ such that $\bigcup_{i \in \mathbb{N}} \{x_i\} = \text{supp}(h)$. In other words, for every $x \in \text{supp}(h)$, there exists an $i \in \mathbb{N}$ such that $x_i = x$. We will let $[N] := \{1, \ldots, N\}$ and sometimes abbreviate a finite sequence $x_1, \ldots, x_n$ as $x_{1:n}$.

For any class $\mathcal{H}$ and a finite sequence of examples $x_1, \ldots, x_n$, let $\mathcal{H}(x_1, \ldots, x_n) := \{h \in \mathcal{H} : \{x_1, \ldots, x_n\} \subseteq \text{supp}(h)\}$. In learning theory, $\mathcal{H}(x_1, \ldots, x_n)$ is also called the "version space" of $\mathcal{H}$ induced by the sample $\{(x_i, 1)\}_{i=1}^n$ (i.e., the set of all consistent hypothesis). For any class $\mathcal{H}$, define $\langle \cdot \rangle_{\mathcal{H}}$ as its induced closure operator such that

$$\langle x_1, \ldots, x_n \rangle_{\mathcal{H}} := \begin{cases} \bigcap_{h \in \mathcal{H}(x_{1:n})} \text{supp}(h), & \text{if } |\mathcal{H}(x_{1:n})| \geq 1 \\ \perp, & \text{if } |\mathcal{H}(x_{1:n})| = 0 \end{cases}.$$

In learning-theoretic terms, $\langle x_1, \ldots, x_n \rangle_{\mathcal{H}}$ is the set of positive examples common to all hypothesis in the version space of $\mathcal{H}$ consistent with the sample $(x_1, 1), \ldots, (x_n, 1)$. It turns out that one can check closure membership, i.e. given an example x and a sequence of examples $x_1, \ldots, x_n$, return $\mathbb{1}\{\langle x_1, \ldots, x_n \rangle_{\mathcal{H}} \neq \perp \text{ and } x \in \langle x_1, \ldots, x_n \rangle_{\mathcal{H}}\}$, using access to an Empirical Risk Minimization (ERM) oracle. See Appendix I.1 for the full discussion. Finally, we will make the following assumption about hypothesis classes.

Assumption 1 (Uniformly Unbounded Support (UUS)) *A hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ satisfies the* Uniformly Unbounded Support (UUS) *property if $|\text{supp}(h)| = \infty$ for every $h \in \mathcal{H}$.*

As noted by Kleinberg and Mullainathan (2024), such an assumption is necessary to make generation of new examples possible. In the next section, we introduce and define several notions of generatability, including the notion of "generatability in the limit" by Kleinberg and Mullainathan (2024). In Appendix A, we restate the results of Angluin (1979, 1988) for language identification and review existing notions of predictability using the notation above.

2.1. Generatability

We state the model of generation in Kleinberg and Mullainathan (2024) using our notation. Consider the following two-player game. At the start, the adversary picks a hypothesis $h \in \mathcal{H}$ and an enumeration $x_1, x_2, \ldots$ of $\text{supp}(h)$ and does not reveal them to the learner. The game then proceeds over rounds $t = 1, 2, \ldots$. In each round $t \in \mathbb{N}$, the adversary reveals x_t. The generator, after observing $x_1, \ldots, x_t$, must output $\hat{x}_t \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}$ and suffers the loss $\mathbb{1}\{\hat{x}_t \notin \text{supp}(h) \setminus \{x_1, \ldots, x_t\}\}$. Crucially, the generator *never* observes its loss as it does not know h. The goal of the generator is to *eventually* generate new, positive examples $\hat{x}_t \in \text{supp}(h) \setminus \{x_1, \ldots, x_t\}$. To make this all formal, we first define a generator.

Definition 1 (Generator) *A generator is a map $\mathcal{G} : \mathcal{X}^\star \to \mathcal{X}$ that takes a finite sequence of examples $x_1, x_2, \ldots$ and outputs a new example x.*

We can now use the existence of a good generator to define the property of generatability in the limit from Section 2 of Kleinberg and Mullainathan (2024).

Definition 2 (Generatability in the Limit) *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any hypothesis class satisfying the* UUS *property. Then, $\mathcal{H}$ is* generatable in the limit *if there exists a generator $\mathcal{G}$ such that for every $h \in \mathcal{H}$, and any* enumeration $x_1, x_2, \ldots$ *of* $\text{supp}(h)$*, there exists a $t^\star \in \mathbb{N}$ such that $\mathcal{G}(x_{1:s}) \in \text{supp}(h) \setminus \{x_1, \ldots, x_s\}$ for all $s \geq t^\star$.*

Roughly speaking, generatability in the limit captures the *existence* of the ability to eventually generate positive examples, when no feedback is available and the underlying hypothesis is not known. Note that the adversary can repeat examples in its stream, but it must eventually enumerate all the positive examples of the chosen hypothesis. On the other hand, the adversary is still powerful as it can examine/simulate the generator $\mathcal{G}$ in any way imaginable before choosing the true hypothesis and the enumeration of its support. While not explicitly defined, Kleinberg and Mullainathan (2024) also consider a notion of *uniform* generatability in Theorem 2.2 of the Section titled "A Result for Finite Collections." By "uniform", we mean that the amount of time required before the generator should perfectly generate new positive examples should only be a function of the class $\mathcal{H}$ and thus the same across all hypothesis $h \in \mathcal{H}$ and enumerations of $\text{supp}(h)$. This is in contrast to generatability in the limit, where the time step $t^\star$ can depend on both the sequence of examples $x_1, x_2, \ldots$ and the selected hypothesis $h \in \mathcal{H}$. Definition 3 formalizes this notion of uniform generatability.

Definition 3 (Uniform Generatability) *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any hypothesis class satisfying the* UUS *property. Then, $\mathcal{H}$ is* uniformly *generatable, if there exists a generator $\mathcal{G}$ and $d^\star \in \mathbb{N}$, such that for every $h \in \mathcal{H}$ and any sequence $x_1, x_2, \ldots$ with $\{x_1, x_2, \ldots\} \subseteq \text{supp}(h)$, if there exists $t^\star \in \mathbb{N}$ such that $|\{x_1, \ldots, x_{t^\star}\}| = d^\star$, then $\mathcal{G}(x_{1:s}) \in \text{supp}(h) \setminus \{x_1, \ldots, x_s\}$ for all $s \geq t^\star$.*

A subtle detail in Definition 3 is the fact that we must force the adversary to play a sufficient number of distinct examples before we require the generator to be perfect. This is necessary, as otherwise, the adversary can play the same example in all rounds, and even a simple hypothesis class with two hypotheses which share exactly one example in their support cannot be uniformly generatable. This restriction is also captured by Theorem 2.2 in Kleinberg and Mullainathan (2024) and effectively means that a generator witnessing Definition 3 must be able to generate new, positive examples after observing a sufficient number of distinct positive examples. In fact, given a generator $\mathcal{G}$, the number of positive examples needed before perfect generation is akin to "sample complexity" and "mistake-bounds" in PAC and online learning. We expand more about this in Appendix C.

While natural, one can also arrive at uniform generatability by swapping the order of quantifiers in Definition 2. Namely, one also gets uniform generatability by moving the generation sample complexity "left" twice. That is, before the quantifiers on the selected hypotheses and the stream, and therefore, in line with the existence of the generator. This motivates an *intermediate* setting, we term non-uniform generatability, where we only move the generation sample complexity "left" once, and in particular, only before the quantifier on the stream.

Definition 4 (Non-uniform Generatability) *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any hypothesis class satisfying the* UUS *property. Then, $\mathcal{H}$ is* non-uniformly generatable *if there exists a generator $\mathcal{G}$ such that for every $h \in \mathcal{H}$, there exists a $d^\star \in \mathbb{N}$ such that for any sequence $x_1, x_2, \ldots$ with $\{x_1, x_2, \ldots\} \subseteq \text{supp}(h)$, if there exists $t^\star \in \mathbb{N}$ such that $|\{x_1, \ldots, x_{t^\star}\}| = d^\star$, then $\mathcal{G}(x_{1:s}) \in \text{supp}(h) \setminus \{x_1, \ldots, x_s\}$ for all $s \geq t^\star$.*

We use the term "non-uniform" to denote the fact that the number of distinct examples needed before perfect generation can depend on the hypothesis selected by the adversary, and hence, it is "non-uniform" over the hypothesis class $\mathcal{H}$. But do note that the number of distinct examples needed is still uniform over the possible stream chosen by the adversary. Hence, we use the term "uniform" and "non-uniform" only with respect to the hypothesis chosen by the adversary. Again, we require the restriction that the adversary must select a sufficient number of distinct examples, as otherwise, even trivial classes are not non-uniformly generatable.

By inspecting the order of quantifiers, it is clear that uniform generatability is the strongest of the three properties, while generatability in the limit is the weakest. In particular, for any class $\mathcal{H}$, we have that Uniform Generatability $\implies$ Non-uniform Generatability $\implies$ Generatability in the Limit. This ordering is tight in the sense that the reverse directions are *not* true.

Proposition 5 *Let $\mathcal{X}$ be countable. There exists classes $\mathcal{H}_1, \mathcal{H}_2 \subseteq \{0,1\}^{\mathcal{X}}$ satisfying the* UUS *property such that: (i) $\mathcal{H}_1$ is non-uniformly generatable but not uniformly generatable and (ii) $\mathcal{H}_2$ is generatable in the limit but not non-uniformly generatable.*

We will prove Proposition 5 in Appendix H. We end this section by highlighting an important practical property of uniform and non-uniform generators. Our definitions of uniform and non-uniform generatability do not require the adversary to select an enumeration of the support of its selected hypothesis. That is, any valid sequence with a sufficient number of distinct examples is enough. As a consequence, once enough distinct examples are revealed to the generator, the adversary can reveal the generator's prediction on round t as the positive example on round $t+1$. This effectively means that once the generator has observed a sufficient number of distinct example, it can be used *auto-regressively* to produce new, unseen positive examples. This property might be useful when generators are used for downstream tasks.

3. Characterizations of Generatability

In this section, we provide a characterization of which classes are uniformly and non-uniformly generatable, as well as, a weaker sufficiency condition for generatability in the limit. We start with characterizing uniform generation.

3.1. Uniform Generatability

In learning theory, it is often the case that the most "obvious" necessary condition is also sufficient. To that end, we seek a combinatorial dimension of $\mathcal{H}$ whose infiniteness implies that $\mathcal{H}$ is *not uniformly generatable*. By inverting Definition 3, we have that $\mathcal{H}$ is *not* uniformly generatable if for every generator $\mathcal{G}$ and every $d \in \mathbb{N}$, there exists a $h^\star \in \mathcal{H}$ and a sequence $(x_i)_{i\in\mathbb{N}}$ with $\{x_1, x_2, \dots\} \subseteq \text{supp}(h^\star)$ such that for every time point $t \in \mathbb{N}$ where $|\{x_1, \dots, x_t\}| = d$, there exists a $s \geq t$ such that $\mathcal{G}(x_{1:s}) \notin \text{supp}(h^\star) \setminus \{x_1, \dots, x_s\}$. So, our candidate dimension should satisfy the property that when it is infinite, we can find arbitrarily large sequences of examples after which any generator is guaranteed to make a mistake. With this in mind, we are ready to present the Closure dimension, whose finiteness satisfies exactly this property.

Definition 6 (Closure dimension) *The* Closure dimension *of $\mathcal{H}$, denoted $\mathrm{C}(\mathcal{H})$, is the largest natural number $d \in \mathbb{N}$ for which there exists* distinct *$x_1, \dots, x_d \in \mathcal{X}$ such that $\langle x_1, \dots, x_d \rangle_{\mathcal{H}} \neq \perp$ and $|\langle x_1, \dots, x_d \rangle_{\mathcal{H}}| < \infty$. If this is true for arbitrarily large $d \in \mathbb{N}$, then we say that $\mathrm{C}(\mathcal{H}) = \infty$. On the other hand, if this is not true for $d = 1$, we say that $\mathrm{C}(\mathcal{H}) = 0$.*

The following lemma shows that the finiteness of $\mathrm{C}(\mathcal{H})$ is necessary for uniform generatability. The high-level idea is that the adversary can force the learner to make a mistake at time point t, if there are no common positive examples amongst those hypotheses that contain $x_1, \dots, x_t$ in their support. The finiteness of $\mathrm{C}(\mathcal{H})$ guarantees the existence of such a $t \in \mathbb{N}$ and $x_1, \dots, x_t$.

Lemma 7 (Necessity in Theorem 9) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any class satisfying the* UUS *property. If $\mathrm{C}(\mathcal{H}) = \infty$, then $\mathcal{H}$ is* not *uniformly generatable.*

The proof of Lemma 7, provided in Appendix D, shows that the Closure dimension also provides a *quantitative* lower bound on the optimal uniform generation sample complexity. Namely, for any class $\mathcal{H}$ and generator $\mathcal{G}$, we have that $d_{\mathcal{G}} \geq \mathrm{C}(\mathcal{H})$, where $d_{\mathcal{G}}$ is the smallest number of distinct examples that $\mathcal{G}$ needs in order to generate new samples correctly (see Appendix C for a formal definition of $d_{\mathcal{G}}$). Next, we move to the sufficiency condition. The following lemma shows that the finiteness of $\mathrm{C}(\mathcal{H})$ is also sufficient for uniform generatability. The main idea is that if $\mathrm{C}(\mathcal{H}) = d$, then one only needs to observe $d+1$ distinct examples before one can identify an infinite "core" set $S \subseteq \mathcal{X}$ that lies in the support of the hypothesis chosen by the adversary. The generator can then just play from the set S for all future rounds.

Lemma 8 (Sufficiency in Theorem 9) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any class satisfying the* UUS *property. When $\mathrm{C}(\mathcal{H}) < \infty$, there exists a generator $\mathcal{G}$, such that for every $h \in \mathcal{H}$ and any sequence $(x_i)_{i \in \mathbb{N}}$ with $\{x_1, x_2, \dots\} \subseteq \mathrm{supp}(h)$, if there exists a $t \in \mathbb{N}$ such that $|\{x_1, \dots, x_t\}| = \mathrm{C}(\mathcal{H}) + 1$, then $\mathcal{G}(x_{1:s}) \in \mathrm{supp}(h) \setminus \{x_1, \dots, x_s\}$ for all $s \geq t$.*

Proof Let $0 \leq d < \infty$ and suppose $\mathrm{C}(\mathcal{H}) = d$. Then, for every distinct sequence of $d+1$ examples $x_1, \dots, x_{d+1}$ such that $\langle x_{1:d+1} \rangle_{\mathcal{H}} \neq \perp$, we have that $|\langle x_1, \dots, x_{d+1} \rangle_{\mathcal{H}}| = \infty$. Consider the following generator $\mathcal{G}$. Until $d+1$ distinct examples are observed, $\mathcal{G}$ plays any $\hat{x}_s$. Suppose on round $t^\star$, $\mathcal{G}$ observes $d+1$ distinct examples. Then, $\mathcal{G}$ plays any $\hat{x}_s \in \langle x_1, \dots, x_{t^\star} \rangle_{\mathcal{H}} \setminus \{x_1, \dots, x_s\}$ for all $s \geq t^\star$. Let $h^\star$ be the hypothesis chosen by the adversary. It suffices to show that $\hat{x}_s \in \mathrm{supp}(h^\star) \setminus \{x_1, \dots, x_s\}$ for all $s \geq t^\star$. However, this just follows from the fact that $|\langle x_1, \dots, x_{t^\star} \rangle_{\mathcal{H}}| = \infty$ and $\langle x_1, \dots, x_{t^\star} \rangle_{\mathcal{H}} \subseteq \mathrm{supp}(h^\star)$. In particular, $|\langle x_1, \dots, x_{t^\star} \rangle_{\mathcal{H}}| = \infty$ ensures that $\hat{x}_s$ is well-defined and $\langle x_1, \dots, x_{t^\star} \rangle_{\mathcal{H}} \subseteq \mathrm{supp}(h^\star)$ ensures that it always lies in $\mathrm{supp}(h^\star) \setminus \{x_1, \dots, x_s\}$. ■

The generator in Lemma 8 can be efficiently implemented given access to the following max-min oracle $\mathcal{O}_{\textit{max-min}} : 2^{\{0,1\}^{\mathcal{X}}} \times \mathcal{X}^\star \to \mathcal{X}$. Given a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ and a finite sequence of examples $x_1, \dots, x_t$, $\mathcal{O}_{\textit{max-min}}$ returns $\arg\max_{x \in \mathcal{X} \setminus \{x_1, \dots, x_t\}} \min_{h \in \mathcal{H}} \sum_{i=1}^{t} \mathbb{1}\{h(x_i) \neq 1\} + \mathbb{1}\{h(x) \neq 0\}$. This max-min oracle should remind the reader of the min-max objective/two-player game used to motivate Generative Adversarial Networks (see Equation 1 in Goodfellow et al. (2014)). In particular, for our max-min oracle, one can think of the minimizer as the discriminator and the outer maximizer as the generator. See Appendix I.2 for the full discussion. Composing Lemmas 7 and 8 gives a characterization of uniform generatability.

Theorem 9 (Characterization of Uniform Generatability) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ satisfy the* UUS *property. Then, $\mathcal{H}$ is uniformly generatable if and only if $\mathrm{C}(\mathcal{H}) < \infty$.*

We highlight that the Closure dimension not only provides a qualitative characterization of uniform generatability, but also a quantitative characterization – the optimal uniform generation sample complexity is exactly $\Theta(\mathrm{C}(\mathcal{H}))$. Kleinberg and Mullainathan (2024) proved that all countable classes are generatable in the limit. Our next result, proved in Appendix D, improves upon this by proving the existence of *uncountably* infinite hypothesis classes that are uniformly generatable.

Corollary 10 *Let $\mathcal{X}$ be countable. There exists a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ that is uncountably large, satisfies the* UUS *property, and is uniformly generatable.*

3.2. Non-uniform Generatability

We next move to characterize non-uniform generatability. Similar to non-uniform PAC and online learnability, we show that our characterization of uniform generatability leads to a characterization of non-uniform generatability. The proof of Theorem 11 is in Appendix E.1. While the proof of necessity uses non-constructive arguments, for the sufficiency direction, we explicitly construct a non-uniform generator for $\mathcal{H}$ using uniform generators for $\mathcal{H}_1, \mathcal{H}_2, \ldots$.

Theorem 11 (Characterization of Non-uniform Generatability) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ satisfy the* UUS *property. Then, $\mathcal{H}$ is non-uniformly generatable if and only if there exists a* non-decreasing *sequence of classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots$ such that $\mathcal{H} = \bigcup_{i \in \mathbb{N}} \mathcal{H}_i$ and $\mathrm{C}(\mathcal{H}_n) < \infty$ for every $n \in \mathbb{N}$.*

One might wonder whether we can drop "non-decreasing" condition in Theorem 11 and write $\mathcal{H}$ as the countable union of uniformly generatable classes $\mathcal{H}_1, \mathcal{H}_2, \ldots$. However, this is *false* as we will show in Lemma 15 – while such a condition is necessary, it is *not* sufficient. Nevertheless, we can use Theorem 11 and the fact that all finite classes are uniformly generatable (Kleinberg and Mullainathan, 2024) to show that every countable hypothesis class is actually non-uniformly generatable. The proof of Corollary 12 is in Appendix E.2.

Corollary 12 (Countable Classes are Non-uniformly Generatable) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any hypothesis class that satisfies the* UUS *property. If $\mathcal{H}$ is countable, then $\mathcal{H}$ is non-uniformly generatable.*

This improves upon Kleinberg and Mullainathan (2024)'s result that every countable hypothesis class is generatable in the limit since non-uniform generation implies generation in the limit, but not vice versa. We note that Charikar and Pabbaraju (2024) also independently establish Corollary 12.

3.3. Generatability in the Limit

Kleinberg and Mullainathan (2024) showed that all countable classes are generatable in the limit. Here, we provide an alternate sufficiency condition for generatability in the limit which, in conjunction with countableness, expands the collection of classes which are generatable in the limit.

Theorem 13 (Sufficient Condition for Generatability in the Limit) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any class satisfying the* UUS *property. If there exists a finite sequence of classes $\mathcal{H}_1, \ldots, \mathcal{H}_n$ such that $\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{H}_i$ and $\mathrm{C}(\mathcal{H}_i) < \infty$ for all $i \in [n]$, then $\mathcal{H}$ is generatable in the limit.*

In fact, the following corollary, proved in Appendix F, gives examples of uncountably infinite classes that, using Theorem 13, are generatable in the limit.

Corollary 14 *Let $\mathcal{X} = \mathbb{N}$ and $S_1, \ldots, S_n \subseteq \mathbb{N}$ be any finite sequence of countable infinite subsets of $\mathbb{N}$. For every $i \in [n]$, let $\mathcal{H}_i = \{x \mapsto \mathbb{1}\{x \in S_i \cup A\} : A \in 2^{\mathbb{N}}\}$. Then, $\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{H}_i$ is generatable in the limit.*

Proof (of Theorem 13) Let $\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{H}_i$ be such that $\mathrm{C}(\mathcal{H}_i) < \infty$ for all $i \in [n]$. Let $c := \max_{i \in [n]} \mathrm{C}(\mathcal{H}_i)$. Consider the following generator $\mathcal{G}$. Let $t^\star \in \mathbb{N}$ be the smallest time point for which $|\{x_1, \ldots, x_{t^\star}\}| = c + 1$. $\mathcal{G}$ plays arbitrarily up to, but not including, time point $t^\star$. On time point $t^\star$, $\mathcal{G}$ computes $\langle x_1, \ldots, x_{t^\star} \rangle_{\mathcal{H}_i}$ for all $i \in [n]$. Let $S \subseteq [n]$ be the subset of indices such that $i \in S$ if and only if $\langle x_1, \ldots, x_{t^\star} \rangle_{\mathcal{H}_i} \neq \perp$. For every $i \in S$, let $(z_j^{(i)})_{j \in \mathbb{N}}$ be the natural ordering of $\langle x_1, \ldots, x_{t^\star} \rangle_{\mathcal{H}_i}$, which is guaranteed to exist since $\mathcal{X}$ is countable. For every $t \geq t^\star$, sequence of revealed examples $x_1, \ldots, x_t$, and $i \in S$, $\mathcal{G}$ computes $n_t^i := \max\{n \in \mathbb{N} : \{z_1^{(i)}, \ldots, z_n^{(i)}\} \subset \{x_1, \ldots, x_t\}\}$ and $i_t \in \arg\max_{i \in S} n_t^i$. Finally, $\mathcal{G}$ plays any $\hat{x}_t \in \langle x_1, \ldots, x_{t^\star} \rangle_{\mathcal{H}_{i_t}} \setminus \{x_1, \ldots, x_t\}$. We claim that $\mathcal{G}$ generates from $\mathcal{H}$ in the limit.

Let $h^\star \in \mathcal{H}$ be the hypothesis chosen by the adversary and $x_1, x_2, \ldots$ be the selected enumeration of $\mathrm{supp}(h^\star)$. Let $c = \max_{i \in [n]} \mathrm{C}(\mathcal{H}_i)$ and $t^\star \in \mathbb{N}$ be the smallest time point for which $|\{x_1, \ldots, x_{t^\star}\}| = c + 1$. By definition of $\mathrm{C}(\cdot)$, we know that for every $j \in S$, $|\langle x_1, \ldots, x_{t^\star} \rangle_{\mathcal{H}_j}| = \infty$. Let $S^\star \subseteq S$ be such that $i \in S^\star$ if and only if $\langle x_1, \ldots, x_{t^\star} \rangle_{\mathcal{H}_i} \subseteq \mathrm{supp}(h^\star)$. It suffices to show that there exists a finite time point $s^\star \in \mathbb{N}$ such that for all $t \geq s^\star$, we have that $i_t \in S^\star$. To see why such an $s^\star$ must exist, pick some $j^\star \in S^\star$. Note that $n_t^{j^\star} \to \infty$ because $x_1, x_2, \ldots$ is an enumeration of $\mathrm{supp}(h^\star)$. On the other hand, observe that for every $j \notin S^\star$, there exists a $n^j \in \mathbb{N}$ such that $n_t^j \leq n^j$. This is because, if $j \notin S^\star$, then there must be an index $n^j \in \mathbb{N}$ such that $z_{n^j}^{(j)} \notin \mathrm{supp}(h^\star)$. Thus, n_t^j must be at most n^j. Since there are at most a finite number of indices not in $S^\star$, we have that $\max_{j \notin S^\star} n^j < \infty$, which means that eventually, $n_t^{j^\star} > n_t^j$ for all $j \notin S^\star$, and thus there exists a $s^\star \in \mathbb{N}$ such that $i_t \in S^\star$ for all $t \geq s^\star$. This completes the proof. ∎

The algorithm in the proof of Theorem 13 can be efficiently implemented given access to an ERM oracle $\mathcal{O} : 2^{\{0,1\}^{\mathcal{X}}} \times (\mathcal{X} \times \{0,1\})^\star \to \mathbb{N} \cup \{0\}$ and an oracle $\mathcal{O}_{\mathrm{C}} : \mathcal{H} \to \mathbb{N}$ that can compute upper bounds on the Closure dimension. See Appendix I.3 for more details. One might ask whether the sufficiency condition in Theorem 13 can be extended to account for classes $\mathcal{H}$ which can be written as the *countable* union of uniformly generatable classes. Lemma 16 shows that this is actually *not* the case – there exists a countable sequence of uniformly generatable classes whose union is not generatable in the limit! Nevertheless, in Appendix G, we do give an even weaker sufficiency condition for generatability in the limit.

3.4. Generation is Unlike Prediction

In this section, we flesh out the landscape of generation versus prediction for countable hypothesis classes. Namely, we seek to compare generation with prediction and understand how these two properties of hypothesis classes compare with one another. To evaluate the predictability of a hypothesis class $\mathcal{H}$, we use the standard notions of PAC and online learnability. It is well known that the VC dimension and Littlestone dimension characterize PAC and online learnability respectively (Vapnik and Chervonenkis, 1971; Littlestone, 1987). See Appendix A for precise definitions and characterizations of predictability.

Informally, our main result is Figure 1 which captures the landscape of generatability and predictability for countable classes. Figure 1 shows that even amongst countable classes $\mathcal{H}$ (which we know, via Corollary 12, are always non-uniformly generatable), uniform generatability and prediction are truly incomparable – knowing whether a class is PAC or online learnable tells you *nothing* about whether it is uniformly generatable, and vice versa. As such, these are two fundamentally different properties of a hypothesis class. Perhaps the best evidence of this is the difference in their

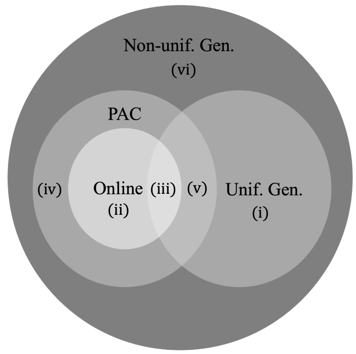

Figure 1: Generation vs. Prediction for countable classes. (i-vi) map to items in Theorem 47.

behavior under unions. PAC and online learnability behave very nicely under unions – if $\mathcal{H}_1$ and $\mathcal{H}_2$ are PAC/online learnable, then their union $\mathcal{H}_1 \cup \mathcal{H}_2$ is also PAC/online learnable (Dudley, 1978; Alon et al., 2020). However, the same *cannot* be said for generation – both uniform generation and non-uniform generation are *not* closed under finite unions.

Lemma 15 *Let $\mathcal{X}$ be countable. There exists a* UUS *class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ and a hypothesis $h : \mathcal{X} \to \{0,1\}$ with $|\text{supp}(h)| = \infty$ such that $\mathrm{C}(\mathcal{H}) = 0$, but $\mathcal{H} \cup \{h\}$ is* not *non-uniformly generatable.*

Lemma 15 shows something stronger – the addition of a single hypothesis can change uniform and non-uniform generatability! Such divergent behavior is not present in PAC/online learnability. Surprisingly, a similar, but weaker, statement can be said about generatability in the limit.

Lemma 16 *Let $\mathcal{X}$ be countable. There exists a countable sequence of classes $\mathcal{H}_1, \mathcal{H}_2, \ldots,$ all satisfying the* UUS *property, such that $\mathrm{C}(\mathcal{H}_i) = 0$ for all $i \in \mathbb{N}$, but the class $\mathcal{H} = \bigcup_{i \in \mathbb{N}} \mathcal{H}_i$ is* not *generatable in the limit.*

Lemma 16 shows that even generatability in the limit, the *weakest* definition of generatability, is not very well behaved under (countable) unions. An interesting question is whether generatability in the limit is even closed under *finite* unions. We leave this as an open question in Section M. Lemma 15 and 16 are proved in Appendix J.1.

4. Extension to Prompted Generation

So far, generators are only required to eventually produce new positive examples. This setup does not account for the fact that in many real-life situations, we would like to generate objects with respect to a *prompt*. For example, we may like to generate an image based on a text caption, respond to a query from a user, or generate a protein from a protein family (Madani et al., 2023).

Inspired by recent interests in multiclass learning, we capture a prompted-version of generation by taking $\mathcal{Y}$ to be an abstract prompt space and $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ to be a *multiclass* hypothesis class. Given a hypothesis $h \in \mathcal{H}$ and a prompt $y \in \mathcal{Y}$, one should think of the set $\{x \in \mathcal{X} : h(x) = y\}$ as the collection of valid generatable objects for prompt y with respect to hypothesis h. Roughly speaking, if $h \in \mathcal{H}$ captures the true world, and the prompt on round t is $y_t \in \mathcal{Y}$, then the goal of the generator should be to output an example $\hat{x}_t$ such that $h(\hat{x}_t) = y_t$.

To handle prompts, we need a slight modification of the game defined in Section 2. As in the binary case, the adversary first selects a hypothesis $h \in \mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and a sequence of examples $x_1, x_2, \ldots$. But now, it also selects a sequence of prompts $y_1, y_2, \ldots$. In each round $t \in \mathbb{N}$, the adversary reveals the tuple $(x_t, h(x_t), y_t)$ and the goal of the learner is output $\hat{x}_t \in \{x \in \mathcal{X} : h(x) = y_t\} \setminus \{x_1, \ldots, x_t\}$.

To make the outlined notion of prompted generatability more formal, we extend the "support" and UUS property to the prompted setting. For any $h \in \mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and any $y \in \mathcal{Y}$, define the y-support of h as $\operatorname{supp}(h, y) := \{x \in \mathcal{X} : h(x) = y\}$. Then, the Prompted Uniformly Unbounded Support property just requires that for every $h \in \mathcal{H}$ and any $y \in \mathcal{Y}$, the y-support of h is unbounded.

Assumption 2 (Prompted Uniformly Unbounded Support (PUUS)) *A hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ satisfies the* Prompted Uniformly Unbounded Support (PUUS) *property if for every $y \in \mathcal{Y}$ and $h \in \mathcal{H}$, we have that $|\operatorname{supp}(h, y)| = \infty$.*

Like the UUS property, the PUUS property is only needed for bookkeeping purposes to prevent the adversary from presenting the generator with an impossible task (i.e. generating new examples for prompt $y \in \mathcal{Y}$ when no new, examples exist.) One can remove the PUUS restriction, but restrict the adversary to choose a prompted sequence so that the generator is always guaranteed the existence of new, unseen examples with the selected prompt y. This assumption is also captured in Section 7 of Kleinberg and Mullainathan (2024), where they assume that the adversary only reveals "non-trivial" prompts to the generator.

Next, we extend the definition of Generator to a Prompted Generator.

Definition 17 (Prompted Generator) *A prompted generator is a map $\mathcal{G} : (\mathcal{X} \times \mathcal{Y} \times \mathcal{Y})^{\star} \to \mathcal{X}$ that takes a finite sequence of tuples $(x_1, h(x_1), y_1), (x_2, h(x_1), y_2), \ldots$ and outputs an example x.*

Then, we can then define analogous notion of prompted uniform generatability, non-uniform generatability, and generatability in the limit.

Definition 18 (Prompted Uniform Generatability) *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be any hypothesis class satisfying the* PUUS *property. Then, $\mathcal{H}$ is prompted* uniformly *generatable, if there exists a prompted generator $\mathcal{G}$ and a number $d^{\star} \in \mathbb{N}$, such that for every $h \in \mathcal{H}$, any sequence $(x_i, y_i)_{i \in \mathbb{N}}$, and any $y^{\star} \in \mathcal{Y}$, if there exists $t^{\star} \in \mathbb{N}$ such that $|\{x_1, \ldots, x_{t^{\star}}\} \cap \operatorname{supp}(h, y^{\star})| = d^{\star}$, then*

$$\mathcal{G}((x_1, h(x_1), y_1), \ldots, (x_s, h(x_s), y_s)) \in \operatorname{supp}(h, y_s) \setminus \{x_1, \ldots, x_s\}.$$

for all $s \geq t^{\star}$ where $y_s = y^{\star}$.

Definition 19 (Prompted Non-uniform Generatability) *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be any hypothesis class satisfying the* PUUS *property. Then, $\mathcal{H}$ is prompted* non-uniformly *generatable, if there exists a prompted generator $\mathcal{G}$, such that for every $h \in \mathcal{H}$, there exists a number $d^{\star} \in \mathbb{N}$, such that for every sequence $(x_i, y_i)_{i \in \mathbb{N}}$, and any $y^{\star} \in \mathcal{Y}$, if there exists $t^{\star} \in \mathbb{N}$ such that $|\{x_1, \ldots, x_{t^{\star}}\} \cap \operatorname{supp}(h, y^{\star})| = d^{\star}$, then*

$$\mathcal{G}((x_1, h(x_1), y_1), \ldots, (x_s, h(x_s), y_s)) \in \operatorname{supp}(h, y_s) \setminus \{x_1, \ldots, x_s\}$$

for all $s \geq t^{\star}$ where $y_s = y^{\star}$.

Definition 20 (Prompted Generatability in the Limit) *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be any hypothesis class satisfying the* PUUS *property. Then, $\mathcal{H}$ is prompted generatable in the limit, if there exists a prompted generator $\mathcal{G}$, such that for every $h \in \mathcal{H}$, any sequence $(x_i, y_i)_{i \in \mathbb{N}}$, and any $y^\star \in \mathcal{Y}$, if $\operatorname{supp}(h, y^\star) \subseteq \{x_1, x_2, \dots\}$, then there exists $t^\star \in \mathbb{N}$ such that*

$$\mathcal{G}((x_1, h(x_1), y_1), \dots, (x_s, h(x_s), y_s)) \in \operatorname{supp}(h, y_s) \setminus \{x_1, \dots, x_s\}.$$

for all $s \geq t^\star$ where $y_s = y^\star$.

Roughly speaking, Definitions 18, 19, and 20 state that a class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is prompted generatable if for any prompt $y \in \mathcal{Y}$, after a sufficient number of distinct examples with prompt y are observed, one can generate new examples with prompt y. Like in the binary case, generators $\mathcal{G}$ that witness Definitions 18 and 19 have the following nice property – for any prompt $y \in \mathcal{Y}$, once a sufficient number of distinct examples are observed with prompt y, $\mathcal{G}$ can produce new, unseen examples for prompt y auto-regressively and without any form of supervision. We highlight the differences and similarities between our notion of prompted generation and that by Kleinberg and Mullainathan (2024) in Appendix K.

4.1. Characterizations of Prompted Generatability

To characterize prompted uniform and non-uniform generatability, we extend the Closure dimension to the prompted case. To do so, we need an extension of the closure operator to a *prompted* closure operator $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Namely, for every finite sequence of examples $x_1, \dots, x_n$ and prompt $y \in \mathcal{Y}$, we define $\mathcal{H}(x_{1:n}, y) := \{h \in \mathcal{H} : h(x_i) = y, i = 1, \dots, n\}$ and

$$\langle x_{1:n}, y \rangle_{\mathcal{H}} := \begin{cases} \bigcap_{h \in \mathcal{H}(x_{1:n}, y)} \operatorname{supp}(h, y), & \text{if } |\mathcal{H}(x_{1:n}, y)| \geq 1 \\ \perp, & \text{if } |\mathcal{H}(x_{1:n}, y)| = 0 \end{cases}$$

We are now ready to define the Prompted Closure dimension.

Definition 21 (Prompted Closure dimension) *The* Prompted Closure dimension *of $\mathcal{H}$, denoted $\operatorname{PC}(\mathcal{H})$, is the largest number $d \in \mathbb{N}$ for which there exists* distinct $x_1, \dots, x_d \in \mathcal{X}$ *and a prompt $y \in \mathcal{Y}$ such that $|\langle (x_1, \dots, x_d), y \rangle_{\mathcal{H}}| \neq \perp$ and $|\langle (x_1, \dots, x_d), y \rangle_{\mathcal{H}}| < \infty$. If this is true for arbitrarily large $d \in \mathbb{N}$, then we say that $\operatorname{PC}(\mathcal{H}) = \infty$. On the other hand, if this is not true for $d = 1$, we say that $\operatorname{PC}(\mathcal{H}) = 0$.*

Using analogous techniques, one can prove that finiteness of the Prompted Closure dimension is both necessary and sufficient for prompted uniform generatability.

Theorem 22 (Characterization of Prompted Uniform Generatability) *Let $\mathcal{X}$ and $\mathcal{Y}$ be countable. Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be any hypothesis class satisfying the* PUUS *property. Then, $\mathcal{H}$ is prompted uniformly generatable if and only if $\operatorname{PC}(\mathcal{H}) < \infty$.*

Likewise, the same characterization of non-uniform generatability for the binary case also goes through when considering the prompted setting.

Theorem 23 (Characterization of Prompted Non-uniform Generatability) *Let $\mathcal{X}$ and $\mathcal{Y}$ be countable. Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be any hypothesis class satisfying the* PUUS *property. Then, $\mathcal{H}$ is prompted non-uniformly generatable if and only if there exists a non-decreasing sequence of classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots$ such that $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$ and $\mathrm{PC}(\mathcal{H}_n) < \infty$ for every $n \in \mathbb{N}$.*

We highlight that we make no assumptions about the size $\mathcal{Y}$ in Theorems 22 and 23 apart from its countableness. That is, these theorems hold even when $\mathcal{Y}$ is countably infinite. The proofs of Theorem 22 and 23 are very similar to that of Theorem 9 and 11. For the sake of conciseness, we omit the details in the main text, but provide proof sketches in Appendix L. When $|\mathcal{Y}| < \infty$, we can show that all finite classes are prompted uniformly generatable and all countable classes are prompted non-uniformly generatable. The latter also implies that all countable classes are prompted generatable in the limit.

Corollary 24 *Let $\mathcal{X}$ be countable and $\mathcal{Y}$ be finite. Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be any hypothesis class satisfying the* PUUS *property. The following statements are true: (i) $|\mathcal{H}| < \infty \implies \mathcal{H}$ is prompted uniformly generatable and (ii) $\mathcal{H}$ is countably infinite $\implies \mathcal{H}$ is prompted non-uniformly generatable and hence prompted generatable in the limit.*

However, quite surprisingly, we find that this is not the case when $|\mathcal{Y}| = \infty$ – there exists a finite class which is not prompted non-uniformly generatable!

Lemma 25 *Let $\mathcal{X}$ be countable and $\mathcal{Y}$ be countably infinite. There exists a finite class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ satisfying the* PUUS *property such that $\mathcal{H}$ is not prompted non-uniformly generatable.*

We prove Corollary 24 and Lemma 25 in Appendix L. We leave as an open question whether such a separation exists for prompted generatability in the limit. These results highlight that the behavior of prompted generatability changes significantly when the prompt space is allowed to be unbounded. This sort of phase transition is not unique to generation, and has also been observed in the context of multiclass PAC/online learning and uniform convergence (Natarajan, 1992; Daniely et al., 2011; Daniely and Shalev-Shwartz, 2014; Hanneke et al., 2023).

5. Discussion and Future Directions

In this work, we reinterpreted the model and results of Kleinberg and Mullainathan (2024) for language generation through the lens of modern learning theory. By doing so, we are able to formalize three frameworks for generation rooted in learning theory, i.e. generation in the limit, non-uniform generation, and uniform generation, and draw connections between multiclass learning and prompted generation. By abstracting the problem of generation to an arbitrary example space and binary hypothesis classes, we are able to study the fundamental nature of generation beyond language modeling. In Appendix M, we highlight several important directions for future work.

Acknowledgments

VR acknowledges the support of the NSF Graduate Research Fellowship Program.

References

Noga Alon, Amos Beimel, Shay Moran, and Uri Stemmer. Closure properties for private classification and online prediction. In *Conference on Learning Theory*, pages 119–152. PMLR, 2020.

Dana Angluin. Finding patterns common to a set of strings. In *Proceedings of the eleventh annual ACM Symposium on Theory of Computing*, pages 130–141, 1979.

Dana Angluin. Inductive inference of formal languages from positive data. *Information and control*, 45(2):117–135, 1980.

Dana Angluin. *Identifying languages from stochastic examples*. Yale University. Department of Computer Science, 1988.

Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, volume 3, page 1, 2009.

Olivier Bousquet, Steve Hanneke, Shay Moran, Ramon Van Handel, and Amir Yehudayoff. A theory of universal learning. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 532–541, 2021.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Moses Charikar and Chirag Pabbaraju. Exploring facets of language generation in the limit. *arXiv preprint arXiv:2411.15364*, 2024.

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

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. In Sham M. Kakade and Ulrike von Luxburg, editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 207–232, Budapest, Hungary, 09–11 Jun 2011. PMLR.

Richard M Dudley. Central limit theorems for empirical measures. *The Annals of Probability*, pages 899–929, 1978.

E Mark Gold. Language identification in the limit. *Information and control*, 10(5):447–474, 1967.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Steve Hanneke, Shay Moran, Vinod Raman, Unique Subedi, and Ambuj Tewari. Multiclass online learning and uniform convergence. *Proceedings of the 36th Annual Conference on Learning Theory (COLT)*, 2023.

GM Harshvardhan, Mahendra Kumar Gourisaria, Manjusha Pandey, and Siddharth Swarup Rautaray. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285, 2020.

Tony Jebara. *Machine learning: discriminative and generative*, volume 755. Springer Science & Business Media, 2012.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Alkis Kalavasis, Anay Mehrotra, and Grigoris Velegkas. Characterizations of language generation with breadth. 2024a. URL https://arxiv.org/abs/2412.18530.

Alkis Kalavasis, Anay Mehrotra, and Grigoris Velegkas. On the limits of language generation: Trade-offs between hallucination and mode collapse. *arXiv preprint arXiv:2411.09642*, 2024b.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s): 1–41, 2022.

Jon Kleinberg and Sendhil Mullainathan. Language generation in the limit. *arXiv preprint arXiv:2404.06757*, 2024.

Jon Kleinberg and Fan Wei. Density measures for language generation. 2025. URL https://arxiv.org/abs/2504.14370.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1987.

Zhou Lu. Non-uniform online learning: Towards understanding induction. *arXiv preprint arXiv:2312.00170*, 2023.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, 2023.

Maryanthe Malliaris and Shay Moran. The unstable formula theorem revisited via algorithms. *arXiv preprint arXiv:2212.05050*, 2022.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN 026201825X.

Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL http://probml.github.io/book2.

BK Natarajan. Probably approximate learning over classes of distributions. *SIAM Journal on Computing*, 21(3):438–449, 1992.

Charlotte Peale, Vinod Raman, and Omer Reingold. Representative language generation. 2025. URL https://arxiv.org/abs/2505.21819.

Ananth Raman and Vinod Raman. Generation from noisy examples. 2025. URL https://arxiv.org/abs/2501.04179.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014.

Quentin Vanhaelen, Yen-Chu Lin, and Alex Zhavoronkov. The advent of generative chemistry. *ACS Medicinal Chemistry Letters*, 11(8):1496–1505, 2020.

Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974.

Vladimir Naumovich Vapnik and Aleksei Yakovlevich Chervonenkis. On uniform convergence of the frequencies of events to their probabilities. *Teoriya Veroyatnostei i ee Primeneniya*, 16(2): 264–279, 1971.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Appendix A. Identifiability and Predictability

A.1. Identifiability

In identification, one seeks not to output new, positive examples $x \in \mathcal{X}$, but rather, to identify the true, underlying hypothesis $h \in \mathcal{H}$ chosen by the adversary. Historically, identification has been studied in the context of language modeling, with works dating as far back as Gold's seminal work on language identification in the limit (Gold, 1967). For consistency sake, we will formally define Gold's model in the notation of this paper. As in generation, we start by defining an Identifier.

Definition 26 (Identifier) *An Identifier is a map $\mathcal{I} : \mathcal{X}^\star \to \{0,1\}^{\mathcal{X}}$ that takes as input a finite sequence of examples $x_1, x_2, \ldots$ and outputs a hypothesis.*

The notion of identifiability in the limit can now be written in terms of the existence of good identifiers, and one can verify that our definition of identifiability in the limit is equivalent to that from Gold (1967) and Angluin (1979, 1980).

Definition 27 (Identifiability in the limit) *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any hypothesis class. Then, $\mathcal{H}$ is* identifiable in the limit *if there exists a identifier $\mathcal{I}$ such that for every $h \in \mathcal{H}$ and any enumeration $x_1, x_2, \ldots$ of $\operatorname{supp}(h)$, there exists a $t^\star \in \mathbb{N}$ such that $\mathcal{I}(x_{1:s}) = h$ for all $s \geq t^\star$.*

Although analogous definitions of uniform and non-uniform identifiability exist, we do not define or focus on them here as they are stronger than identifiability in the limit, which is already a very restrictive requirement.

A.2. Predictability

It is also natural to understand the predictability of a hypothesis class $\mathcal{H}$. Informally, the predictability of a class $\mathcal{H}$ should measure how easy it is to predict the labels of new examples $x_1, x_2, \ldots$ when the labels are produced by some unknown hypothesis $h \in \mathcal{H}$. In this paper, we will measure predictability of a hypothesis class $\mathcal{H}$ through their PAC and online *learnability* – properties of hypothesis classes that have been extensively studied by learning theorists (Vapnik and Chervonenkis, 1974; Littlestone, 1987; Ben-David et al., 2009).

In the PAC learning model, an adversary picks both a distribution $\mathcal{D}$ over $\mathcal{X}$ and a hypothesis $h \in \mathcal{H}$. The learner receives n iid samples $S = \{x_i, h(x_i)\}_{i=1}^n \sim (\mathcal{D} \times h)^n$, where we use $\mathcal{D} \times h$ to denote the distribution over $\mathcal{X} \times \{0,1\}$ defined procedurally by first sampling $x \sim \mathcal{D}$ and then outputting $(x, h(x))$. The goal of the learner is to use the sample S to output a hypothesis $f \in \{0,1\}^{\mathcal{X}}$ such that f has low error probability on a *future* labeled example drawn from $\mathcal{D}$.

Definition 28 (PAC Learnability) *A hypothesis class $\mathcal{H}$ is PAC learnable, if there exists a function $m : (0,1)^2 \to \mathbb{N}$ and a learning algorithm $\mathcal{A} : (\mathcal{X} \times \{0,1\})^\star \to \{0,1\}^{\mathcal{X}}$ with the following property: for every $\epsilon, \delta \in (0,1)$, distribution $\mathcal{D}$ on $\mathcal{X}$, and $h \in \mathcal{H}$, algorithm $\mathcal{A}$ when run on $n \geq m(\epsilon, \delta)$ iid samples $S = \{(x_i, h(x_i))\}_{i=1}^n \sim (\mathcal{D} \times h)^n$, outputs a predictor $f := \mathcal{A}(S) \in \{0,1\}^{\mathcal{X}}$ such that with probability at least $1 - \delta$ over $S \sim (\mathcal{D} \times h)^n$,*

$$\mathbb{E}_{x \sim \mathcal{D}}[\mathbb{1}\{f(x) \neq h(x)\}] \leq \epsilon.$$

The seminal result by Vapnik and Chervonenkis (1971) shows that the finiteness of the Vapnik–Chervonenkis (VC) dimension characterizes which hypothesis classes are PAC learnable.

Definition 29 (VC dimension) *A sequence $(x_1, \ldots, x_d) \in \mathcal{X}^d$ is shattered by $\mathcal{H}$, if $\forall\, (y_1, \ldots, y_d) \in \{0,1\}^d$, $\exists h \in \mathcal{H}$, such that $\forall i \in [d]$, $h(x_i) = y_i$. The VC dimension of $\mathcal{H}$, denoted $\mathrm{VC}(\mathcal{H})$, is the largest number $d \in \mathbb{N}$ such that there exists a sequence $(x_1, \ldots, x_d) \in \mathcal{X}^d$ that is shattered by $\mathcal{H}$. If there exists shattered sequences of arbitrarily large length $d \in \mathbb{N}$, then we say that $\mathrm{VC}(\mathcal{H}) = \infty$.*

In the online learning model, no distributional assumptions are placed (Littlestone, 1987; Ben-David et al., 2009). Instead, an adversary plays a sequential game with the learner over $T \in \mathbb{N}$ rounds. Before the game begins, the adversary selects a sequence of examples $x_1, x_2, \ldots, x_T$ and a hypothesis $h \in \mathcal{H}$. Then, in each round $t \in [T]$, the reveals first reveals x_t to the learner, the learner makes a prediction $\hat{y}_t \in \{0,1\}$, the adversary reveals the true label $h(x_t)$, and finally the learner suffers the loss $\mathbb{1}\{\hat{y}_t \neq h(x_t)\}$.The goal of the learner is to output predictions $\hat{y}_t$ such that its cumulative number of mistakes is "small."

Definition 30 (Online Learnability) *A hypothesis class $\mathcal{H}$ is online learnable if there exists an algorithm $\mathcal{A}$ and sublinear function $\mathrm{R} : \mathbb{N} \to \mathbb{N}$ such that for any $T \in \mathbb{N}$, any sequence of examples $x_1, \ldots, x_T$, and any $h \in \mathcal{H}$, the algorithm outputs $\hat{y}_t \in \{0,1\}$ at every time point $t \in [T]$ such that*

$$\sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq h(x_t)\} \leq \mathrm{R}(T).$$

The online learnability of a hypothesis class $\mathcal{H}$ is characterized by the finiteness of a different combinatorial parameter called the Littlestone dimension (Littlestone, 1987). To define the Littlestone dimension, we first need to define a Littlestone tree and an appropriate notion of shattering.

Definition 31 (Littlestone tree) *A Littlestone tree of depth d is a* complete *binary tree of depth d where the internal nodes are labeled by examples of $\mathcal{X}$ and the left and right outgoing edges from each internal node are labeled by 0 and 1 respectively.*

Given a Littlestone tree $\mathcal{T}$ of depth d, a root-to-leaf path down $\mathcal{T}$ is a bitstring $\sigma \in \{0,1\}^d$ indicating whether to go left ($\sigma_i = 0$) or to go right ($\sigma_i = 1$) at each depth $i \in [d]$. A path $\sigma \in \{0,1\}^d$ down $\mathcal{T}$ gives a sequence of labeled examples $\{(x_i, \sigma_i)\}_{i=1}^d$, where x_i is the example labeling the internal node following the prefix $(\sigma_1, \ldots, \sigma_{i-1})$ down the tree. A hypothesis $h_\sigma \in \mathcal{H}$ shatters a path $\sigma \in \{0,1\}^d$, if for every $i \in [d]$, we have $h_\sigma(x_i) = \sigma_i$. In other words, h_σ is consistent with the labeled examples when following σ. A Littlestone tree $\mathcal{T}$ is *shattered* by $\mathcal{H}$ if for every root-to-leaf path σ down $\mathcal{T}$, there exists a hypothesis $h_\sigma \in \mathcal{H}$ that shatters it. Using this notion of shattering, we define the Littlestone dimension of a hypothesis class.

Definition 32 (Littlestone dimension) *The Littlestone dimension of $\mathcal{H}$, denoted $\mathrm{L}(\mathcal{H})$, is the largest $d \in \mathbb{N}$ such that there exists a Littlestone tree $\mathcal{T}$ of depth d shattered by $\mathcal{H}$. If there exists shattered Littlestone trees $\mathcal{T}$ of arbitrary large depth, then we say that $\mathrm{L}(\mathcal{H}) = \infty$.*

It is well known that online learnability implies PAC learnability, but not the other way around. That is, for every $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, we have that $\mathrm{L}(\mathcal{H}) \geq \mathrm{VC}(\mathcal{H})$, and that the inequality can be strict. Our definitions of PAC and online learnability are uniform in nature, as is standard in learning theory literature (Shalev-Shwartz and Ben-David, 2014). There are also non-uniform and "in-the-limit" versions of PAC and online learnability. However, we will not be concerned with them in this paper and will not make explicit the distinction between uniform and non-uniform predictability. We refer the reader Chapter 7 in Shalev-Shwartz and Ben-David (2014) and Lu (2023) for more details about the non-uniform versions of PAC and online learnability respectively and Malliaris and Moran (2022) for an "in-the-limit" version of PAC learnability (termed PEC learnability).

Appendix B. Existing Results in Identification and Generation

In this section, we restate the results for language identification and generation in learning theory notation. Gold (1967), Angluin (1979), and Angluin (1980) studied the problem of identification in the context of language modeling. In our notation, they showed that many natural hypothesis classes are *not* identifiable in the limit according to Definition 27. This result is often interpreted as a hardness result – identification in the limit is *impossible* in full generality, even for some natural countable classes.

Theorem 33 (Gold (1967); Angluin (1979, 1980)) *Let $\mathcal{X}$ be countable. There exists a countable $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ which is not identifiable in the limit.*

In particular, Angluin (1980) provides a precise characterization of which classes are identifiable in the limit. Roughly, the condition states that every language L must have a "tell-tale" finite subset of strings $S \subset L$ such that any other language L' which also contains S cannot be a proper subset of L. Theorem 34 restates this condition in the notation of this paper.

Theorem 34 (Theorem 1 in Angluin (1980)) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any hypothesis class. Then $\mathcal{H}$ is identifiable in the limit if and only if for every $h \in \mathcal{H}$, there exists $S \subseteq \operatorname{supp}(h)$ such that:*

(i) $|S| < \infty$.

(ii) $\forall h' \in \mathcal{H}$, $S \subseteq \operatorname{supp}(h') \implies \operatorname{supp}(h') \not\subset \operatorname{supp}(h)$.

On the other hand, Kleinberg and Mullainathan (2024) recently show that this is not the case for generatability in the limit – all *countable* $\mathcal{H}$ that satisfy the UUS property are generatable in the limit!

Theorem 35 (Theorem 4.1 in Kleinberg and Mullainathan (2024)) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. If $\mathcal{H}$ is countable and satisfies the* UUS *property, then $\mathcal{H}$ is generatable in the limit.*

In addition, they also prove that finite hypothesis classes satisfy the stronger notion of uniform generatability.

Theorem 36 (Theorem 2.2 in Kleinberg and Mullainathan (2024)) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. If $\mathcal{H}$ is finite and satisfies the* UUS *property, then $\mathcal{H}$ is* uniformly *generatable.*

Unfortunately, Kleinberg and Mullainathan (2024) do not give a full characterization of which classes are uniformly generatable, non-uniformly generatable, and generatable in the limit. In this paper, we are interested in closing these gaps. In particular, we are interested in identifying necessary and sufficient conditions under which a hypothesis class $\mathcal{H}$ is uniformly and non-uniformly generatable. In learning theory, such conditions are often derived in terms of *combinatorial dimensions*, which are mappings

$$\dim : 2^{\{0,1\}^{\mathcal{X}}} \to \mathbb{N} \cup \{0, \infty\}$$

such that $\dim(\mathcal{H})$ measures an appropriate notion of expressivity of a class $\mathcal{H}$. For example, the PAC learnability of a binary hypothesis class is fully characterized by the finiteness of the VC dimension (Vapnik and Chervonenkis, 1974). Similarly, the online learnability of a binary hypothesis class is characterized by the finiteness of its Littlestone dimension (Littlestone, 1987). Are there analogous dimensions that characterize uniform, non-uniform generatability, and generatability in the limit? These questions are the main focus of this paper.

Appendix C. Sample Complexity for Uniform Generation

In this section, we formalize the intuition in Section 2 regarding a notion of "sample complexity" for uniform generation.

Definition 37 (Uniform Generation Sample Complexity) *Given a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ and a generator $\mathcal{G}$, the* uniform generation sample complexity *of a generator $\mathcal{G}$ is the smallest number $d_{\mathcal{G}} \in \mathbb{N}$, such that $\mathcal{G}$ perfectly generates according to Definition 3 after it observes $d_{\mathcal{G}}$ unique positive examples. If no such number exists, we set $d_{\mathcal{G}} = \infty$.*

Appendix D. Proofs for Uniform Generation

D.1. Proof of Lemma 7

Proof Let $\mathcal{G}$ be any generator and suppose $\mathrm{C}(\mathcal{H}) = \infty$. We need to show that for every $d \in \mathbb{N}$, there exists a $h^\star \in \mathcal{H}$ and a sequence $(x_i)_{i \in \mathbb{N}}$ with $\{x_1, x_2, \dots\} \subseteq \mathrm{supp}(h^\star)$ such that for every time point $t \in \mathbb{N}$ where $|\{x_1, \dots, x_t\}| = d$, there exists $s \geq t$ such that $\mathcal{G}(x_{1:s}) \notin \mathrm{supp}(h^\star) \setminus \{x_1, \dots, x_s\}$.

To that end, fix a $d \in \mathbb{N}$. Since $\mathrm{C}(\mathcal{H}) = \infty$, we know that there exists some $d^\star \geq d$ and distinct $z_1, \dots, z_{d^\star}$ such that $\mathcal{H}(z_1, \dots, z_{d^\star}) \neq \bot$ and $|\langle z_1, \dots, z_{d^\star} \rangle_{\mathcal{H}}| < \infty$. Since $\mathcal{H}(z_{1:d^\star}) \subseteq \mathcal{H}(z_{1:d})$, we also know that $|\langle z_1, \dots, z_d \rangle_{\mathcal{H}}| < \infty$.

Let $p := |\langle z_1, \dots, z_d \rangle_{\mathcal{H}}|$. Note that for every $x \in \mathcal{X} \setminus \langle z_1, \dots, z_d \rangle_{\mathcal{H}}$, there exists a $h \in \mathcal{H}(\langle z_1, \dots, z_d \rangle_{\mathcal{H}})$ such that $x \notin \mathrm{supp}(h)$. Let $\hat{x}_p = \mathcal{G}(\langle z_1, \dots, z_d \rangle_{\mathcal{H}})$ denote the prediction of $\mathcal{G}$ when given as input $\langle z_1, \dots, z_d \rangle_{\mathcal{H}}$ sorted in its natural order. Without loss of generality suppose that $\hat{x}_p \notin \langle z_1, \dots, z_d \rangle_{\mathcal{H}}$. Then, using the previous observation, there exists $h^\star \in \mathcal{H}(\langle z_1, \dots, z_d \rangle_{\mathcal{H}})$ such that $\hat{x}_p \notin \mathrm{supp}(h^\star) \setminus \langle z_1, \dots, z_d \rangle_{\mathcal{H}}$. Pick this $h^\star$ and consider the stream $x_1, x_2, \dots$ by sorting $\langle z_1, \dots, z_d \rangle_{\mathcal{H}}$ in its natural ordering and then appending the stream $x_{p+1}, x_{p+2}, \dots \subseteq \mathrm{supp}(h^\star)$ such that $\langle z_1, \dots, z_d \rangle_{\mathcal{H}} \cap \bigcup_{i=p+1}^{\infty} \{x_i\} = \emptyset$.

It remains to show that for every time point $t \in \mathbb{N}$ where $|\{x_1, \dots, x_t\}| = d$, there exists $s \geq t$ such that $\mathcal{G}(x_{1:s}) \notin \mathrm{supp}(h^\star) \setminus \{x_1, \dots, x_s\}$. By definition, we know that $x_1, \dots, x_d$ are distinct. Thus, when $t = d$, we have that $|x_1, \dots, x_t| = d$. Moreover, this is the only such time point. Accordingly, it suffices to show that there exists $s \geq t$ such that $\mathcal{G}(x_{1:s}) \notin \mathrm{supp}(h^\star) \setminus \{x_1, \dots, x_s\}$. However, by definition, we know that $\mathcal{G}(x_{1:p}) \notin \mathrm{supp}(h^\star) \setminus \{x_1, \dots, x_p\}$. Since $p \geq d$ and $d \in \mathbb{N}$ was chosen arbitrarily, our proof is complete. ■

D.2. Proof of Corollary 10

Proof Let $\mathcal{X} = \mathbb{Z}$ and $\mathcal{H} = \{x \mapsto \mathbb{1}\{x \in A \text{ or } x \leq 0\} : A \in 2^{\mathbb{N}}\}$. It is not hard to see that $\mathcal{H}$ satisfies the UUS property. Moreover, since $2^{\mathbb{N}}$ is uncountably large, so is $\mathcal{H}$. Finally, to see that $\mathcal{H}$ is uniformly generatable, note that for every $x \in \mathbb{Z}$, we have that $\langle x \rangle_{\mathcal{H}} = \mathbb{Z}_{\leq 0}$. Thus, $\mathrm{C}(\mathcal{H}) = 0$ and $\mathcal{H}$ is trivially uniformly generatable. ■

Appendix E. Proofs for Non-uniform Generation

E.1. Proof of Theorem 11

We prove Theorem 11 across two lemmas, starting with the necessity direction.

Lemma 38 (Necessity in Theorem 11) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any class satisfying the* UUS *property. If $\mathcal{H}$ is non-uniformly generatable, then there exists a sequence of* non-decreasing, uniformly generatable *classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots$ such that $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$.*

Proof Suppose $\mathcal{H}$ is non-uniformly generatable and $\mathcal{G}$ is a non-uniform generator for $\mathcal{H}$. For every $h \in \mathcal{H}$, let $d_h \in \mathbb{N}$ be the smallest natural number such that for any sequence $x_1, x_2, \ldots$ with $\{x_1, x_2, \ldots\} \subset \text{supp}(h)$, if there exists a $t \in \mathbb{N}$ such that $|\{x_1, \ldots, x_t\}| = d_h$, then $\mathcal{G}(x_{1:s}) \in \text{supp}(h) \setminus \{x_1, \ldots, x_s\}$ for all $s \geq t$. Let $\mathcal{H}_n := \{h \in \mathcal{H} : d_h \leq n\}$ for all $n \in \mathbb{N}$. Then, by the definition of $\mathcal{G}$, we know for every $n \in \mathbb{N}$, $\mathcal{G}$ is a uniform generator for $\mathcal{H}_n$, and therefore $\mathcal{H}_n$ is uniformly generatable. The proof is complete after noting that $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots$ and $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$.
■

The next lemma shows that the condition in Theorem 11 is sufficient. Our proof is constructive and by a reduction – given uniform generators $\mathcal{G}_1, \mathcal{G}_2, \ldots$ for $\mathcal{H}_1, \mathcal{H}_2, \ldots$ and their uniform generation sample complexities $d_{\mathcal{G}_1}, d_{\mathcal{G}_2}, \ldots$ [1] (see Definition 37), we construct a non-uniform generator $\mathcal{G}$ for $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$. This aligns with existing sufficiency proofs for non-uniform PAC and online learning (Shalev-Shwartz and Ben-David, 2014; Lu, 2023), which are also through reductions.

Lemma 39 (Sufficiency in Theorem 11) *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any class satisfying the* UUS *property. If there exists a sequence of* non-decreasing, uniformly generatable *classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots$ such that $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$, then $\mathcal{H}$ is non-uniformly generatable.*

Proof Suppose $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is a class satisfying the UUS property for which there exists a sequence of non-decreasing, uniformly generatable classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots$ with $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}_n$. By definition of uniform generatability, for every $n \in \mathbb{N}$, there exists a uniform generator $\mathcal{G}_n$ for $\mathcal{H}_n$. Let $d_{\mathcal{G}_n}$ denote the uniform generation sample complexity of $\mathcal{G}_n$ with respect to $\mathcal{H}_n$.

Consider the following generator $\mathcal{G}$. Fix $t \in \mathbb{N}$ and consider any sequence $\{x_1, \ldots, x_t\}$ such that $|\mathcal{H}(x_1, \ldots, x_t)| \geq 1$. Let $d_t := |\{x_1, \ldots, x_t\}|$ be the number of unique examples up to and including time point t. $\mathcal{G}$ first computes $n_t = \max\{n \in [t] : d_{\mathcal{G}_n} \leq d_t\} \cup \{0\}$. If $n_t = 0$, $\mathcal{G}$ plays any $\hat{x}_t \in \mathcal{X}$. If $n_t > 0$, $\mathcal{G}$ uses $\mathcal{G}_{n_t}$ to generate new instances, which means $\mathcal{G}(x_1, \ldots, x_t) = \mathcal{G}_{n_t}(x_1, \ldots, x_t)$.

1. One actually only needs an upper bound on the uniform generation sample complexities.

We now prove that such a $\mathcal{G}$ is a non-uniform generator for $\mathcal{H}$. To that end, let $h^\star$ be the hypothesis chosen by the adversary and suppose that $h^\star$ belongs to $\mathcal{H}_{n^\star}$. Let $d^\star = \max\{d_{\mathcal{G}_{n^\star}}, n^\star\}$. We show that for any sequence $x_1, x_2, \dots$ with $\{x_1, x_2, \dots\} \subseteq \text{supp}(h^\star)$, if there exists $t^\star \in \mathbb{N}$ such that $|\{x_1, \dots, x_{t^\star}\}| = d^\star$, then $\mathcal{G}(x_{1:s}) \in \text{supp}(h) \setminus \{x_1, \dots, x_s\}$ for all $s \geq t^\star$. Fix any valid sequence $x_1, x_2, \dots \subseteq \text{supp}(h^\star)$, and suppose, without of loss of generality, that $|\{x_1, \dots, x_{t^\star}\}| = d^\star$ for some $t^\star \in \mathbb{N}$. Fix any $s \geq t^\star$. By definition, $\mathcal{G}$ first computes $n_s = \max\{n \in [s] : d_{\mathcal{G}_n} \leq d_s\} \cup \{0\}$. Note that $n_s \geq n^\star$ since $s \geq n^\star$ and $d_{\mathcal{G}_{n^\star}} \leq d_s$. Thus, $|\mathcal{H}_{n_s}(x_{1:s})| \geq 1$ since $h^\star \in \mathcal{H}_{n_s}$. Accordingly, by construction of $\mathcal{G}$, it uses $\mathcal{G}_{n_s}$ to generate a new instance. The proof is complete by noting that $h^\star \in \mathcal{H}_{n_s}$ and $d_s \geq d_{\mathcal{G}_{n_s}}$ which guarantees that $\mathcal{G}_{n_s}(x_1, \dots, x_s) \in \text{supp}(h^\star) \setminus \{x_1, \dots, x_s\}$. ■

Since only upper bounds on the uniform generation sample complexities of $\mathcal{G}_1$, $\mathcal{G}_2, \dots$ are needed in the proof of Lemma 11, the algorithm in the proof of Lemma 11 can be efficiently implemented as long as each $\mathcal{G}_i$ is efficient. This is because in each round $t \in \mathbb{N}$, the number n_t can be efficiently computed if the sample complexities $d_{\mathcal{G}_1}, d_{\mathcal{G}_2}, \dots$ are non-decreasing. However, even if the sample complexities $d_{\mathcal{G}_1}, d_{\mathcal{G}_2}, \dots$ are not presented in non-decreasing order, we can run the algorithm on a new sequence of sample complexities $d'_{\mathcal{G}_1}, d'_{\mathcal{G}_2}, \dots$ such that $d'_{\mathcal{G}_1} = d_{\mathcal{G}_1}$ and $d'_{\mathcal{G}_i} = \max\{d'_{\mathcal{G}_{i-1}}, d_{\mathcal{G}_i}\}$ for all $i \geq 2$.

E.2. Proof of Corollary 12

Proof Suppose $\mathcal{H}$ is a countable hypothesis class satisfying the UUS property. Consider an arbitrary enumeration $h_1, h_2, \dots$ of $\mathcal{H}$. Let $\mathcal{H}_n = \{h_1, \dots, h_n\}$ for all $n \in \mathbb{N}$. Then, $\mathcal{H}_1, \mathcal{H}_2, \dots$ is a non-decreasing sequence of classes such that $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$. Moreover, since for every $n \in \mathbb{N}$, we have that $|\mathcal{H}_n| = n < \infty$, Theorem 36 gives that $\mathcal{H}_n$ is uniformly generatable, completing the proof. ■

Appendix F. Proofs for Generatability in the Limit

Proof (of Corollary 14) Note that $\mathrm{C}(\mathcal{H}_i) = 0$ for all $i \in [n]$. Thus, Theorem 13 gives that $\mathcal{H}$ is generatable in the limit. ■

Appendix G. Weaker Sufficiency Conditions for Generatability in the Limit

In this section, we prove a weaker sufficiency condition than the one in Theorem 13 for generatability in the limit. Before we present the main result, we define a new property of a class termed the Eventually Unbounded Closure property.

Definition 40 (Eventually Unbounded Closure) *A class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ has the* Eventually Unbounded Closure (EUC) *property if for every $h \in \mathcal{H}$ and any enumeration of its support $x_1, x_2, \dots,$ there exists a $t \in \mathbb{N}$ such that $|\langle x_1, x_2, \dots, x_t \rangle_{\mathcal{H}}| = \infty$*

Because the closure with respect to a class $\mathcal{H}$ does not depend on any one particular hypothesis, Definition 40 is really only a stream dependent property. That is, an equivalent representation of Definition 40 is as follows – $\mathcal{H}$ satisfies the EUC property if and only if for every sequence of examples $x_1, x_2, \cdots$ there exists a $t \in \mathbb{N}$ such that $|\langle x_1, \dots, x_t \rangle_{\mathcal{H}}| = \infty$ or $\langle x_1, \dots, x_t \rangle_{\mathcal{H}} = \perp$.

The EUC property is sufficient for generation in the limit. Indeed, just consider the generator that plays arbitrarily until the closure is unbounded, after which it only plays from this infinite set. Since the EUC property guarantees that the closure will become infinite in finite time, this is a valid generator. Moreover, note that such a generator is eventually auto regressive – once the closure is infinite in size, the generator no longer needs to observe positive examples to generate new, unseen examples in the future. One might be tempted to think that the EUC property is also necessary for generatability in the limit. However, the following lemma shows that this is not the case. For an infinite bit string $b \in \{0,1\}^{\mathbb{N}}$, let $|b|$ denote the number of 1's.

Lemma 41 *Let $\mathcal{X}$ be countable. There exists a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ such that $\mathcal{H}$ satisfies the* UUS *property, is non-uniformly generatable, but does not satisfy the* EUC *property.*

Proof Let $\mathcal{X} = \mathbb{N}$ and $\{p_n\}_{n \in \mathbb{N}}$ be the sequence of prime numbers. Consider the class $\mathcal{H} := \{h_b : b \in \{0,1\}^{\mathbb{N}}, |b| < \infty\}$ where h_b is defined such that $\text{supp}(h_b) := \{p_n^{1+\sum_{i=1}^{n} b_i}\}_{n \in \mathbb{N}}$. Note that $\mathcal{H}$ satisfies the UUS property. Moreover, $\mathcal{H}$ is countable since the collection of all countably infinite bit strings with finite size is countable. Thus, by Corollary 12, $\mathcal{H}$ is non-uniformly generatable. Finally, to see that $\mathcal{H}$ does not satisfy the EUC property, observe that for every finite sequence of prime powers, its closure can only contain this sequence itself. To see why, note that for any prime not in the sequence and any power, one can always construct a hypothesis which contains the finite sequence, but not that prime power. ■

That said, we can use the EUC property to weaken our sufficiency condition in Theorem 13 by replacing finite Closure dimension with the EUC property.

Theorem 42 *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any class satisfying the* UUS *property. If there exists a finite sequence of classes $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_n$, all satisfying the* EUC *property, such that $\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{H}_i$, then $\mathcal{H}$ is generatable in the limit.*

Theorem 42 replaces the constraint that each of the finite number of sub-classes need to be uniformly generatable with the constraint that they need to satisfy the EUC property. This is a weakening as uniform generatability implies EUC but not the other way around. The proof is effectively the same as the proof of Theorem 13 with the only difference being that the amount of time before which the generator computes closures is now stream dependent.

Proof (sketch of Theorem 42) Let $\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{H}_i$ be such that $\mathcal{H}_i$ satisfies the EUC property for all $i \in [n]$. Consider the following generator $\mathcal{G}$. On a valid input sequence $x_1, x_2, \ldots$, let $t_i \in \mathbb{N}$ be the smallest time point such that either $\langle x_1, \ldots, x_{t_i} \rangle_{\mathcal{H}_i} = \perp$ or $|\langle x_1, \ldots, x_{t_i} \rangle_{\mathcal{H}_i}| = \infty$ for all $i \in [n]$. Note that such an $t_i \in \mathbb{N}$ must exist because $\mathcal{H}_i$ satisfies the EUC property. Let $t^\star = \max_{i \in [n]} t_i$ be the largest time point. $\mathcal{G}$ plays arbitrarily up to, but not including, time point $t^\star$. On time point $t^\star$, $\mathcal{G}$ computes $\langle x_1, \ldots, x_{t_i} \rangle_{\mathcal{H}_i}$ for all $i \in [n]$. Let $S \subseteq [n]$ be the subset of indices such that $i \in S$ if and only if $\langle x_1, \ldots, x_{t_i} \rangle_{\mathcal{H}_i} \neq \perp$. For every $i \in S$, let $(z_j^{(i)})_{j \in \mathbb{N}}$ be the natural ordering of $\langle x_1, \ldots, x_{t_i} \rangle_{\mathcal{H}_i}$, which is guaranteed to exist since $\mathcal{X}$ is countable. For every $t \geq t^\star$, valid sequence of revealed examples $x_1, \ldots, x_t$, and $i \in S$, $\mathcal{G}$ computes

$$n_t^i := \max\{n \in \mathbb{N} : \{z_1^{(i)}, \ldots, z_n^{(i)}\} \subset \{x_1, \ldots, x_t\}\} \tag{1}$$

and $i_t \in \arg\max_{i \in S} n_t^i$. Finally, $\mathcal{G}$ plays any $\hat{x}_t \in \langle x_1, \ldots, x_{t_i} \rangle_{\mathcal{H}_{i_t}} \setminus \{x_1, \ldots, x_t\}$. The rest of the proof is identical to that of Theorem 13. ■

Due to its connection to autoregressive generation in the limit, understanding which classes satisfy the EUC property is an interesting property on its own. For example, while it is clear that the EUC property is weaker than uniform generatability but stronger than generatability in the limit, its relationship to non-uniform generatability is less clear. Lemma 41 shows that if one restricts to countable classes, then the EUC property is strictly stronger than non-uniform generatability – there exists a countable class which does not have the EUC property. We leave as an open question whether the EUC property is strictly stronger than non-uniform generatability even amongst uncountable classes.

Question 43 *Does there exists an uncountable class $\mathcal{H}$ which is non-uniformly generatable, but does not satisfy the* EUC *property?*

We can also provide a sufficient condition for generatability in the limit akin to that of non-uniform generatability in terms of the EUC property.

Theorem 44 *Let $\mathcal{X}$ be countable and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be any class satisfying the* UUS *property. If there exists a non-decreasing sequence of classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots$, all satisfying the* EUC *property, such that $\mathcal{H} = \bigcup_{i=1}^{\infty} \mathcal{H}_i$, then $\mathcal{H}$ is generatable in the limit.*

Note the sufficiency condition in Theorem 44 is weaker than the sufficiency condition for non-uniform generation as uniform generatability is stronger than EUC. Our proof of Theorem 44 is constructive – we give an algorithm which generates in the limit as long as the sufficiency condition is met. The algorithm can be thought of as a generalization of the algorithm by Kleinberg and Mullainathan (2024) for countable classes. The high-level idea is to play from the closure of rightmost class whose closure is infinite in size.

Algorithm 1 Generator

Input: Hypothesis class $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}_n$ such that $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \ldots$ and $\mathcal{H}_n$ satisfies EUC for all $n \in \mathbb{N}$

for $t = 1, 2, \ldots$ **do**

  Adversary reveals positive example x_t

  Let $n_t = \max\{n \in [t] : |\langle x_1, \ldots, x_t \rangle_{\mathcal{H}_n}| = \infty\} \cup \{0\}$

  if $n_t = 0$ **then**

    Play arbitrarily from $\mathcal{X}$

  else

    Play arbitrarily from $\langle x_1, \ldots, x_t \rangle_{\mathcal{H}_{n_t}} \setminus \{x_1, \ldots, x_t\}$

end

Proof We will show that Algorithm 1 generates in the limit. To that end, let $h^\star \in \mathcal{H}$ be the hypothesis and $x_1, x_2, \ldots$ be an enumeration of $\operatorname{supp}(h^\star)$ chosen by the adversary. Let $n^\star \in \mathbb{N}$ be the smallest number such that $h^\star \in \mathcal{H}_{n^\star}$. For every $n \in \mathbb{N}$, since $\mathcal{H}_n$ satisfies the EUC property, there exists a $t_n \in \mathbb{N}$ such that either $|\langle x_1, \ldots, x_{t_n} \rangle_{\mathcal{H}_n}| = \infty$ or $\langle x_1, \ldots, x_{t_n} \rangle_{\mathcal{H}_n} = \perp$.

We claim that for all $n \geq n^\star$, we have that $|\langle x_1, \ldots, x_s \rangle_{\mathcal{H}_n}| = \infty$ and $\langle x_1, \ldots, x_s \rangle_{\mathcal{H}_n} \subseteq \operatorname{supp}(h^\star)$ for all $s \geq t_n$. Fix some $n \geq n^\star$. By definition, we know that $h^\star \in \mathcal{H}_n$. In addition, since the stream $x_1, x_2, \ldots$ is an enumeration of $h^\star$ and $\mathcal{H}_n$ satisfies the EUC property, it must be the case that $|\langle x_1, \ldots, x_{t_n} \rangle_{\mathcal{H}_n}| = \infty$. Moreover, $\langle x_1, \ldots, x_{t_n} \rangle_{\mathcal{H}_n} \subseteq \operatorname{supp}(h^\star)$ because $h^\star \in \mathcal{H}_n$. Now, fix some $s \geq t_n$. Then, it must be the case that $\langle x_1, \ldots, x_s \rangle_{\mathcal{H}_n} \supseteq \langle x_1, \ldots, x_{t_n} \rangle_{\mathcal{H}_n}$ and therefore $|\langle x_1, \ldots, x_s \rangle_{\mathcal{H}_n}| = \infty$. Because $h^\star \in \mathcal{H}_n$, it also must be the case that $\langle x_1, \ldots, x_s \rangle_{\mathcal{H}_n} \subseteq \text{supp}(h^\star)$. This completes the proof of the claim as $n \geq n^\star$ was chosen arbitrarily.

Now, we complete the overall proof of Theorem 44 by showing that Algorithm 1 generates perfectly on and after round $t_{n^\star}$. On round $t = t_{n^\star}$, $n_t = n^\star$, and thus by Line 7, Algorithm 1 generates from $\text{supp}(h^\star) \setminus \{x_1, \ldots, x_t\}$ since $\langle x_1, \ldots, x_t \rangle_{\mathcal{H}_{n^\star}} \subseteq \text{supp}(h^\star)$. Now fix a round $t > t_{n^\star}$. Then, observe that by the claim above we have that $n_t \geq n^\star$ and $h^\star \in \mathcal{H}_{n_t}$. Accordingly, we have that $\langle x_1, \ldots, x_t \rangle_{\mathcal{H}_{n_t}} \subseteq \text{supp}(h^\star)$ and therefore by Line 7, Algorithm 1 plays from $\text{supp}(h^\star) \setminus \{x_1, \ldots, x_t\}$. Since $t > t_{n^\star}$ was picked arbitrarily, the proof is complete. ∎

We note that in addition to Theorem 11, Theorem 44 also recovers the result by Kleinberg and Mullainathan (2024) that all countable classes are generatable in the limit.

Appendix H. Proof of Proposition 5

We prove Proposition 5 over two lemmas. The first shows that uniform generation is strictly harder than non-uniform generation.

Lemma 45 (Uniform Generatability $\neq$ Non-uniform Generatability) *Let $\mathcal{X}$ be countable. There exists a countable class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ that satisfies the* UUS *property, is non-uniformly generatable, but not uniformly generatable.*

Proof Let $\mathcal{X} = \mathbb{Z}$. Let E denote the set of all even negative integers and O the set of all odd negative integers. Consider the hypothesis classes

$$\mathcal{H}^e = \left\{ x \mapsto \mathbb{1}\left\{ x \in \left\{ \frac{d(d-1)}{2} + 1, \ldots, \frac{d(d-1)}{2} + d \right\} \text{ or } x \in E \right\} : d \in \mathbb{N} \right\} \tag{2}$$

and

$$\mathcal{H}^o = \left\{ x \mapsto \mathbb{1}\left\{ x \in \left\{ \frac{d(d-1)}{2} + 1, \ldots, \frac{d(d-1)}{2} + d \right\} \text{ or } x \in O \right\} : d \in \mathbb{N} \right\} \tag{3}$$

and define $\mathcal{H} = \mathcal{H}^e \cup \mathcal{H}^o$. First, its not too hard to see that $\mathcal{H}$ satisfies the UUS. Second, we claim that $\mathrm{C}(\mathcal{H}) = \infty$. To see why, we need to show that for every $d \in \mathbb{N}$, there exists a $d^\star \geq d$ and a distinct sequence of examples $x_1, \ldots, x_{d^\star}$, such that $|\mathcal{H}(x_1, \ldots, x_{d^\star})| \geq 1$ and

$$|\langle x_1, \ldots, x_{d^\star} \rangle_{\mathcal{H}}| < \infty.$$

To that end, pick any $d \in \mathbb{N}$ and let $d^\star = d$. Consider the sequence of examples $x_1 = \frac{d(d-1)}{2} + 1, \ldots, x_d = \frac{d(d-1)}{2} + d$. First, observe that this is a sequence of $d = d^\star$ distinct examples. Then,

$$\langle x_1, \ldots, x_d \rangle_{\mathcal{H}} = \text{supp}(h_d^e) \cap \text{supp}(h_d^o) = \{x_1, \ldots, x_d\}$$

where we let

$$h_d^e := \mathbb{1}\left\{ x \in \left\{ \frac{d(d-1)}{2} + 1, \ldots, \frac{d(d-1)}{2} + d \right\} \text{ or } x \in E \right\},$$

and

$$h_d^o := \mathbb{1}\left\{x \in \left\{\frac{d(d-1)}{2}+1, \ldots, \frac{d(d-1)}{2}+d\right\} \text{ or } x \in O\right\}.$$

Thus, we have that

$$|\langle x_1, \ldots, x_d \rangle_{\mathcal{H}}| < \infty.$$

Since d was chosen arbitrarily, this is true for all $d \in \mathbb{N}$, implying that $\mathrm{C}(\mathcal{H}) = \infty$. Thus, by Theorem 9, we have that $\mathcal{H}$ is not uniformly generatable. To show that $\mathcal{H}$ is non-uniformly generatable, note that $\mathcal{H}$ is countable. Thus, Corollary 12 completes the proof. ■

The second lemma shows that non-uniform generation can be strictly harder than generation in the limit.

Lemma 46 (Non-uniform Generatability $\neq$ Generatability in the Limit) *Let $\mathcal{X}$ be countable. There exists a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ which satisfies the* UUS *property that is generatable in the limit but not non-uniformly generatable.*

Proof Let $\mathcal{X} = \mathbb{Z}$ and $\mathcal{H} = \{x \mapsto \mathbb{1}\{x \in A \text{ or } x \leq 0\} : A \in 2^{\mathbb{N}}\} \cup \{x \mapsto \mathbb{1}\{x \in \mathbb{N}\}\}$. Observe that $\mathcal{H}$ satisfies the UUS property. We first show that $\mathcal{H}$ is generatable in the limit. Let $\mathcal{G}$ be a generator such that for any valid sequence $\{x_1, \ldots, x_t\}$, if $x_1, \ldots, x_t$ are all positive, then $\mathcal{G}$ plays any $\hat{x}_t \in \mathbb{N} \setminus \{x_1, \ldots, x_t\}$. Otherwise, it plays any $\hat{x}_t \in \mathbb{Z}_{\leq 0} \setminus \{x_1, \ldots, x_t\}$. Now, suppose the adversary picks a $h^\star \in \mathcal{H}$ and an enumeration $x_1, x_2, \ldots$ of $\operatorname{supp}(h^\star)$. If $\operatorname{supp}(h^\star) = \mathbb{N}$, then by our construction of $\mathcal{G}$, we have that $\mathcal{G}(x_{1:s}) \in \operatorname{supp}(h^\star) \setminus \{x_1, \ldots, x_s\}$ for all $s \geq 1$. Otherwise if $\mathbb{Z}_{\leq 0} \subseteq \operatorname{supp}(h^\star)$, since $\bigcup_{i \in \mathbb{N}}\{x_i\} = \operatorname{supp}(h^\star)$, there exists a $t^\star \in \mathbb{N}$ such that $x_{t^\star} \leq 0$. Then, $\mathcal{G}(x_{1:s}) \in \mathbb{Z}_{\leq 0} \setminus \{x_1, \ldots, x_s\} \subseteq \operatorname{supp}(h^\star) \setminus \{x_1, \ldots, x_s\}$ for any $s \geq t^\star$. Thus, $\mathcal{G}$ is a valid generator and $\mathcal{H}$ is generatable in the limit.

Now, we will show that $\mathcal{H}$ is not non-uniformly generatable. Suppose for the sake of contradiction that $\mathcal{H}$ is non-uniformly generatable. Then, there exists a non-uniform generator $\mathcal{G}$ for $\mathcal{H}$. For every $h \in \mathcal{H}$, let $d_h \in \mathbb{N}$ be the smallest natural number such that for any sequence $(x_i)_{i \in \mathbb{N}}$ with $\{x_1, x_2, \ldots\} \subseteq \operatorname{supp}(h)$, if there exists a $t \in \mathbb{N}$ such that $|\{x_1, \ldots, x_t\}| = d_h$, then $\mathcal{G}(x_{1:s}) \in \operatorname{supp}(h) \setminus \{x_1, \ldots, x_s\}$ for all $s \geq t$. We now construct a $h \in \mathcal{H}$ such that $d_h \geq n, \forall n \in \mathbb{N}$, which leads to a contradiction.

Let $h_0 \in \mathcal{H}$ and $\operatorname{supp}(h_0) = \mathbb{N}$, then for any observed sequence $\{x_1, \ldots, x_t\} \subset \mathbb{N} = \operatorname{supp}(h_0)$ such that $|\{x_1, \ldots, x_t\}| \geq d_{h_0}$, we have that $\mathcal{G}(x_{1:t}) \in \mathbb{N} \setminus \{x_1, \ldots, x_t\}$. Now let $\{p_n\}_{n \in \mathbb{N}} = \{2, 3, 5, 7, \ldots\}$ be the set of all prime numbers. Let $h_1 \in \mathcal{H}$ be such that $\operatorname{supp}(h_1) = \{p_n\}_{n \in \mathbb{N}} \cup \mathbb{Z}_{\leq 0}$. Let $d_1 = \max(d_{h_1}, d_{h_0})$, then by definition, we have that

$$\mathcal{G}(\{2, \ldots, p_{d_1}\}) \in (\mathbb{N} \cap \{p_n\}_{n \in \mathbb{N}}) \setminus \{2, \ldots, p_{d_1}\} = \{p_n\}_{n \in \mathbb{N}} \setminus \{2, \ldots, p_{d_1}\}.$$

Let $h_2 \in \mathcal{H}$ be such that

$$\operatorname{supp}(h_2) = \{2, \ldots, p_{d_1}, p_{d_1+1}^2, p_{d_1+2}^2, \ldots\} \cup \mathbb{Z}_{\leq 0}.$$

Denote $d_2 = d_{h_2}$, then $d_2 \geq d_1 + 1$ since $\mathcal{G}(\{2, \ldots, p_{d_1}\}) \in (\mathbb{N} \cap \operatorname{supp}(h_1)) \setminus \{2, \ldots, p_{d_1}\}$, which means that,

$$\mathcal{G}(\{2, \ldots, p_{d_1}\}) \notin \operatorname{supp}(h_2) \setminus \{2, \ldots, p_{d_1}\}.$$

Let $h_3 \in \mathcal{H}$ such that

$$\operatorname{supp}(h_3) = \{2, \ldots, p_{d_1}, p^2_{d_1+1}, \ldots, p^2_{d_2}, p^3_{d_2+1}, p^3_{d_2+2}, \ldots\} \cup \mathbb{Z}_{\leq 0}.$$

Denote $d_3 = d_{h_3}$. Suppose, we observe

$$\{x_1, \ldots, x_{d_2}\} = \{2, \ldots, p_{d_1}, p^2_{d_1+1}, \ldots, p^2_{d_2}\}.$$

Then, it must be the case that $\mathcal{G}(x_{1:d_2}) \in \mathbb{N} \cap \operatorname{supp}(h_2) \setminus \{x_1, \ldots, x_{d_2}\}$. Since

$$(\mathbb{N} \cap \operatorname{supp}(h_2) \setminus \{x_1, \ldots, x_{d_2}\}) \cap \operatorname{supp}(h_3) = \emptyset,$$

we have $\mathcal{G}(x_{1:d_2}) \notin \operatorname{supp}(h_3) \setminus \{x_{1:d_2}\}$ and as a result $d_3 \geq d_2 + 1$. Inductively, suppose $h_1, h_2, \ldots, h_n$ and $d_1, d_2, \ldots, d_n$ are all defined. Let $h_{n+1} \in \mathcal{H}$ be such that

$$\operatorname{supp}(h_{n+1}) = \{2, \ldots, p_{d_1}, \ldots\ldots, p^n_{d_{n-1}+1}, \ldots, p^n_{d_n}, p^{n+1}_{d_n+1}, p^{n+1}_{d_n+2}, \ldots\ldots\} \cup \mathbb{Z}_{\leq 0}.$$

Let $d_{n+1} = d_{h_{n+1}}$. Then $d_{n+1} \geq d_n + 1$ since

$$\mathcal{G}(\{2, \ldots, p_{d_1}, \ldots\ldots, p^n_{d_{n-1}+1}, \ldots, p^n_{d_n}\}) \notin \operatorname{supp}(h_{n+1}).$$

Finally, let $h_\infty \in \mathcal{H}$ be such that

$$\operatorname{supp}(h_\infty) = \{2, \ldots, p_{d_1}, p^2_{d_1+1}, \ldots, p^2_{d_2}, p^3_{d_2+1}, \ldots, p^3_{d_3}, p^4_{d_3+1}, \ldots\ldots\} \cup \mathbb{Z}_{\leq 0}.$$

For every $t \in \mathbb{N}$, consider the sequence

$$\{x_1, \ldots, x_{d_t}\} = \{2, \ldots, p_{d_1}, p^2_{d_1+1}, \ldots, p^2_{d_2}, p^3_{d_2+1}, \ldots, p^t_{d_{t-1}+1}, \ldots, p^t_{d_t}\}.$$

Then, $\mathcal{G}(x_{1:d_t}) \in \mathbb{N} \cap \operatorname{supp}(h_t) \setminus \{x_1, \ldots, x_{d_t}\}$. Since

$$(\mathbb{N} \cap \operatorname{supp}(h_t) \setminus \{x_1, \ldots, x_{d_t}\}) \cap (\operatorname{supp}(h_\infty) \setminus \{x_1, \ldots, x_{d_t}\}) = \emptyset,$$

it must be the case that $d_{h_\infty} \geq d_t, \forall t \in \mathbb{N}$. Since $d_t \to \infty$ as $t \to \infty$, this leads to a contradiction, as we have found a hypothesis $h_\infty \in \mathcal{H}$ for which there is no uniform upper bound $d \in \mathbb{N}$ such that $\mathcal{G}$ perfectly generates new examples after observing d unique examples. ■

Appendix I. Remarks on Computability

I.1. Computing Closures

In learning-theoretic terms, $\langle x_1, \ldots, x_n \rangle_{\mathcal{H}}$ is the set of positive examples common to all hypothesis in the version space of $\mathcal{H}$ consistent with the sample $(x_1, 1), \ldots, (x_n, 1)$. From this perspective, one can check closure membership, i.e. given an example x and a sequence of examples $x_1, \ldots, x_n$, return $\mathbb{1}\{\langle x_1, \ldots, x_n \rangle_{\mathcal{H}} \neq \perp \text{ and } x \in \langle x_1, \ldots, x_n \rangle_{\mathcal{H}}\}$, using access to an Empirical Risk Minimization (ERM) oracle. Formally, an ERM oracle is a mapping $\mathcal{O} : 2^{\{0,1\}^{\mathcal{X}}} \times (\mathcal{X} \times \{0,1\})^\star \to \mathbb{N} \cup \{0\}$, which given a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ and a labeled sample $S \in (\mathcal{X} \times \{0,1\})^\star$, outputs $\min_{h \in \mathcal{H}} \sum_{(x,y) \in S} \mathbb{1}\{h(x) \neq y\}$. Then, given a class $\mathcal{H}$, a sequence of examples $x_1, \ldots, x_n$, one can compute $\mathbb{1}\{\langle x_1, \ldots, x_n \rangle_{\mathcal{H}} \neq \perp$ and $x \in \mathbb{1}\{\langle x_1, \ldots, x_n \rangle_{\mathcal{H}}\}$ using the following procedure. First, pass to $\mathcal{O}$ the sample $S = \{(x_1, 1), \ldots, (x_n, 1)\}$ and $\mathcal{H}$, and let r be its output. If $r \geq 1$, output 0. Otherwise, define the sample $S_x = \{(x_1, 1), \ldots, (x_n, 1), (x, 0)\}$. Query $\mathcal{O}$ on S_x and $\mathcal{H}$ and let r_x be its output. Output r_x. To see why the latter step works, suppose $r_x = 0$. Then, that means there exists a hypothesis $h \in \mathcal{H}$ such that $\{x_1, \ldots . x_n\} \subseteq \text{supp}(h)$ but $x \notin \text{supp}(h)$. Thus, it cannot be the case that $x \in \langle x_1, \ldots, x_n \rangle_{\mathcal{H}}$. On the other hand, if $r_x = 1$, then it must mean that for every $h \in \mathcal{H}$ such that $\{x_1, \ldots, x_n\} \subseteq \text{supp}(h)$, we have that $h(x) = 1$. Accordingly, $x \in \langle x_1, \ldots, x_n \rangle_{\mathcal{H}}$ by definition.

I.2. Uniform Generator

The generator in Lemma 8 can be efficiently implemented given access to the following max-min oracle $\mathcal{O}_{\textit{max-min}} : 2^{\{0,1\}^{\mathcal{X}}} \times \mathcal{X}^{\star} \to \mathcal{X}$. Given a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ and a finite sequence of examples $x_1, \ldots, x_t$, $\mathcal{O}_{\textit{max-min}}$ returns

$$\underset{x \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}}{\arg\max} \ \min_{h \in \mathcal{H}} \sum_{i=1}^{t} \mathbb{1}\{h(x_i) \neq 1\} + \mathbb{1}\{h(x) \neq 0\}.$$

The inner minimization is simply the output of an ERM oracle $\mathcal{O}_{ERM}$, whch given $\mathcal{H}$ and the sample $S \in (\mathcal{X} \times \{0,1\})^{\star}$, outputs the minimal empirical loss on S amongst all $h \in \mathcal{H}$. Thus, the output of $\mathcal{O}_{\textit{max-min}}$ can be equivalently written as:

$$\mathcal{O}_{\textit{max-min}}(\mathcal{H}, x_{1:t}) = \underset{x \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}}{\arg\max} \ \mathcal{O}_{ERM}(\mathcal{H}, \{(x_1, 1), \ldots, (x_t, 1), (x, 0)\}).$$

In fact, the generator in Lemma 8 can be implemented with a single call to the max-min oracle on every round. To see why, suppose that $t \geq \mathrm{C}(\mathcal{H}) + 1$. Then, since $|\langle x_1, \ldots, x_t \rangle_{\mathcal{H}}| = \infty$, it must be the case that $|\langle x_1, \ldots, x_t \rangle_{\mathcal{H}} \setminus \{x_1, \ldots, x_t\}| \neq \emptyset$ and for every $x \in \langle x_1, \ldots, x_t \rangle_{\mathcal{H}} \setminus \{x_1, \ldots, x_t\}$, we have that

$$\mathcal{O}_{ERM}(\mathcal{H}, \{(x_1, 1), \ldots, (x_t, 1), (x, 0)\}) \geq 1.$$

Accordingly,

$$\max_{x \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}} \mathcal{O}_{ERM}(\mathcal{H}, \{(x_1, 1), \ldots, (x_t, 1), (x, 0)\}) \geq 1$$

and therefore $\mathcal{O}_{\textit{max-min}}(\mathcal{H}, x_{1:t})$ returns an element $\hat{x}_t \in \mathcal{X} \setminus \{x_1, \ldots, x_t\}$ such that:

$$\mathcal{O}_{ERM}(\mathcal{H}, \{(x_1, 1), \ldots, (x_t, 1), (\hat{x}_t, 0)\}) \geq 1.$$

But, such an $\hat{x}_t$ must lie in $\langle x_1, \ldots, x_t \rangle_{\mathcal{H}}$, completing the proof. As a concluding remark, note that the max-min oracle should remind the reader of the min-max objective/two-player game used to motivate Generative Adversarial Networks (see Equation 1 in Goodfellow et al. (2014)). In particular, for our min-max oracle, one can think of the minimizer/ERM oracle as the discriminator and the outer maximizer as the generator.

I.3. Generator in the Limit

The algorithm in the proof of Theorem 13 can be efficiently implemented given access to an ERM oracle $\mathcal{O} : 2^{\{0,1\}^{\mathcal{X}}} \times (\mathcal{X} \times \{0,1\})^\star \to \mathbb{N} \cup \{0\}$ and an oracle $\mathcal{O}_{\mathrm{C}} : \mathcal{H} \to \mathbb{N}$ that can compute the Closure dimension. In particular, before the game begins, the algorithm uses $\mathcal{O}_{\mathrm{C}}$ to compute Closure dimension for each class. After this, the algorithm only needs to use finite calls to an ERM oracle in each round $t \in \mathbb{N}$. To see why, first note that the algorithm plays arbitrarily until time point $t^\star$ where $c+1$ distinct examples are revealed, where $c = \max_{i \in [n]} \mathrm{C}(\mathcal{H}_i)$ (which can be computed from n calls to $\mathcal{O}_{\mathrm{C}}$ at the start.) On round $t^\star$, the subset of indices $S \subseteq [n]$ for which the closure is not $\perp$ can be computed using at most n calls to $\mathcal{O}_{ERM}$ by the discussion in Appendix I.1. Next, note that for every $t \geq t^\star$, and every $i \in S$, the number n_t^i can be computed using on finite number of calls to a closure membership oracle (see Appendix I.1). Since closure membership can be computed using a single call to $\mathcal{O}_{ERM}$, the computation of n_t^i for all $i \in S$ can be computed using only a finite number of calls to $\mathcal{O}_{ERM}$. Finally, because $\hat{x}_t$ only needs a finite number of closure membership calls to compute, it can also be computed using a finite number of calls to $\mathcal{O}_{ERM}$.

Appendix J. Proofs for Generation vs. Prediction

The written form of Figure 1 is Theorem 47, whose proof we provde below.

Theorem 47 (Generation v. Prediction) *Let $\mathcal{X}$ be countable. The following statements are true.*

(i) There exists a countable class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ which is uniformly generatable *but not* PAC learnable.

(ii) There exists a countable class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ which is online learnable *but not* uniformly generatable.

(iii) There exists a countable class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ which is online learnable *and* uniformly generatable.

(iv) There exists a countable class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ that is PAC learnable *but neither* online learnable *nor* uniformly generatable.

(v) There exists a countable class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ that is PAC learnable *and* uniformly generatable*, but not* online learnable.

(vi) There exists a countable class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ that is neither PAC learnable *nor* uniformly generatable.

Proof (of (i) in Theorem 47) Let $\mathcal{X} = \mathbb{Z}$ and consider the hypothesis class $\mathcal{H} = \{x \mapsto \mathbb{1}\{x \in A \text{ or } x \leq 0\} : A \subset \mathbb{N}, |A| < \infty\}$. First, note that $\mathcal{H}$ satisfies the UUS since for every $x \in \mathbb{Z}_{\leq 0}$ and every $h \in \mathcal{H}$, we have that $h(x) = 1$. Second, its not too hard to see that $\mathrm{VC}(\mathcal{H}) = \infty$ since it can shatter arbitrary length sequence of examples of the form $x_1 = 1, x_2 = 2, \ldots, x_d = d$. Finally, observe that $\mathrm{C}(\mathcal{H}) = 0$ since $\langle x \rangle_{\mathcal{H}} = \mathbb{Z}_{\leq 0}$ for all $x \in \mathcal{X}$.

Such a separation also occurs for the more natural hypothesis class of convex polygons defined over the rationals. This result is also noted by Kleinberg and Mullainathan (2024) in Section 3.2, but we summarize it below.

Let $\mathcal{X} = \mathbb{Q}^2$ and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be the class of all convex polygons over $\mathcal{X}$. That is $\mathcal{H} := \{h \in \{0,1\}^{\mathcal{X}} : \text{supp}(h) \text{ is a convex polygon.}\}$. It is well known that $\text{VC}(\mathcal{H}) = \infty$. Moreover, its not too hard to see that $\mathcal{H}$ satisfies the UUS property. We now show that $\text{C}(\mathcal{H}) < \infty$. In fact, we can show that $\text{C}(\mathcal{H}) = 0$. Indeed, pick any $x \in \mathcal{X}$, and note that the set $\langle x \rangle_{\mathcal{H}}$ is a convex polygon since $\text{supp}(h)$ is a convex polygon for all $h \in \mathcal{H}$. Accordingly, we have that $|\langle x \rangle_{\mathcal{H}}| = \infty$, completing the proof. ■

Proof (of (ii) in Theorem 47) Let $\mathcal{X} = \mathbb{Z}$ and consider the same class $\mathcal{H}$ from Lemma 45. Recall, that $\text{C}(\mathcal{H}) = \infty$. Thus, it suffices to show that $\text{L}(\mathcal{H}) = 2$. First, we prove that $\text{L}(\mathcal{H}) \geq 2$ by showing that $\text{VC}(\mathcal{H}) \geq 2$. Indeed, it is not hard to verify that any x_1, x_2 such that $x_1 \in \mathbb{Z}_{<0}$ and $x_2 \in \mathbb{Z}_{>0}$ can be shattered by $\mathcal{H}$, completing this direction.

In fact, we can show that this is the only way that two examples can be shattered. Pick any two examples $x_1, x_2 \in \mathbb{Z}$. None of x_1, x_2 can be 0, as otherwise all hypothesis output 0 on this example. Thus, assume that x_1, x_2 and are non-zero. Our proof will now be in cases.

Suppose that $x_1, x_2 > 0$. For every $d \in \mathbb{N}$, define $A_d := \{\frac{d(d-1)}{2} + 1, , \ldots, \frac{d(d-1)}{2} + d\}$. Note that $A_1, A_2, \ldots$ are all pairwise disjoint. If there exists a $d \in \mathbb{N}$ such that x_1, x_2 lie in A_d, then the labeling $(1, 0)$ is not possible. If x_1, x_2 lie in different sets, then the labeling $(1, 1)$ is not possible. Thus, when $x_1, x_2 > 0$, they cannot be shattered by $\mathcal{H}$.

Suppose, $x_1, x_2 < 0$. If they are both even, then one cannot get the labeling $(1, 0)$. Likewise for odd. If one of them is even, say x_1 without loss of generality, then one cannot get the labeling $(1, 0)$. Likewise for odd. Thus, whenever $x_1, x_2 < 0$, they cannot be shattered by $\mathcal{H}$.

Thus, the only way x_1, x_2 can be shattered is if one of them is strictly negative, but the other is strictly positive.

Using this observation, we will now show that $\text{L}(\mathcal{H}) \leq 2$. Consider any Littlestone tree $\mathcal{T}$ of depth 3. We will show that $\mathcal{T}$ cannot be shattered by $\mathcal{H}$. There are two cases to consider.

Suppose the root node is labeled by a strictly negative integer. Then, using the above analysis it must be the case that the nodes on the second level must be labeled by strictly positive integers in order for $\mathcal{T}$ to be shattered. Now, pick any root-to-leaf path prefix $b = (1, 1)$ down $\mathcal{T}$. Let $(x_1, 1), (x_2, 1)$ denote the sequence of labeled examples obtained by traversing down $\mathcal{T}$ according to the prefix b. By our observation above, we know that $x_1 < 0$ and $x_2 > 0$. However, there can be exactly one hypothesis $h \in \mathcal{H}$ that is consistent with $(x_1, 1)$ and $(x_2, 1)$. Thus, any completion of the path b cannot be shattered by $\mathcal{H}$.

Suppose the root node is labeled by a strictly positive integer. Then, using the above analysis, it must be the case that the nodes on the second level must be labeled by strictly negative integers in order for $\mathcal{T}$ to be shattered. Now, pick the root-to-leaf path prefix $b = (1, 1)$ down $\mathcal{T}$. Let $(x_1, 1), (x_2, 1)$ denote the sequence of labeled examples obtained by traversing down $\mathcal{T}$ according to the prefix b. By our observation above, we know that $x_1 > 0$ and $x_2 < 0$. However, there can be exactly one hypothesis $h \in \mathcal{H}$ that can be consistent with $(x_1, 1)$ and $(x_2, 1)$. Thus, any completion of the path b cannot be shattered by $\mathcal{H}$. Since $\mathcal{T}$ was arbitrary, it must be the case that $\text{L}(\mathcal{H}) \leq 2$. ■

Proof (of (iii)-(vi) in Theorem 47) To see (iii), observe that the class $\mathcal{H} = \{x \mapsto \mathbb{1}\{x = a \text{ or } x \leq 0\} : a \in \mathbb{N}\}$ is trivially online learnable and uniformly generatable. To see (iv), observe that the class $\mathcal{H}_{\text{thresh}} \cup \mathcal{H}^e \cup \mathcal{H}^o$ is PAC learnable but neither uniformly generatable nor online learnable, where $\mathcal{H}_{\text{thresh}} := \{x \mapsto \mathbb{1}\{x \geq a\} : a \in \mathbb{N}\}$ and $\mathcal{H}^e$,$\mathcal{H}^o$ are defined in Equations 2 and 3. To see (v), observe that the class $\mathcal{H}_{\text{thresh}}$ is PAC learnable, uniformly generatable, but not online learnable. Finally, to see (vi), consider the class $\mathcal{H} = \{x \mapsto \mathbb{1}\{x \notin A\} : A \subset \mathbb{N}, |A| < \infty\}$. ■

J.1. Proofs of Lemma 15 and 16

Proof (of Lemma 15) Let $\mathcal{X} = \mathbb{Z}$ and consider $\mathcal{H} = \{x \mapsto \mathbb{1}\{x \in A \text{ or } x \leq 0\} : A \subseteq \mathbb{N}\}$ and $h(x) = \mathbb{1}\{x \in \mathbb{N}\}$. Its not hard to see that $\mathrm{C}(\mathcal{H}) = 0$. Moreover, since $\mathcal{H}$ is the same class used in the proof of Lemma 46, we know that $\mathcal{H} = \mathcal{H}_1 \cup \{h\}$ is not non-uniformly generatable. ■

Proof (of Lemma 16) Let $\mathcal{X} = \mathbb{Q}_+$ be all the positive rational numbers. Let $P = \{p_n\}_{n=1}^{\infty}$ be the set of prime numbers, indexed in increasing order. For each $i \in \mathbb{N}$, define

$$Q_i = \left\{\frac{p}{p_i}, p \in P\right\}, \quad \mathcal{H}_i = \{x \mapsto \mathbb{1}\{x \in Q_i \text{ or } x \in A\} : A \in 2^{\mathbb{Q}_+}\}.$$

Note that $\mathrm{C}(\mathcal{H}_i) = 0$ for all $i \in \mathbb{N}$. We now show that $\mathcal{H} = \bigcup_{i=1}^{\infty} \mathcal{H}_i$ is not generatable in the limit.

Suppose $\mathcal{H}$ is generatable in the limit and $\mathcal{G}$ is such a generator, we now prove a contradiction. Let $h_2 \in \mathcal{H}_2$ be such that $\mathrm{supp}(h_2) = Q_2 \cup \{\frac{3}{2}\}$. Let $\{x_{2,t}\}_{t=1}^{\infty}$ be an enumeration of $\mathrm{supp}(h_2)$ such that

$$\{x_{2,1}, x_{2,2}, x_{2,3}, x_{24}, \dots\} = \left\{\frac{p_2}{p_1}, \frac{p_1}{p_2}, \frac{p_2}{p_2}, \frac{p_3}{p_2}, \dots\right\}.$$

By definition, there exists a time $t_2 \geq 4$ such that $\mathcal{G}(x_{21}, \dots, x_{2t_2}) \in \mathrm{supp}(h_2) \setminus \{x_{21}, \dots, x_{2,t_2}\}$, which implies $\mathcal{G}(x_{21}, \dots, x_{2,t_2}) \in Q_2 \setminus \{x_{2,1}, \dots, x_{2,t_2}\}$. Note that $t_2 \geq 4$ implies that $1 \in \{x_{2,1}, \dots, x_{2,t_2}\}$. Let $h_3 \in \mathcal{H}_3$ be such that $\mathrm{supp}(h_3) = Q_3 \cup \{\frac{5}{2}\} \cup \{x_{2,1}, \dots, x_{2,t_2}\}$. Let $\{x_{3,t}\}_{t=1}^{\infty}$ be an enumeration of $\mathrm{supp}(h_3)$ such that

$$\{x_{3,1}, \dots, x_{3,t_2}\} = \{x_{2,1}, \dots, x_{2,t_2}\}$$

and

$$\{x_{3,t_2+1}, x_{3,t_2+2}, x_{3,t_2+3}, x_{3,t_2+4}, \dots\} = \left\{\frac{p_3}{p_1}, \frac{p_1}{p_3}, \frac{p_2}{p_3}, \frac{p_3}{p_3}, \dots\right\}.$$

We know there exists a time $t_3 \geq t_2+1$ such that $\mathcal{G}(x_{3,1}, \dots, x_{3,t_3}) \in \mathrm{supp}(h_3) \setminus \{x_{3,1}, \dots x_{3,t_3}\}$, which means $\mathcal{G}(x_{3,1}, \dots, x_{3,t_3}) \in Q_3 \setminus \{x_{3,1}, \dots x_{3,t_3}\}$.

Inductively, suppose $h_2, \dots, h_n$ and $t_2, \dots, t_n$ are all defined. Let $h_{n+1} \in \mathcal{H}_{n+1}$ be such that

$$\mathrm{supp}(h_{n+1}) = Q_{n+1} \cup \{\frac{p_{n+1}}{p_1}\} \cup \{x_{n,1}, \dots, x_{n,t_n}\}.$$

Let $\{x_{n+1,t}\}_{t=1}^{\infty}$ be an enumeration of $\mathrm{supp}(h_{n+1})$ such that

$$\{x_{n+1,1}, \dots, x_{n+1,t_n}\} = \{x_{n,1}, \dots, x_{n,t_n}\}$$

and

$$\{x_{n+1,t_n+1}, x_{n+1,t_n+2}, x_{n+1,t_n+3}, x_{n+1,t_n+4}, \dots\} = \left\{\frac{p_n}{p_1}, \frac{p_1}{p_n}, \frac{p_2}{p_n}, \frac{p_3}{p_n}, \dots\right\}.$$

By our construction, there exists a time $t_{n+1} \geq t_n + 1$ such that $\mathcal{G}(x_{n+1,1}, \dots, x_{n+1,t_{n+1}}) \in Q_{n+1} \setminus \{x_{n+1,1}, \dots, x_{n+1,t_{n+1}}\}$ Now, let h_1 be a hypothesis such that

$$\mathrm{supp}(h_1) = \bigcup_{i=2}^{\infty} \{x_{i,1}, \dots, x_{i,t_i}\}.$$

It is clear that $h_1 \in \mathcal{H}_1$. Moreover, let $\{x_i\}_{i=1}^{\infty}$ be an enumeration of h_1 such that for all $n \geq 2$,

$$\{x_1, x_2, \dots, x_{t_n}\} = \{x_{n,1}, x_{n,2}, \dots, x_{n,t_n}\}.$$

Then, by our construction, we know that for all $n \geq 2$, $\mathcal{G}(x_1, x_2, \dots, x_{t_n}) \in Q_n \setminus \mathrm{supp}(h_1)$, which means h_1 is not generatable in the limit. ■

Appendix K. Comparison to Kleinberg and Mullainathan (2024)'s Prompted Generation

Our setting of prompted generation generalizes the model of prompting studied in Section 7 of Kleinberg and Mullainathan (2024). Namely, Kleinberg and Mullainathan (2024) consider the following model. Let $\mathcal{X}$ be a suffix space, $\mathcal{Y}$ be the prompt space, and $\mathcal{Z} \subseteq \{y \circ x : x \in \mathcal{X}, y \in \mathcal{Y}\}$ be the space of completed prompts, where $\circ$ denotes the concatenation operator. Let $\mathcal{L} = \{L_1, L_2, \dots\}$ denote a language family defined over $\mathcal{Z}$. Before the game begins, the adversary picks language $K \in \mathcal{L}$, a sequence of its completed prompts $z_1, z_2, \dots \in K$, and a sequence of prompts $y_1, y_2, \dots$. On round $t \in \mathbb{N}$, the adversary reveals (z_t, y_t), and the goal of the generator is to output $\hat{x}_t \in \mathcal{X}$ such that $y_t \circ \hat{x}_t \in K \setminus \{z_1, \dots, z_t\}$. This model is equivalent to our setting after picking $\mathcal{H} = \{h_L : L \in \mathcal{L}\} \subseteq \mathcal{Y}^{\mathcal{X}}$ as the hypothesis class, where $h_L : \mathcal{X} \to \mathcal{Y}$ such that $h_L(x) = y$ if and only if $y \circ x \in L$.

That said, our definitions of prompted generatability differ from the notion of "prompted generation in the limit" in Section 7 of Kleinberg and Mullainathan (2024). We highlight the key differences below. Notice that in all our definitions of prompted generatability, the time point after which perfect generation must occur can be prompt specific. This means that the generator only needs to perfectly generate with respect to a prompt after it has seen a sufficient number of examples with this prompt. This, however, is not the case for the definition of prompted generation in the limit studied by Kleinberg and Mullainathan (2024). In their model, the generator must eventually perfectly complete prompts it may have never seen in the past. In this sense, our notions of prompted generatability are *weaker*.

Appendix L. Proofs for Prompted Generatability

L.1. Proof of Corollary 24

Proof (ii) follows from (i) and Theorem 23 and (iii) follows from (ii), so we only focus on proving (i). Let $\mathcal{H}$ be any finite class. Fix some $y \in \mathcal{Y}$ and consider the binary hypothesis class: $\mathcal{H}_y := \{x \mapsto \mathbb{1}\{h(x) = y\} : h \in \mathcal{H}\}$. Since $\mathcal{H}$ is finite, so is $\mathcal{H}_y$. Accordingly, by Theorem 36, we know that there exists $d_y \in \mathbb{N}$ that witnesses the fact that $\mathcal{H}_y$ is uniform generatable according to Definition 3. We claim that $\mathrm{PC}(\mathcal{H}) = \max_{y \in \mathcal{Y}} d_y$. For the sake of contradiction, suppose this is not the case. That is, $\mathrm{PC}(\mathcal{H}) \geq (\max_{y \in \mathcal{Y}} d_y) + 1$. Then, by definition, there exists a distinct sequence $x_1, \dots, x_{\mathrm{PC}(\mathcal{H})}$ and a prompt $y^\star \in \mathcal{Y}$ such that $|\langle (x_1, \dots, x_{\mathrm{PC}(\mathcal{H})}), y \rangle_{\mathcal{H}}| \neq \perp$ and $|\langle (x_1, \dots, x_{\mathrm{PC}(\mathcal{H})}), y^\star \rangle_{\mathcal{H}}| < \infty$. This implies that $d_{y^\star} \geq \mathrm{PC}(\mathcal{H})$ which contradicts the fact that $\mathrm{PC}(\mathcal{H}) = \max_{y \in \mathcal{Y}} d_y$. This completes the proof, as $\mathrm{PC}(\mathcal{H}) = \max_{y \in \mathcal{Y}} d_y < \infty$ and thus, by Theorem 22, $\mathcal{H}$ is prompted uniformly generatable. ■

L.2. Proof of Lemma 25

Proof Let $\mathcal{X} = \mathbb{Z}$ and $\mathcal{Y} = \mathbb{N}$. For every $n \in \mathbb{N}$, define the set

$$A_n := \left\{ \frac{n(n-1)}{2} + 1, \dots, \frac{n(n-1)}{2} + n \right\}.$$

Let $\{p_n\}_{n \in \mathbb{N}}$ be the set of all prime numbers. Consider the hypotheses $h_1 : \mathcal{X} \to \mathcal{Y}$ and $h_2 : \mathcal{X} \to \mathcal{Y}$ defined as

$$h_1(x) := \begin{cases} n, & \text{if } x \in A_n \\ n, & \text{if } x \in \{-p_n, -p_n^2, -p_n^3, \dots\} \\ 1, & \text{otherwise} \end{cases},$$

and

$$h_2(x) := \begin{cases} n, & \text{if } x \in A_n \\ n, & \text{if } x \in \{-p_{n+1}, -p_{n+1}^2, -p_{n+1}^3, \dots\} \\ 1, & \text{otherwise} \end{cases}.$$

Let $\mathcal{H} = \{h_1, h_2\}$. Observe that $\mathcal{H}$ satisfies the PUUS property. Our proof will be in two steps. First, we will now show that $\mathcal{H}$ is not prompted uniformly generatable. Then, we will use Theorem 23 to show that $\mathcal{H}$ is not prompted non-uniformly generatable.

By Theorem 22, to show that $\mathcal{H}$ is not prompted uniformly generatable it suffices to show that $\text{PC}(\mathcal{H}) = \infty$. In particular, it suffices to show that for every $d \in \mathbb{N} \setminus \{1\}$, there exists distinct $x_1, \dots, x_d$ and a prompt $y \in \mathcal{Y}$ such that $|\langle (x_1, \dots x_d), y\rangle| \neq \perp$ and $|\langle (x_1, \dots x_d), y\rangle| < \infty$. To that end, fix some $d \in \mathbb{N} \setminus \{1\}$. Consider the sequence $x_1, \dots, x_d$ obtained by sorting A_d in increasing order and consider the prompt $y = d$. Then, we have that $\mathcal{H}(x_{1:d}, y) = \mathcal{H}$ and

$$\langle x_{1:d}, y\rangle_{\mathcal{H}} = \bigcap_{h \in \mathcal{H}} \text{supp}(h, d) = A_d,$$

so that $|\langle x_{1:d}, y\rangle_{\mathcal{H}}| < \infty$. The proof that $\mathcal{H}$ is not prompted uniformly generatable is complete after noting that $d \in \mathbb{N} \setminus \{1\}$ is picked arbitrarily.

Now, we complete the overall proof by showing that $\mathcal{H}$ is not prompted non-uniformly generatable. By Theorem 23, $\mathcal{H}$ is not prompted non-uniformly generatable if for every non-decreasing sequence of classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots$ satisfying $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$, there exists a $i \in \mathbb{N}$ such that $\text{PC}(\mathcal{H}_i) = \infty$. Trivially, for every non-decreasing sequence of classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots$ satisfying $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$, there must be an index $i \in \mathbb{N}$ such that $\mathcal{H} = \mathcal{H}_i$. Since $\text{PC}(\mathcal{H}) = \infty$, our proof is complete. ■

L.3. Proof of Theorem 22

Proof (of sufficiency) Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be such that it satisfies the PUUS property and $\text{PC}(\mathcal{H}) < \infty$. Consider the following prompted generator $\mathcal{G}$. For any finite sequence of tuples $(x_1, p_1, y_1), \dots, (x_t, p_t, y_t) \in (\mathcal{X}, \mathcal{Y}, \mathcal{Y})^t$, $\mathcal{G}$ extracts $B_t := \{x_i : p_i = y_t\}$, the subset of examples where p_i is y_t. Then, $\mathcal{G}$ checks whether $|B_t| \geq \text{PC}(\mathcal{H}) + 1$. If so, $\mathcal{G}$ computes $\langle B_t, y_t\rangle_{\mathcal{H}}$ and plays $\hat{x}_t \in \langle B_t, y_t\rangle_{\mathcal{H}} \setminus \{x_1, \dots, x_t\}$. Otherwise, $\mathcal{G}$, plays an arbitrary $\hat{x}_t \in \mathcal{X}$. We claim that $\mathcal{G}$ is a prompted uniform generator for $\mathcal{H}$. To see this, let $h \in \mathcal{H}$ and $(x_1, h(x_1), y_1), (x_2, h(x_2), y_2), \dots$ be the hypothesis and sequence of tuples chosen by the adversary. Fix an arbitrary reference label $y^\star \in \mathcal{Y}$. It suffices to show that if there exists a $t^\star \in \mathbb{N}$ such that

$$|\{x_1, \dots, x_{t^\star}\} \cap \text{supp}(h, y^\star)| = \text{PC}(\mathcal{H}) + 1,$$

then

$$\mathcal{G}((x_1, h(x_1), y_1), \dots, (x_s, h(x_s)), y_s) \in \text{supp}(h, y_s) \setminus \{x_1, \dots, x_s\}$$

for all $s \geq t^\star$ where $y_s = y^\star$. To that end, suppose there exists a $t^\star \in \mathbb{N}$ such that

$$|\{x_1, \dots, x_{t^\star}\} \cap \text{supp}(h, y^\star)| = \text{PC}(\mathcal{H}) + 1.$$

Fix an arbitrary $s \geq t^\star$ such that $y_s = y^\star$. By construction, we have that $|B_s| \geq \text{PC}(\mathcal{H}) + 1$. Accordingly, $\mathcal{G}$ computes $\langle B_s, y_s \rangle_{\mathcal{H}} = \langle B_s, y^\star \rangle_{\mathcal{H}}$. By definition of the Prompted Closure dimension, it must be the case that $|\langle B_s, y_s \rangle_{\mathcal{H}}| = \infty$. Accordingly, $\langle B_s, y_s \rangle_{\mathcal{H}} \setminus \{x_1, \dots, x_s\} \neq \emptyset$, and we have that $\hat{x}_s \in \text{supp}(h, y_s) \setminus \{x_1, \dots, x_s\}$, completing the proof. ■

Proof (of necessity) Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be such that it satisfies the PUUS property and $\text{PC}(\mathcal{H}) = \infty$. Let $\mathcal{G}$ be any prompted generator. It suffices to show that for arbitrarily large $d \in \mathbb{N}$, there exists a sequence of distinct examples $x_1, \dots, x_d$, a prompt $y \in \mathcal{Y}$, and a hypothesis $h \in \mathcal{H}$ such that $h(x_i) = y$ for all $i \in [d]$, and

$$\mathcal{G}((x_1, h(x_1), y), \dots, (x_d, h(x_d)), y) \notin \text{supp}(h, y) \setminus \{x_1, \dots, x_d\}.$$

By definition of the Prompted Closure dimension, for every $n \in \mathbb{N}$ there exists distinct $x_1, \dots, x_n \in \mathcal{X}$ and a label $y^\star \in \mathcal{Y}$ such that $|\langle (x_1, \dots, x_n), y^\star \rangle_{\mathcal{H}}| \neq \perp$ and $|\langle (x_1, \dots, x_n), y^\star \rangle_{\mathcal{H}}| < \infty$. Thus, for some $d \geq n$, there exists a distinct $x_1, \dots, x_d \in \mathcal{X}$ such that $|\langle (x_1, \dots, x_d), y^\star \rangle_{\mathcal{H}}| \neq \perp$ and $|\langle (x_1, \dots, x_d), y^\star \rangle_{\mathcal{H}} \setminus \{x_1, \dots, x_d\}| = 0$. Accordingly, for every $x \in \mathcal{X} \setminus \{x_1, \dots, x_d\}$, there exists a $h \in \mathcal{H}((x_1, \dots, x_d), y^\star)$ such that $x \notin \text{supp}(h, y^\star) \setminus \{x_1, \dots, x_d\}$. Let $\hat{x}_d = \mathcal{G}((x_1, h(x_1), y^\star), \dots, (x_d, h(x_d), y^\star))$, and suppose without loss of generality that $\hat{x}_d \notin \{x_1, \dots, x_d\}$. Then, by the previous observation, there exists a $h^\star \in \mathcal{H}((x_1, \dots, x_d), y^\star)$ for which $\hat{x}_d \notin \text{supp}(h^\star, y^\star)$. Thus, we have shown that there exists a sequence of distinct examples $x_1, \dots, x_d$, a label $y \in \mathcal{Y}$, and a hypothesis $h \in \mathcal{H}$ such that $h(x_i) = y$ for all $i \in [d]$, and

$$\mathcal{G}((x_1, h(x_1), y), \dots, (x_d, h(x_d), y)) \notin \text{supp}(h, y) \setminus \{x_1, \dots, x_d\}.$$

The proof is complete after noting that n was arbitrary, and thus this holds for all $n \in \mathbb{N}$. ■

L.4. Proof of Theorem 23

The proof of the necessity direction follows identically to that of Theorem 11, so we omit that proof and only prove the sufficiency direction. Although one can prove the sufficiency direction in Theorem 23 through a reduction to prompted uniform generation (like we did for Theorem 11), we provide a more direct proof using the prompted closure dimension to avoid repetition.

Proof (sketch of sufficiency) Suppose $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is a hypothesis class satisfying the PUUS property such that there exists a non-decreasing sequence of classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots$ with $\mathcal{H} = \bigcup_{i \in \mathbb{N}} \mathcal{H}_i$ and $\text{PC}(\mathcal{H}_n) < \infty$ for every $n \in \mathbb{N}$. First note that $\text{PC}(\mathcal{H}_n)$ is monotonic increasing in n. We consider two cases: $\lim_{n \to \infty} \text{PC}(\mathcal{H}_n) = \infty$ and $\lim_{n \to \infty} \text{PC}(\mathcal{H}_n) < \infty$.

In the first case, consider the following generator $\mathcal{G}$. Fix $t \in \mathbb{N}$ and consider any finite sequence of tuples $(x_1, p_1, y_1), \dots, (x_t, p_t, y_t) \in (\mathcal{X}, \mathcal{Y}, \mathcal{Y})^t$. $\mathcal{G}$ extracts $B_t := \{x_i : p_i = y_t\}$. Let $d_t := |B_t|$ be the number of unique examples whose label is y_t. $\mathcal{G}$ first computes

$$n_t = \max\{n \in \mathbb{N} : \mathrm{PC}(\mathcal{H}_n) < d_t\} \cup \{0\}.$$

If $n_t = 0$, meaning $\mathrm{PC}(\mathcal{H}_1) \geq d_t$, $\mathcal{G}$ plays any $\hat{x}_t \in \mathcal{X}$. If $n_t > 0$ but $|\mathcal{H}_{n_t}(B_t, y_t)| = 0$, $\mathcal{G}$ also plays any $\hat{x}_t \in \mathcal{X}$. If $n_t > 0$ and $|\mathcal{H}_{n_t}(B_t, y_t)| \geq 1$, $\mathcal{G}$ plays any

$$\hat{x}_t \in \langle B_t, y_t \rangle_{\mathcal{H}_{n_t}} \setminus \{x_1, \ldots, x_t\}.$$

We now prove that such a $\mathcal{G}$ is a non-uniform generator for $\mathcal{H}$. To that end, let $h^\star$ be the hypothesis chosen by the adversary and suppose that $h^\star$ belongs to $\mathcal{H}_{n^\star}$. Let $d^\star := \mathrm{PC}(\mathcal{H}_{n^\star})$. We show that for a label sequence $(x_1, h^\star(x_1), y_1), \ldots, (x_t, h^\star(x_t), y_t)$, such that $d_t := |\{x_i : h^\star(x_i) = y_t\}| \geq d^\star + 1$, we have

$$\mathcal{G}((x_1, h^\star(x_1), y_1), \ldots, (x_t, h^\star(x_t), y_t)) \in \mathrm{supp}(h^\star, y_t) \setminus \{x_1, \ldots, x_t\}.$$

By definition, $\mathcal{G}$ first computes

$$n_t = \max\{n \in \mathbb{N} : \mathrm{PC}(\mathcal{H}_n) < d_t\} \cup \{0\}.$$

Note that $n_t \geq n^\star$ since $\mathrm{PC}(\mathcal{H}_{n^\star}) = d^\star < d^\star + 1 \leq d_t$. Thus, $|\mathcal{H}_{n_t}(B_t, y_t)| \geq 1$ since $h^\star \in \mathcal{H}_{n_t}$, where $B_t = \{x_i : h^\star(x_i) = y_t\}$. Accordingly, by construction of $\mathcal{G}$, we have that it computes

$$V_t := \langle B_t, y_t \rangle_{\mathcal{H}_{n_t}},$$

and plays any $\hat{x}_t \in V_t \setminus \{x_1, \ldots, x_t\}$. The proof is complete by noting that $h^\star \in \mathcal{H}_{n_t}$ and $d_t \geq \mathrm{PC}(\mathcal{H}_{n_t}) + 1$ which gives that $|V_t| = \infty$ and $V_t \subseteq \mathrm{supp}(h^\star)$.

In the second case, suppose $\lim_{n\to\infty} \mathrm{PC}(\mathcal{H}_n) := c < \infty$. Consider the following generator $\mathcal{G}$. Fix $t \in \mathbb{N}$ and consider any finite sequence of tuples $(x_1, p_1, y_1), \ldots, (x_t, p_t, y_t) \in (\mathcal{X}, \mathcal{Y}, \mathcal{Y})^t$, $\mathcal{G}$ extracts $B_t = \{x_i : p_i = y_t\}$. Let $d_t := |B_t|$ be the number of unique examples whose label is y_t. If $d_t < c$ or $|\mathcal{H}_{d_t}(B_t, y_t)| = 0$, $\mathcal{G}$ plays any $\hat{x}_t \in \mathcal{X}$. Otherwise, if $d_t \geq c$ and $|\mathcal{H}_{d_t}(B_t, y_t)| > 0$, $\mathcal{G}$ plays any

$$\hat{x}_t \in \langle B_t, y_t \rangle_{\mathcal{H}_{d_t}} \setminus \{x_1, \ldots, x_t\}.$$

Let $h^\star$ be the hypothesis chosen by the adversary and suppose $h^\star$ belongs to $\mathcal{H}_{n^\star}$. We show that for every labeled sequence $(x_1, h^\star(x_1), y_1), \ldots, (x_t, h^\star(x_t), y_t)$ such that $d_t := |\{x_i : h^\star(x_i) = y_t\}| > \max(c, n^\star)$, we have

$$\mathcal{G}((x_1, h^\star(x_1), y_1), \ldots, (x_t, h^\star(x_t), y_t)) \in \mathrm{supp}(h^\star, y_t) \setminus \{x_1, \ldots, x_t\}.$$

Because $d_t \geq n^\star$, we have that $h^\star \in \mathcal{H}_{d_t}$. Therefore, by construction, $\mathcal{G}$ computes

$$V_t := \langle \{x_i : h^\star(x_i) = y_t\}, y_t \rangle_{\mathcal{H}_{[d_t]}},$$

and plays any $\hat{x}_t \in V_t \setminus \{x_1, \ldots, x_t\}$. The proof is complete after noting that $|V_t| = \infty$ and $V_t \subseteq \mathrm{supp}(h^\star, y_t)$ using the fact that $d_t > \max(c, n^\star)$. ■

Appendix M. Open Questions

We highlight several important directions for future work.

Characterizing Generatability in the Limit. Kleinberg and Mullainathan (2024) proved that every countable hypothesis class is generatable in the limit. We gave an alternate sufficiency condition which showed the existence of many uncountably infinite classes that are generatable in the limit. However, it is unclear (and unlikely) that our sufficiency condition, in conjunction with countableness, provides a characterization of generatability in the limit. This motivates our first open question.

Question 48 *What characterizes generatability in the limit?*

Ideally, a characterization of generatability in the limit can be written neatly in set-theoretic language like Angluin's characterization of identifiability in the limit.

Generatability in the Limit under Finite Unions. In Section 3.4, we showed that uniform and non-uniform generatability are not closed under finite unions. However, we were able to only show that generatability in the limit is not closed under countable unions. This motivates our second open question.

Question 49 *Is generatability in the limit closed under finite unions?*

Resolving Question 49 is important for two reasons: (1) it may lead to a complete characterization of generatability in the limit and (2) it may provide insight on how to optimally combine generators. Recall that Theorem 42 shows that the finite union of uniformly generatable classes are generatable in the limit. Thus, as a first step towards resolving Question 49, it might be helpful to resolve the following open question.

Question 50 *Is the finite union of non-uniformly generatable classes generatable in the limit?*

Characterizing Prompted Generatability in the Limit. In Section 4, we provided complete characterizations of prompted uniform and non-uniform generation. When $|\mathcal{Y}| < \infty$, we showed that all countable classes are prompted generatable in the limit. However, we left open the complete characterization of prompted generatability in the limit, which motivates our first question.

Question 51 *When $|\mathcal{Y}| < \infty$, what characterizes prompted generatability in the limit?*

When $|\mathcal{Y}| = \infty$, we show that there are finite classes which are not even prompted non-uniformly generatable. This begs the question of whether all countable classes continue to be prompted generatable in the limit when $|\mathcal{Y}| = \infty$.

Question 52 *When $|\mathcal{Y}| = \infty$, are all countable classes prompted generatable in the limit?*

Unlike the case for prompted uniform and non-uniform generatability, we conjecture that all countable classes are still prompted generatable in the limit when $|\mathcal{Y}| = \infty$. Our claim is due to the positive result by Kleinberg and Mullainathan (2024), who show that in their model of prompting, which can be stronger than ours (see Section K), all countable classes are still prompted generatable in the limit.