A Theory of Time-Sensitive Language Generation: Sparse Hallucination Beats Mode Collapse

Atul Ganju
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
ganju@usc.edu

Travis McVoy
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
mcvoy@usc.edu

Shaddin Dughmi
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
shaddin@usc.edu

Shang-Hua Teng
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
shanghua@usc.edu

Abstract

We study language generation in the limit under a global preference ordering on strings, as introduced by Kleinberg and Wei. As in [17, 18], we aim for *breadth*, but impose an additional requirement of timeliness: higher-ranked strings should be generated earlier. A string is then only credited if it is generated before a deadline, where its deadline is defined by a function that maps a string's rank in the target language to the time by which it must be produced. This is in keeping with a central consideration in machine learning, where inductive bias favors "simpler" or "more plausible" outputs, all else being equal. We show that timely generation is impossible in a strong sense for eventually consistent generators—the protagonists of most prior related work. Under what is perhaps the mildest natural relaxation of consistency, a hallucination rate that vanishes over time, we show that we can circumvent our impossibility result. In particular, we can achieve optimal density with respect to any superlinear deadline function. We also show this is tight by ruling out timely generation with linear deadlines and vanishing hallucination rate.

1 Introduction

Generative language models are increasingly a key building block of modern AI systems, enabling a multitude of downstream applications such as general-purpose dialog, creative writing, mathematical reasoning, and code synthesis [23, 7, 19]. Alongside this undeniable practical impact, a growing body of theoretical work seeks to understand the principles and trade-offs governing language generation.

An increasingly influential approach, often referred to as *language generation in the limit*, was spurred by the seminal work of Kleinberg and Mullainathan [16], building on classical work concerned with language identification by Gold [10] and Angluin [1]. At a high level, language generation is modeled as a game played over time between the generator and an adversary: At the outset the adversary chooses a *target language* from some countable collection of permissible languages, then the adversary and generator take turns generating strings. The adversary always generates a string from the target language, and the generator's goal is to eventually do the same. By focusing on the "bare-bones" essentials of language generation and pursuing guarantees only in the limit over time, this approach enables theoretical analysis of what is possible *in principle*, including identifying and quantifying the intrinsic *tension* between various design desiderata.

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

Much of the recent literature has focused on the tension between two desiderata: avoiding *hallucination* by the generator, often formalized through notions such as *consistency*, and achieving *breadth* in the generator's outputs over the target language. We introduce a third concern in this paper which we call *timeliness*, and study its interplay with hallucination and breadth. Timeliness postulates that "more plausible" or "simpler" outputs should be generated preferentially, which we take to mean earlier in the language generation game. Plausibility can represent any of the myriad forms of inductive bias employed in machine learning: preference for short programs in coding tasks, simpler or human-checkable proofs in mathematical reasoning tasks (judged, perhaps, by a trained reward model), or distributional plausibility in dialog or creative writing tasks [22, 26, 9]. We measure timeliness with respect to a *deadline* function, mapping a string's position in the plausibility order to a time in the game by which it must be generated, lest the generator forego the associated credit.

In addition to being compelling in its own right, timeliness shares technical and conceptual connections with recent notions of breadth introduced by Kleinberg and Wei [17]. As background, there are several natural ways of modeling breadth, most of which lead to strong negative results—See [13, 14] for examples. This paints the theory into a corner by suggesting the inevitability of *mode collapse*—the eventual deterioration of the generator to a tiny sliver of the target language. While mode collapse is a very real concern in practice [2, 3], such a stark prediction is perhaps increasingly out of step with empirical reality [27]. A key conceptual insight came with the work of [17], who measure breadth with respect to a preference ordering on strings. In particular, a generator achieves breadth in the target language if, when strings in the language are sorted in order of preference, its outputs are *dense* in every sufficiently long prefix of the language. This approach to modeling breadth with respect to an inductive-bias on strings enabled remarkable progress in [17, 18], culminating in an algorithm achieving both consistency and optimal breadth in the limit. Timeliness can best be understood as a strengthening of the breadth requirements of [17, 18]: in each prefix of the target language, a generated string only "counts" towards density if it is generated before its deadline. This reflects a natural desideratum for LLMs and other generative agents: concise, plausible, or otherwise high-value outputs should emerge from limited evidence in a finite or limited data regime, rather than after an unbounded stream of demonstrations.

Our technical work starts with the realization that the algorithms of [17, 18], and in fact any eventually consistent generator, can not achieve timely generation. This impossibility holds for any deadline function, and any nontrivial guarantee on the prefix-density of timely outputs. In other words, consistent generators inevitably suffer *mode collapse under a deadline*. This motivates relaxing consistency: by permitting hallucinations at a rate tending to zero over time, we recover timely generation in the limit for any superlinear deadline function, and provide essentially-matching impossibility results. Next, we present the relevant technical preliminaries before formally stating our results. We defer more thorough discussion of related work to Appendix A.

1.1 Technical preliminaries

Basic notation. We often make use of countably infinite sets which are ordered (i.e. enumerated) by a surjective map from the natural numbers. We denote such an *ordered set* as $S = (s_1, s_2, \ldots)$, where s_i denotes the ith element of the enumeration. We also use $S_i = (s_1, \ldots, s_i) = s_{1:i}$ to denote the first i elements of the enumeration of S, and often refer to it as the prefix of S of size i. Alternatively, we sometimes do not fix the ordering of a countably infinite set K, and instead consider the space of all enumerations which we denote by $\mathcal{E}(K)$. Note that our notion of enumeration here can allow repetition, though in some cases we restrict attention to bijective enumerations.

The language generation game. As in [16, 17], we represent language generation as a game between an adversary and a generator. Both players have access to a countably infinite collection of languages $\mathcal{L} = (L^1, L^2, \ldots)$, defined over a countably infinite *universe* $U = (u_1, u_2, \ldots)$ of "strings." Note that both $\mathcal{L}$ and U are equipped with a bijective enumeration order, which is fixed and known. In the case of U, we think of the order $(u_1, u_2, \ldots)$ as encoding a preference (or inductive bias) on strings. This order on U induces a preference order on the strings in each language $L \in \mathcal{L}$ by restriction—i.e, strings in L are also enumerated by a total order which is consistent with the order on U.

The adversary first selects a target language $K \in \mathcal{L}$ and an enumeration $E = (e_1, e_2, \ldots)$, where $E \in \mathcal{E}(K)$. The generator $\mathcal{A} = \{\mathcal{A}_t : U^t \to U\}_{t=1}^{\infty}$ observes E sequentially. At time t, after having observed $e_{1:t}$, the generator outputs $o_t = \mathcal{A}_t(e_{1:t}) \in U \setminus \{e_1, \ldots, e_t\}$. The generator may repeat its own previous outputs, provided they have not been observed from the adversary. Let

$\mathcal{A}(E) = (o_1, o_2, \ldots)$ denote the output sequence of the generator. We allow $\mathcal{A}$ to be randomized, in which case each $\mathcal{A}_t(e_{1:t})$ is a random variable, where expectations are over the internal randomness of the algorithm.

Hallucination and consistency. We say the generator *hallucinates* at time t if $o_t \notin K$. We say it is *eventually consistent* if it is guaranteed to eventually stops hallucinating; i.e., there exists some finite time T (perhaps depending on $\mathcal{L}$, K, and E) such that for all $t \geq T$ the output $o_t \in K$. The celebrated result of [16] exhibits an eventually consistent generator for every language collection $\mathcal{L}$.

Our impossibility results will necessitate relaxing consistency by allowing hallucinations that vanish over time. We define the *hallucination rate* of $\mathcal{A}$ at step t, as a function of $(\mathcal{L}, K, E)$, as follows

$$H_{\mathcal{A}}^{\mathcal{L},K,E}(t) := \frac{1}{t}\mathbb{E}\left[\sum_{s=1}^{t} \mathbf{1}\{o_s \notin K\}\right],$$

where the expectation is over the internal randomness of $\mathcal{A}$. Note that repeated outputs outside K count as separate hallucinating rounds and that hallucination rate bounds are imposed separately for each fixed instance, and their eventual time thresholds may depend on $(\mathcal{L}, K, E)$.

We say that $\mathcal{A}$ has vanishing hallucination rate if, for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $E \in \mathcal{E}(K)$,

$$\lim_{t\to\infty} H_{\mathcal{A}}^{\mathcal{L},K,E}(t) = 0.$$

All asymptotic hallucination-rate guarantees are pointwise in $(\mathcal{L}, K, E)$ meaning the time after which a bound holds may depend on the instance.

Breadth and timeliness. For a deadline function $D : \mathbb{N} \to \mathbb{N}$ and two sequences $S = (s_1, s_2, \ldots)$ and $R = (r_1, r_2, \ldots)$, we define the *timely density of S in R under deadline D* at index i as follows

$$\mu_i(S, R, D) = \frac{\sum_{j=1}^{i} \mathbf{1}\{r_j \in S_{D(j)}\}}{i}.$$

The generator's objective is to maximize the (lower) density of timely outputs, in the limit over prefixes of the target language. Formally, for any deterministic generation algorithm $\mathcal{A}$ and deadline function D, we define its timely lower density, taken in the limit for a worst-case instance, as follows.

$$\mu_{low}(\mathcal{A}, D) = \inf\left\{\inf_{K\in\mathcal{L}}\inf_{E\in\mathcal{E}(K)} \liminf_{i\to\infty} \mu_i(\mathcal{A}(E), K, D) : \mathcal{L} \text{ is a countable language collection}\right\}.$$

For randomized generators, density guarantees are understood to be almost surely for each fixed instance. In particular, a guarantee $\mu_{low}(\mathcal{A}, D) \geq \rho$ means that, for every $\mathcal{L}$, $K \in \mathcal{L}$, and $E \in \mathcal{E}(K)$,

$$\Pr\Big(\liminf_{i\to\infty} \mu_i(\mathcal{A}(E), K, D) \geq \rho\Big) = 1.$$

We note that the lower density employed in [17, 18] corresponds to employing the trivial deadline function $D(i) = \infty$. There, the remarkable result of [18] exhibits a generator which guarantees a lower density of $1/2$, which can easily be seen to be the best possible.[1] In what follows, we will frequently use the generalized inverse of the deadline function, defined by $D^{-1}(t) := \min\{n \in \mathbb{N} : D(n) \geq t\}$. Intuitively, $D^{-1}(t)$ is the smallest index whose deadline has not yet passed by time t.

1.2 Overview of results

Our first result (Theorem 1) exhibits a language family where every eventually-consistent generator—such as those in [17, 18] and much of the prior work—must suffer a timely lower density of 0 in the limit. This motivates a minimal relaxation of consistency, by allowing a vanishing hallucination rate over time. Even with this relaxation, we show another impossibility result (Theorem 2): Nontrivial timeliness under linear deadlines is also impossible, again suffering a lower density of 0 in the limit.

These two impossibilities set the stage for our main positive result (Theorem 3): For any super-linear deadline function, we exhibit a randomized generation algorithm guaranteeing a timely density of

[1] This is due to the turn-taking nature of the language generation game, whereby an aggressive adversary can "beat" the generator to half the strings in the language. This holds even for a trivial collection consisting of a single language.

$1/2$ with vanishing hallucination rate. Our guarantees are optimal in several respects: the density guarantee of $1/2$ is the best possible as in [18], superlinear deadlines and some hallucination are necessary by our impossibility results, and furthermore we show quantitatively that our hallucination rate is essentially tight (Lemma 1). Our proof is via a density-preserving black-box reduction to non-timely generation with breadth, into which we plug in the optimal generator of [18].

Finally, in the appendix we extend these results to general data models (e.g. partial enumeration and contamination models) and discuss a more permissive upper-density notion.

2 Fundamental limits for consistent, dense, timely generation

In this section, we establish that *no* eventually consistent generator can achieve nontrivial lower density in a timely fashion under *any* deadline function.

Theorem 1 (Impossibility for Eventually Consistent Generators)**.** *There exists a language family $\mathcal{L}$ such that, for every deadline function $D : \mathbb{N} \to \mathbb{N}$ and every (potentially randomized) eventually consistent generator $\mathcal{A}$, there exist a target language $K \in \mathcal{L}$ and a total enumeration $E \in \mathcal{E}(K)$ such that*

$$\Pr\Big(\liminf_{i\to\infty} \mu_i(\mathcal{A}(E), K, D) = 0\Big) = 1.$$

It follows that $\mu_{low}(\mathcal{A}, D) = 0$ almost surely for every deadline function D and every eventually-consistent generator $\mathcal{A}$.

This rules out timeliness for the eventually-consistent generators of [17, 18]. More broadly, it shows that eventual consistency itself precludes nontrivial timely density, thus necessitating its relaxation. Our next theorem shows that even after relaxing consistency to allow for a vanishing rate of hallucination, no generator can achieve nontrivial lower density under linear deadlines.

Theorem 2 (Impossibility for Linear-Time Generators)**.** *There exists a language family $\mathcal{L}$ such that, for every deadline function $D(i) = O(i)$ and every generator $\mathcal{A}$ with vanishing hallucination rate, there exist a target language $K \in \mathcal{L}$ and a total enumeration $E \in \mathcal{E}(K)$ such that*

$$\Pr\Big(\liminf_{i\to\infty} \mu_i(\mathcal{A}(E), K, D) = 0\Big) = 1.$$

It follows that $\mu_{low}(\mathcal{A}, D) = 0$ almost surely whenever $D(i) = O(i)$ and the generator $\mathcal{A}$ has a vanishing hallucination rate.

This motivates designing timely generators with respect to super-linear deadlines, as we do Section 3. Both Theorems 1 and 2 are consequences of the following fundamental lemma, which isolates the necessity of "gradually" vanishing hallucination in obtaining a nontrivial lower density.

Lemma 1 (Hallucination Barrier)**.** *There exists a language family $\mathcal{L}$ such that, for any monotone non-decreasing deadline function $D : \mathbb{N} \to \mathbb{N}$ and any randomized generator $\mathcal{A}$ satisfying*

$$H_{\mathcal{A}}^{\mathcal{L},K,E}(t) = o\left(\frac{D^{-1}(t)}{t}\right)$$

for every $K \in \mathcal{L}$ and every $E \in \mathcal{E}(K)$, there exist a target language $K \in \mathcal{L}$ and a total enumeration $E \in \mathcal{E}(K)$ such that

$$\Pr\Big(\liminf_{i\to\infty} \mu_i(\mathcal{A}(E), K, D) = 0\Big) = 1.$$

This result shows that any algorithm $\mathcal{A}$ whose hallucination rate is $o(D^{-1}(t)/t)$ by time t must suffer $\mu_{low}(\mathcal{A}, D) = 0$ almost surely with respect to any deadline function D. In fact, it establishes something stronger: there exists a single language collection that serves as a universal witness against all such algorithms.

Proof of Theorems 1 and 2. The first theorem follows by observing that eventual consistency implies the condition of Lemma 1 for any deadline function. The second theorem follows by applying the lemma to linear deadline functions. □

2.1 Discussion of Lemma 1: hallucination is necessary for dense generation

We identify a structural property of language families such that any collection containing a subfamily with this property witnesses this impossibility. This structure is a nested sequence of languages that increasingly approximate a limiting language L^∞, while ensuring that each intermediate language becomes arbitrarily sparse within L^∞. This structure forces any generator with vanishing hallucination rate on each of the intermediate languages to achieve zero lower density on the limiting language.

Definition 1 (Measure-Zero Chain)**.** *A collection $\mathcal{L}_{chain} = \{L^\infty\} \cup \bigcup_{j\geq 1}\{L^j\}$ forms a measure-zero chain if $L^1 \subseteq L^2 \subseteq \cdots \subseteq L^\infty$ is a nested sequence of languages for which:*

$$\bigcup_{j\geq 1} L^j = L^\infty \qquad \textit{and} \qquad \liminf_{i\to\infty} \mu_i(L^j, L^\infty, \infty) = 0 \qquad \forall j \in \mathbb{N}.$$

This structure allows the adversary to present the generator with an enumeration of L^∞ in such a way that, for long intervals of time, the sequence is indistinguishable from an enumeration of some intermediate language L^j, before eventually revealing additional elements of the true target. Since the generator must keep hallucinations infrequent with respect to L^j, it is forced to behave conservatively and output at most $o(D^{-1}(t))$ elements outside of L^j by time t. However, because each L^j captures only a vanishing fraction of L^∞ along suitable prefixes, this conservative behavior prevents the generator from achieving nontrivial timely density in L^∞.

Such families arise naturally in the study of language generation. For instance, the language family in Example 7 of [17] satisfies this property. For completeness, we provide a proof of this in Appendix B along with several other examples of language families that have this structure.

Below we give a proof sketch of Lemma 1. A full proof can be found in Appendix C.

Proof Sketch. The adversary constructs a total enumeration X^∞ of L^∞ in stages. At stage j, it fixes a total enumeration of L^j extending the current prefix, with any unseen elements of $L^j \cap L^\infty_j$ revealed next. It follows this enumeration until a suitably large deterministic time $t_j := D(i_j)$, after these revelations, before proceeding to stage $j+1$. The indices i_j are chosen increasingly so that (i) the expected number of hallucinating rounds on L^j by time t_j is sufficiently small relative to i_j, and (ii) the density of L^j in the prefix $L^\infty_{i_j}$ is sufficiently small. The first condition holds at every sufficiently large index by the assumed hallucination bound and $D^{-1}(D(i)) \leq i$, while the second holds at arbitrarily large indices by the measure-zero-chain property. The prescribed revelations ensure that X^∞ is a total enumeration of L^∞.

Since the adversary's constructed sequence agrees with the fixed enumeration of L^j up to time t_j, the generator produces identical outputs in both possible settings up to this time for each fixed random seed. Accordingly, we decompose the outputs contributing to $L^\infty_{i_j}$ into two parts: elements of L^j, which dominate, and distinct outputs outside L^j. The first contribution is small by (ii), and the second is bounded by the number of hallucinating rounds when L^j is the target. Thus, the second contribution can be made small with high probability via Markov's inequality applied using (i), yielding a failure probability that decays exponentially in j. A Borel–Cantelli argument then implies that, almost surely, the hallucination contribution is small for all sufficiently large j. It follows that along the subsequence $\{i_j\}_{j\in\mathbb{N}}$, the fraction of elements of $L^\infty_{i_j}$ generated by time $D(i_j)$ tends to zero. Since $D(i_j)$ upper bounds the deadline of every element in $L^\infty_{i_j}$, this yields $\liminf_{i\to\infty} \mu_i(\mathcal{A}(X^\infty), L^\infty, D) = 0$ almost surely. □

3 Timely dense language generation with sparse hallucination

In this section, we present a (randomized) timely generative algorithm for dense language generation with sparse hallucination for any super-linear deadline function. We prove that our algorithm achieves optimal density guarantees and has an essentially optimal hallucination rate, with the boundary regime specified below remaining open. To simplify upcoming statements, we define the notion of a feasible profile motivated by Lemma 1.

Definition 2 (Feasible Profile)**.** *We say (D, H) is a feasible profile if $D : \mathbb{N} \to \mathbb{N}$ is a monotone non-decreasing, super-linear function, and $H : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is a monotone non-increasing vanishing function satisfying $H(t) = \omega(D^{-1}(t)/t)$.*

We now turn to the positive result, which shows that the impossibility phenomena identified in the previous section can be overcome when the deadline function grows sufficiently quickly.

Theorem 3 (Dense Generation in Nearly-Linear Time)**.** *For any feasible profile (D, H), there exists a randomized generation algorithm $\mathcal{A}$ that achieves density $\mu_{low}(\mathcal{A}, D) \geq 1/2$ almost surely for each fixed instance and satisfies the following: for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $E \in \mathcal{E}(K)$, there exists a finite time T such that*

$$H_{\mathcal{A}}^{\mathcal{L},K,E}(t) \leq H(t) \qquad \forall t \geq T.$$

This theorem characterizes a sharp tradeoff between the growth of the deadline function and the minimial hallucination rate sufficient to achieve optimal density. In particular, the permitted hallucination rate at time t can be set to lie in the range

$$D^{-1}(t)/t \;\ll\; H(t) \;\ll\; 1.$$

The upper bound $H(t) = o(1)$ captures our requirement of a vanishing hallucination rate. The sufficient condition $H(t) = \omega(D^{-1}(t)/t)$ complements Lemma 1, which rules out nontrivial lower-density guarantees when the hallucination rate is $o(D^{-1}(t)/t)$. The boundary regime $H(t) = \Theta(D^{-1}(t)/t)$ remains open. Crucially, as the deadline function grows more quickly, this constraint weakens, thereby allowing the generator to hallucinate at a much slower rate while still achieving optimal lower density.

To simplify our proof, we introduce a weaker notion of timely density that relaxes the strict element-wise requirement to one more amenable to analysis.

Definition 3 (Prefix-Wise Density)**.** *For any two sequences S and R and function $F : \mathbb{N} \to \mathbb{N}$, define*

$$\mu_i^{\mathrm{pfx}}(S, R, F) := \frac{|S_{F(i)} \cap R_i|}{i}.$$

Then for a generator $\mathcal{A}$ and deadline function $D : \mathbb{N} \to \mathbb{N}$ define

$$\mu_{low}^{\mathrm{pfx}}(\mathcal{A}, D) := \inf \left\{ \inf_{K \in \mathcal{L}} \inf_{E \in \mathcal{E}(K)} \liminf_{i \to \infty} \mu_i^{\mathrm{pfx}}(\mathcal{A}(E), K, D) : \mathcal{L} \textit{ is countable} \right\}.$$

The prefix density metric measures how many elements of the prefix K_i have been generated by time $D(i)$, regardless of those elements' individual deadlines. Consequently, an element k_j for $j < i$ that is generated after its own deadline $D(j)$ may still contribute to the generator's performance when measured against a larger prefix K_i provided its generation time is at most $D(i)$. The next lemma shows that working with this prefix-wise metric is without loss: any constant lower bound on prefix-wise density for all feasible profiles suffices to obtain the same guarantee under the original element-wise notion.

Lemma 2 (Reduction Between Density Metrics)**.** *Fix any density $\rho \in [0, 1]$ and feasible profile (D, H). There exists a feasible profile $(D_{\mathrm{pfx}}, H_{\mathrm{pfx}})$ such that the following holds.*

Suppose a generator $\mathcal{A}$ achieves $\mu_{low}^{\mathrm{pfx}}(\mathcal{A}, D_{\mathrm{pfx}}) \geq \rho$ almost surely for each fixed instance and satisfies the following: for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $E \in \mathcal{E}(K)$, there exists a finite time T_{pfx} such that

$$H_{\mathcal{A}}^{\mathcal{L},K,E}(t) \leq H_{\mathrm{pfx}}(t) \qquad \forall t \geq T_{\mathrm{pfx}}.$$

Then $\mathcal{A}$ achieves $\mu_{low}(\mathcal{A}, D) \geq \rho$ almost surely for each fixed instance and satisfies the following: for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $E \in \mathcal{E}(K)$, there exists a finite time T such that

$$H_{\mathcal{A}}^{\mathcal{L},K,E}(t) \leq H(t) \qquad \forall t \geq T.$$

It therefore remains to construct an algorithm that achieves density ρ with respect to the prefix metric for any feasible profile. To do so, we further reduce the problem to the setting without deadlines. In particular, for any feasible profile, we present a black-box construction that converts any deterministic eventually consistent dense generator that performs well in the limit into a timely generator with vanishing hallucination rate, while preserving its lower-density guarantee under the prefix-wise metric.

Lemma 3 (Black-Box Reduction)**.** *Fix any density $\rho \in [0,1]$ and feasible profile (D,H). For any deterministic eventually consistent generator $\mathcal{B}$ such that $\mu_{low}(\mathcal{B},\infty) = \rho$, there exists a randomized algorithm $\mathcal{A} = \mathcal{A}_{\mathcal{B}}$ achieving $\mu_{low}^{\mathrm{pfx}}(\mathcal{A},D) \geq \rho$ almost surely for each fixed instance and satisfying the following: for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $E \in \mathcal{E}(K)$, there exists a finite time T such that*

$$H_{\mathcal{A}}^{\mathcal{L},K,E}(t) \leq H(t) \qquad \forall t \geq T.$$

Furthermore, $\mathcal{A}$ only makes black-box calls to $\mathcal{B}$.

Combining these two lemmas with the eventually consistent generator of Kleinberg and Wei [18], which achieves lower density $1/2$ in the limit, yields the desired result. We defer a deeper discussion of Lemma 2 and Lemma 3 to the next two subsections.

3.1 Discussion of Lemma 2: reducing from element-wise to prefix-wise density

At a high level, the reduction can be viewed as a controlled time-dilation argument. Rather than attempting to satisfy the element-wise deadlines directly, we construct a slower-growing prefix time scale under which meeting prefix deadlines is sufficient to meet almost all element-wise deadlines.

Given a feasible element-wise profile (D,H), we construct a prefix deadline function D_{pfx} that grows more gradually. Then, for each prefix index i, we compare the prefix deadline $D_{\mathrm{pfx}}(i)$ with the element-wise deadlines $D(j)$ for $j \leq i$. This comparison identifies how many elements have deadlines that occur before the prefix deadline. By construction, any element whose deadline occurs at or after $D_{\mathrm{pfx}}(i)$ automatically satisfies its constraint if generated by time $D_{\mathrm{pfx}}(i)$. Thus, among elements of K_i generated by this time, only those with earlier deadlines may fail to meet their deadline. By choosing D_{pfx} to grow sufficiently slowly, we ensure that this set is a vanishing fraction of the prefix, so the loss in density is negligible.

The more delicate issue is ensuring that the earlier prefix deadlines remain compatible with the prescribed hallucination rate. Moving deadlines earlier can increase $D_{\mathrm{pfx}}^{-1}(t)$, so we must choose D_{pfx} to satisfy $D_{\mathrm{pfx}}^{-1}(t)/t = o(H(t))$. This allows us to choose a vanishing non-increasing rate H_{pfx} with $D_{\mathrm{pfx}}^{-1}(t)/t = o(H_{\mathrm{pfx}}(t))$ and $H_{\mathrm{pfx}}(t) \leq H(t)$ for all sufficiently large t. The hallucination bounds then transfer directly at the same time t.

The choice of D_{pfx} must therefore balance two competing requirements: it must give sufficiently early prefix deadlines that only a vanishing fraction of elements have earlier element-wise deadlines, while retaining the inverse-rate condition needed for feasibility. This necessitates a coupled choice of D_{pfx} and the prefix-wise hallucination rate H_{pfx}.

Below we give a proof sketch of Lemma 2. A full proof can be found in Appendix D.1.

Proof Sketch. We make the above argument precise. Let D_{pfx} be a prefix deadline function. For each prefix index i, define

$$r(i) := \max\big(\{0\} \cup \{j \leq i : D(j) < D_{\mathrm{pfx}}(i)\}\big),$$

to be the number of elements whose element-wise deadlines occur before the prefix deadline $D_{\mathrm{pfx}}(i)$.

By definition, any element among $\{k_{r(i)+1}, \ldots, k_i\}$ that is generated by time $D_{\mathrm{pfx}}(i)$ automatically satisfies its element-wise deadline. Thus, among the elements of K_i generated by time $D_{\mathrm{pfx}}(i)$, only those in $K_{r(i)}$ may fail to meet their deadlines. For every realized output sequence, this gives

$$\mu_i(\mathcal{A}(E),K,D) \geq \mu_i^{\mathrm{pfx}}(\mathcal{A}(E),K,D_{\mathrm{pfx}}) - \frac{r(i)}{i}.$$

By choosing D_{pfx} to grow sufficiently slowly relative to D, we ensure that $r(i) = o(i)$, so this loss is negligible when measuring density.

To control hallucinations, we relate time t to the prefix index via

$$\tau(t) := \min\{n \in \mathbb{N} : D_{\mathrm{pfx}}(n) \geq t\} = D_{\mathrm{pfx}}^{-1}(t).$$

The construction chooses D_{pfx} to be monotone and super-linear while ensuring both $r(i) = o(i)$ and $\tau(t)/t = o(H(t))$. The latter condition allows us to construct a vanishing non-increasing rate H_{pfx}

satisfying $\tau(t)/t = o(H_{\mathrm{pfx}}(t))$ and $H_{\mathrm{pfx}}(t) \le H(t)$ for all sufficiently large t. Thus $(D_{\mathrm{pfx}}, H_{\mathrm{pfx}})$ is feasible.

Taking lower limits in the density inequality transfers the almost-sure density guarantee for each fixed instance. On each such instance, the hallucination bound also transfers. In particular, for all sufficiently large t,

$$H_{\mathcal{A}}^{\mathcal{L},K,E}(t) \le H_{\mathrm{pfx}}(t) \le H(t),$$

establishing the desired reduction. □

3.2 Discussion of Lemma 3: a randomized algorithm for prefix-wise density

At a high level, the procedure in Algorithm 1 can be viewed as a controlled mixture of two processes. Rather than relying solely on the blackbox generator $\mathcal{B}$, which achieves density ρ only in the limit, we introduce a sparse sequence of speculative steps that directly target elements of the prefix. The goal is to ensure that, by time $t = D(i)$, sufficiently many elements of K_i have been generated, while keeping the total number of speculative steps small enough to stay within the hallucination constraint.

The algorithm first replaces D by an earlier deadline with growing epoch lengths. In particular, in the appendix we show that every feasible profile (D, H) admits a deadline $\widetilde{D} \le D$ such that $(\widetilde{D}, H)$ remains feasible and $\widetilde{D}(m) - \widetilde{D}(m-1) \to \infty$. Running the algorithm with these earlier deadlines preserves the desired guarantee under the original deadline, while keeping the same hallucination budget H. We continue to write D for the deadline used after this replacement. The construction and proof are given in Appendix D.2.

Algorithm 1 Speculative Blackbox Generation

Parameters: Feasible profile (D, H), blackbox $\mathcal{B}$
Replace D with $\widetilde{D}$, set $D(0) := 0$ and define $m(t) := \min\{m \in \mathbb{N} : t \le D(m)\} = D^{-1}(t)$
Choose an unbounded monotone non-decreasing $C : \mathbb{N} \to \mathbb{N}$ such that $\sum_{k \le m(t)} C(k) = o(tH(t))$ and $C(m) = o(D(m) - D(m-1))$.
for $m = 1, 2, \ldots$ **do**
  if $D(m) = D(m-1)$ **then**
    $\delta_m \leftarrow 0$
    continue
  $\delta_m \leftarrow \min\left\{1, \frac{10\,C(m)}{D(m) - D(m-1)}\right\}$
  for $t = D(m-1) + 1, \ldots, D(m)$ **do**
    Observe e_t
    Sample $\mathrm{decision}_t \sim \mathrm{Bern}(\delta_m)$ and $\mathrm{choice}_t \sim \mathrm{Unif}([C(m)])$
    if $\mathrm{decision}_t = 0$ **then**
      $o_t \leftarrow \mathcal{B}(e_{1:t})$ // t-th output of black box
    else
      $o_t \leftarrow \mathsf{NextUnused}(L^{\mathrm{choice}_t})$ // first element not yet observed or output by $\mathcal{A}$
    Generate o_t

We partition time into epochs $[D(m-1)+1, D(m)]$, skipping empty epochs. Within each epoch m, we maintain a list $L^1, \ldots, L^{C(m)}$ of candidate languages, and during each round of every sufficiently large epoch we speculate with probability δ_m. On a speculative step at time t, the algorithm selects an index $\mathrm{choice}_t \in [C(m)]$ uniformly at random and outputs the *next unused* element of L^{choice_t}. By this choice of δ_m, the algorithm performs $\Theta(C(m))$ speculative steps in expectation in every sufficiently large epoch m. Moreover, since $C(m) = o(D(m) - D(m-1))$, we have $\delta_m \to 0$, so speculation—and, after the blackbox becomes consistent, hallucination—becomes increasingly rare across epochs.

Let $i^\star := \min\{j : L^j = K\}$. Since $C(m) \to \infty$, the true target $K = L^{i^\star}$ is eventually included among these candidates. From that point onward, each speculative step in epoch m selects index $i^\star$ with probability $1/C(m)$, so every sufficiently large epoch yields a constant expected number of fresh elements of K through these speculations. Consequently, for all sufficiently large i, at least i such speculations occur by time $D(i)$, except with probability exponentially small in i. This behavior is illustrated in Figure 1.

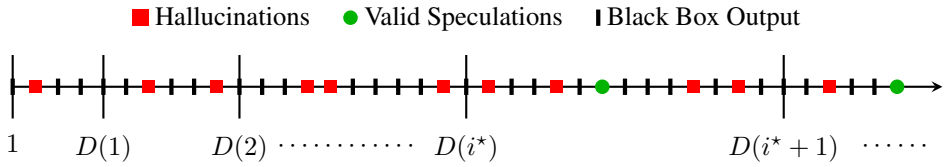

Figure 1: Visual for behavior of SBG algorithm: every tick mark represents a generated element and larger tick marks represent deadlines for corresponding prefixes of the target language.

At the same time, we choose C so that $\sum_{k\le m(t)} C(k) = o(tH(t))$. This bounds the expected number of speculative steps by time t by $o(tH(t))$. Since the blackbox contributes only finitely many hallucinating outputs on each fixed instance, the total expected number of hallucinations eventually remains within the budget $tH(t)$. Thus, speculation is frequent enough to ensure linear coverage of K, while remaining negligible relative to the hallucination budget.

Below we give a proof sketch of Lemma 3. A full proof can be found in Appendix D.2.

Proof Sketch. Fix any countable collection $\mathcal{L}$, any $K \in \mathcal{L}$, and any enumeration $E \in \mathcal{E}(K)$. We analyze the algorithm after the deadline replacement described above. We first bound the expected number of hallucinations of the generator. Since the blackbox is deterministic and eventually consistent, it produces outputs outside K on only finitely many rounds; denote this number by $N_{\mathcal{B}}(E,K)$. Counting every hallucinating round, including repetitions, we have

$$tH_{\mathcal{A}}^{\mathcal{L},K,E}(t) = \mathbb{E}\left[\sum_{s=1}^{t} \mathbf{1}\{o_s \notin K\}\right] \le N_{\mathcal{B}}(E,K) + 10 \sum_{m\le m(t)} C(m) = o(tH(t)),$$

where we used the choice of C and $tH(t) \to \infty$. Thus $H_{\mathcal{A}}^{\mathcal{L},K,E}(t) \le H(t)$ for all sufficiently large t, with the threshold allowed to depend on the fixed instance.

We next argue that the algorithm speculates sufficiently many elements of K. Once $C(m) \ge i^\star$ and m is sufficiently large, the generator is expected to speculate a constant number of times per epoch. Summing over epochs gives at least $\Omega(i)$ such speculations in expectation by time $D(i)$ for all sufficiently large i. Let S_i be the event that at least i such speculations occur by this time. A Chernoff bound gives $\Pr(S_i^c) \le e^{-ci}$ for some constant $c > 0$. On S_i, every element of K_i has either been observed or generated by time $D(i)$, since each speculation from K outputs the earliest element unused by both players.

It remains to show that the contribution of $\mathcal{B}$ is not destroyed by the random deletions induced by speculation. Fix $0 < \epsilon < 1$, and let U_i be the set of elements of K_i eventually output by $\mathcal{B}$. By the blackbox density guarantee, $|U_i| \ge (\rho - \epsilon/2)i$ for all sufficiently large i. Partition U_i into $U_i^{\le}$ and $U_i^{>}$ according to whether the blackbox first outputs each element by time $D(i)$ or afterward. If $u \in U_i^{>}$, the adversary cannot have revealed u by time $D(i)$, since otherwise the blackbox's later output of u would be invalid. Consequently, on S_i, every element of $U_i^{>}$ has already been generated by $\mathcal{A}$ by time $D(i)$.

For elements in $U_i^{\le}$, we count those already generated by $\mathcal{A}$ before their first blackbox occurrence separately and apply the retention argument to the remaining first occurrences. There is a finite time after which $\delta_{m(t)} \le \epsilon/2$, and the preceding rounds account for at most $\epsilon i/2$ elements for sufficiently large i. On subsequent rounds, whether a first occurrence remains ungenerated is determined before the retention coin is sampled. A martingale concentration argument via Freedman's inequality therefore shows that the total loss from $U_i^{\le}$ is at most $3\epsilon i/2$, except with probability $e^{-c_\epsilon i}$ for some $c_\epsilon > 0$. Counting first occurrences ensures that all retained contributions are distinct.

Combining the contributions from $U_i^{\le}$ and $U_i^{>}$ yields

$$\Pr\Big(\mu_i^{\mathrm{pfx}}(\mathcal{A}(E),K,D) < \rho - 2\epsilon\Big) \le e^{-ci} + e^{-c_\epsilon i}$$

for all sufficiently large deterministic i. These probabilities are summable. Applying the first Borel–Cantelli lemma and then taking $\epsilon = 1/n$ for all integers $n \ge 2$ proves the desired density guarantee almost surely for each fixed instance. Since the deadline replacement only moves deadlines earlier, the same guarantee holds under the original deadline. □

In fact, the algorithm satisfies a stronger guarantee: the union of the outputs of the generator and the adversary achieves prefix-wise lower density 1 under the deadline function D almost surely for each fixed instance. Indeed, the speculation bound and the first Borel–Cantelli lemma imply that, almost surely, for all sufficiently large i, the generator has made at least i speculations from K by time $D(i)$. Each such speculation outputs the earliest element of K unused by both players, so every element of K_i must have been observed or generated by this time. Thus, $K_i \subseteq E_{D(i)} \cup \mathcal{A}(E)_{D(i)}$ for all sufficiently large i almost surely, yielding prefix-wise density 1.

Finally, the growth of C determines when the target $K = L^{i^\star}$ enters the candidate list, at epoch $\min\{m : C(m) \geq i^\star\}$. From then on, it can be selected in every nonempty epoch. When C grows slowly, this inclusion is delayed, yielding an initial phase in which speculation targets other candidate languages, followed by a phase in which sufficiently many epochs provide a linear number of speculations from K. The constraints on C therefore relate the epoch lengths and hallucination budget to when the target language can be directly included in speculation; the onset of the density guarantee also depends on the blackbox's behavior on the fixed instance.

4 Discussion and open directions

Our results highlight a central and somewhat counterintuitive phenomenon: controlled hallucination can provide the flexibility needed for timely dense generation. Therefore, hallucination should not merely be viewed as an artifact of imperfect generation, but as a necessary ingredient for general guarantees of breadth when evidence is bounded. In particular, our impossibility results show that any eventually consistent generator must sacrifice timely density, while our constructions demonstrate that allowing controlled hallucination restores the ability to generate densely in a timely fashion. This suggests that hallucination plays a fundamental role in enabling exploration. A deeper investigation of this phenomenon remains an important direction for future work.

For clarity of exposition, we have presented our results in the setting where the adversary sequentially reveals a total enumeration of the target language. However, the scope of our techniques extends beyond this setting: the reduction is agnostic to the observed sequence and applies whenever an eventually consistent blackbox generator achieves nontrivial lower density in the limit on every admissible observation sequence (see Appendix D for more details). In particular, our results extend to the partial enumeration model of [18] and the finite contamination model of [20]. More generally, the reduction transfers suitable density guarantees in other data models into timely density guarantees with a controlled hallucination rate. Furthermore, Section 2 establishes an impossibility for consistent generators to achieve nontrivial timely lower density guarantees, this limitation can be circumvented under a weaker notion, first introduced in [17], that requires density only infinitely often at isolated indices.

Several quantitative questions remain open. For linear deadlines $D(i) = ci$, our impossibility result rules out every positive worst-case timely lower density under a vanishing hallucination rate, regardless of the finite constant c. This leaves open the tradeoff when the hallucination rate is allowed to remain nonvanishing. Our present blackbox reduction does not resolve this question and understanding this regime may require different algorithmic ideas.

Our results suggest a complementary notion of timeliness at the language level. Although our deadlines constrain when individual strings must be generated to receive credit, our density guarantees remain asymptotic and instance-wise. They do not quantify how quickly a generator approaches its density guarantee. One could ask how this convergence depends on the position of the target language within the ordered collection. Our results suggest a tension between stringent string-level deadlines and the onset of good generation: the growth of the candidate list determines when speculation can begin to target a given language, while the requirement of sparse hallucination constrains that growth.

Finally, we note that—much like [17, 18]—our model postulates a fixed preference ordering on strings that is known to the generator, and relative to which (timely) breadth is evaluated. In reality, however, this preference may be context dependent and subject to learning from data, as is the case for distributional plausibility in LLMs and other generative models. Exploring such generalizations is therefore a promising direction for future work.

Acknowledgments. Atul Ganju and Travis McVoy were supported by the Viterbi School of Engineering Graduate Student Fellowship. Shang-Hua Teng was supported in part by the Simons Foundation Investigator Award. Shaddin Dughmi was supported by NSF Grant CCF-2432219.

References

[1] D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45(2):117–135, 1980.

[2] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

[4] Y. Bai, D. Panigrahi, and I. Zhang. Language generation in the limit: Noise, loss, and feedback. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 794–816. SIAM, 2026.

[5] M. Charikar and C. Pabbaraju. Exploring facets of language generation in the limit. In N. Haghtalab and A. Moitra, editors, *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 854–887. PMLR, 30 Jun–04 Jul 2025.

[6] M. Charikar and C. Pabbaraju. Pareto-optimal non-uniform language generation. In *37th International Conference on Algorithmic Learning Theory*, 2026.

[7] T. Feng, T. H. Trinh, G. Bingham, D. Hwang, Y. Chervonyi, J. Jung, J. Lee, C. Pagano, S. hyun Kim, F. Pasqualotto, S. Gukov, J. N. Lee, J. Kim, K. Hou, G. Ghiasi, Y. Tay, Y. Li, C. Kuang, Y. Liu, H. Lin, E. Z. Liu, N. Nayakanti, X. Yang, H.-T. Cheng, D. Hassabis, K. Kavukcuoglu, Q. V. Le, and T. Luong. Towards autonomous mathematics research, 2026.

[8] D. A. Freedman. On tail probabilities for martingales. *The Annals of Probability*, 3(1):100–118, 1975.

[9] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig. Pal: program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[10] E. M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[11] S. Hanneke, A. Karbasi, A. Mehrotra, and G. Velegkas. On union-closedness of language generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[12] A. T. Kalai and S. S. Vempala. Calibrated language models must hallucinate. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 160–171, New York, NY, USA, 2024. Association for Computing Machinery.

[13] A. Kalavasis, A. Mehrotra, and G. Velegkas. On the limits of language generation: Trade-offs between hallucination and mode-collapse. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, page 1732–1743, New York, NY, USA, 2025. Association for Computing Machinery.

[14] A. Kalavasis, A. Mehrotra, and G. Velegkas. On characterizations for language generation: Interplay of hallucinations, breadth, and stability. In *37th International Conference on Algorithmic Learning Theory*, 2026.

[15] A. Karbasi, O. Montasser, J. Sous, and G. Velegkas. (im)possibility of automated hallucination detection in large language models. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025.

[16] J. Kleinberg and S. Mullainathan. Language generation in the limit. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 66058–66079. Curran Associates, Inc., 2024.

[17] J. Kleinberg and F. Wei. Density measures for language generation. In *2025 IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 620–658, 2025.

[18] J. Kleinberg and F. Wei. Language generation and identification from partial enumeration: Tight density bounds and topological characterizations, 2025. arxiv preprint; accepted to STOC 2026.

[19] J. Li, Y. Xia, R. Yan, H. Sun, D. Zhao, and T.-Y. Liu. Stylized dialogue generation with multi-pass dual learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 28470–28481. Curran Associates, Inc., 2021.

[20] A. Mehrotra, G. Velegkas, X. Yu, and F. Zhou. Language generation with infinite contamination, 2025. arxiv preprint; to appear in COLT 2026.

[21] C. Peale, V. Raman, and O. Reingold. Representative language generation. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 48518–48541. PMLR, 13–19 Jul 2025.

[22] K. Pillutla, S. Swayamdipta, R. Zellers, J. Thickstun, S. Welleck, Y. Choi, and Z. Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4816–4828. Curran Associates, Inc., 2021.

[23] C. Qian, P. Han, Q. Luo, B. He, X. Chen, Y. Zhang, H. Du, J. Yao, X. Yang, D. Zhang, Y. Li, and H. Ji. EscapeBench: Towards advancing creative intelligence of language model agents. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 798–820, Vienna, Austria, July 2025. Association for Computational Linguistics.

[24] A. Raman and V. Raman. Generation from noisy examples. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 51079–51093. PMLR, 13–19 Jul 2025.

[25] V. Raman, J. Li, and A. Tewari. Generation through the lens of learning theory. In N. Haghtalab and A. Moitra, editors, *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 4740–4776. PMLR, 30 Jun–04 Jul 2025.

[26] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022.

[27] J. Zhang, S. Yu, D. Chong, A. Sicilia, M. R. Tomz, C. D. Manning, and W. Shi. Verbalized sampling: How to mitigate mode collapse and unlock llm diversity, 2025.

A Additional discussion of related work

The literature on language generation is vast and multifaceted. We therefore cannot do all of it justice, and emphasize only the most relevant papers to our work.

We work in the framework of language generation in the limit, as posed by introduced by Kleinberg and Mullainathan [16]. This builds on analogous questions for language identification introduced by Gold [10] and studied by Angluin [1]. In contrast to language identification, [16] show that consistent generation is possible in general. However, somewhat tellingly their algorithm suffers mode collapse in a manner which seems attributable to a tension with consistency; this sets the stage for much of the subsequent work exploring the interplay between consistency and breadth—a theme which permeates our paper as well.

Several notions of breadth have been considered in the theoretical literature on language generation. Strong notions which require the generator to eventually generate from all or almost all unseen strings in K were studied by Kalavasis et al [13, 14], leading to largely negative results in general settings for consistent generators. Similar impossibility results also hold for a notion of breadth, called exhaustive generation, as shown by Peale et al [21].

Things took a turn for the positive with the work of Kleinberg and Wei [17], which proposed measuring breadth with respect to a fixed global order on strings—this is the "preference order" we also consider in our paper. They showed that nontrivial breadth guarantees—in the sense of positive density—were achievable by consistent generators with respect to such an order. Subsequent work by the same authors in [18] strengthens those results to an optimal density of $1/2$.

It is this latter line of work which is most related to our approach in this paper. We start with the breadth notions in [17, 18] and naturally strengthen them to incorporate timely generation, measured with respect to the same preference order on strings. We find that this modification changes the picture, implying strong impossibility results for consistent generation with timely breadth that are reminiscent of the negative results in aforementioned previous approaches to reconciling breadth and consistency. However, the presence of the global order on strings proves particularly useful here: by prioritizing preferred strings carefully over time, we are able to obtain simultaneous and meaningful guarantees both on timely breadth as well as the hallucination rate.

There are many other works which consider largely orthogonal concerns to ours in language generation. This includes work which explores stronger guarantees on the time to consistency [25, 5, 6, 11], as well as works which strengthen the adversary to allow for partial or noisy training data [18, 24, 20, 4]. The latter group of works is consequential to ours to the extent that our positive result—by virtue of its black-box nature—is well-positioned to port over our timely breadth guarantees to such more general settings as we will show in Appendix D.3.

Finally, we would be remiss not to mention some of the work examining statistical perspectives on hallucination. We single out [12, 15] as arguing for the inevitability of hallucination in both theory and practice. This is in-keeping with a key message of our paper, that hallucination serves to enable learning with breadth and can be minimized yet not completely eliminated.

B Examples of measure-zero chains

In this section we provide examples of language families that fall under Definition 1. First we restate the definition of a Measure-Zero Chain.

Definition 4 (Definition 1, restated)**.** *A collection $\mathcal{L}_{chain} = \{L^\infty\} \cup \bigcup_{j\geq 1}\{L^j\}$ forms a measure-zero chain if $L^1 \subseteq L^2 \subseteq \cdots \subseteq L^\infty$ is a nested sequence of languages for which:*

$$\bigcup_{j\geq 1} L^j = L^\infty \qquad \textit{and} \qquad \liminf_{i\to\infty} \mu_i(L^j, L^\infty, \infty) = 0 \qquad \forall j \in \mathbb{N}.$$

We now provide a very simple language family which satisfies this definition.

Example 1 (Block Partition Construction)**.** *Let $U = \mathbb{N}$ with the canonical order, and partition U into consecutive disjoint blocks*

$$U = \bigsqcup_{m\geq 0} B_m, \qquad B_m = [2^m, 2^{m+1} - 1],$$

so that $|B_m| = 2^m$. For each $m \geq 0$, let $b^\star_m \in B_m$ denote the first element of B_m in the canonical order, i.e., $b^\star_m = 2^m$.

Define the limiting language $L^\infty := U$, and for each $j \in \mathbb{N}$,

$$L^j := \left(\bigcup_{m\leq j} B_m\right) \cup \left(\bigcup_{m>j} \{b^\star_m\}\right).$$

Then, by construction $L^1 \subseteq L^2 \subseteq \cdots \subseteq L^\infty$ and $\bigcup_{j\geq 1} L^j = L^\infty$. Moreover, for each fixed j,

$$\liminf_{i\to\infty} \mu_i(L^j, L^\infty, \infty) = 0,$$

since, up to index n, the number of blocks is on the order of $\log n$, and beyond the first j blocks each contributes at most one element to L^j. Thus, $|L^j \cap [n]| \leq O(\log n)$, while $|L^\infty \cap [n]| = n$, implying that the density vanishes.

This construction is not tied to the particular dyadic choice of block sizes. More generally, the same example goes through whenever the block sizes $|B_n|$ grow according to any super-constant function of n, and the distinguished element $b_n^\star$ may be chosen arbitrarily within each block. The only property used is that each fixed L^j contains all elements from the first finitely many blocks, but only one element from each later block, so its density inside L^∞ vanishes as the block sizes grow.

Interestingly, another instructive example of a measure-zero chain arises in the construction of Kleinberg and Wei in their initial study of lower density in the limit. Below we provide this example and show that it satisfies the definition of a measure-zero chain.

Example 2 (Marker-Interval Construction (Example 7 in [17]))**.** *Let $U = \mathbb{N}$ with the canonical order. Define a sequence of marker locations $\{a_i\}_{i\geq 0}$ by $a_0 = 0$, and $a_i = 3^i$ for $i \geq 1$. For each $j \in \mathbb{N}$, define the language*

$$L^j := \bigcup_{i\geq 0} [a_i, a_i + j],$$

where we can think of this construction as attaching an interval of length j at each marker a_i. Let $L^\infty := \mathbb{N}$.

Then, by construction $L^1 \subseteq L^2 \subseteq \cdots \subseteq L^\infty$ and $\bigcup_{j\geq 1} L^j = L^\infty$. Moreover, for each fixed j,

$$\liminf_{i\to\infty} \mu_i(L^j, L^\infty, \infty) = 0,$$

since, up to index n, the number of markers is on the order of $\log n$, and each contributes at most $j + 1$ elements to L^j. Thus, $|L^j \cap [n]| \leq (j + 1) \cdot O(\log n)$, while $|L^\infty \cap [n]| = n$, implying that the density vanishes.

This construction is not tied to the specific choice $a_i = 3^i$. More generally, like in Example 1, the same example goes through for any sequence of markers $\{a_i\}$ that grows super-constantly, with intervals of length j attached at each marker. The only property used is that, up to index n, the number of markers is $o(n)$, so each fixed L^j captures only $o(n)$ elements of $L^\infty = \mathbb{N}$, implying that the density vanishes.

More generally, both constructions above are instances of a common pattern: we partition U into disjoint pieces whose sizes grow, and define L^j by retaining each piece entirely up to some index j, while selecting only a vanishing fraction of each subsequent piece. Whenever the total contribution of these partial selections is negligible compared to the full pieces, the resulting chain satisfies

$$\liminf_{i\to\infty} \mu_i(L^j, L^\infty, \infty) = 0.$$

Note that such a language family need only be contained in $\mathcal{L}$ for the impossibility result of Lemma 1 to kick in.

C Impossibility results

Recall the definition in Section 2 of a measure-zero chain defined in measure-zero chain, which we restate for convenience in Appendix B. We will now show that any language collection $\mathcal{L}$ which contains a measure-zero chain witnesses the impossibility result of Lemma 1

Lemma 1 (Hallucination Barrier)**.** *There exists a language family $\mathcal{L}$ such that, for any monotone non-decreasing deadline function $D : \mathbb{N} \to \mathbb{N}$ and any randomized generator $\mathcal{A}$ satisfying*

$$H_{\mathcal{A}}^{\mathcal{L},K,E}(t) = o\left(\frac{D^{-1}(t)}{t}\right)$$

for every $K \in \mathcal{L}$ and every $E \in \mathcal{E}(K)$, there exist a target language $K \in \mathcal{L}$ and a total enumeration $E \in \mathcal{E}(K)$ such that

$$\Pr\Big(\liminf_{i\to\infty} \mu_i(\mathcal{A}(E), K, D) = 0\Big) = 1.$$

Proof. We will prove the result using a measure-zero chain, as defined above in Definition 1. Specifically, we construct a total enumeration X^∞ of the chain's limiting language L^∞ such that $\liminf_{i\to\infty} \mu_i^{\mathrm{pfx}}(\mathcal{A}(X^\infty), L^\infty, D) = 0$ almost surely, where the prefix-wise density metric is formally defined in Definition 3. Since element-wise density is bounded above by prefix-wise density, this suffices to prove the result.

Fix a chain $\mathcal{L} = \mathcal{L}_{chain} = \{L^\infty\} \cup \{L^j : j \in \mathbb{N}\}$. We construct a total enumeration X^∞ of L^∞ incrementally, in stages indexed by $j \in \mathbb{N}$. At stage j, we temporarily pretend that the target language is L^j: we choose an enumeration of L^j extending the prefix constructed so far, follow it for a sufficiently long finite interval ending at time t_j, and then pass to stage $j+1$. We will choose indices i_j and set $t_j := D(i_j)$ so that the generator produces only a negligible fraction of elements from $L_{i_j}^\infty$ by time t_j. This will force $\liminf_{i\to\infty} \mu_i^{\mathrm{pfx}}(\mathcal{A}(X^\infty), L^\infty, D) = 0$ almost surely.

We represent the randomized algorithm as a deterministic map indexed by a random seed: for each seed R, let $\mathcal{A}_R$ denote the corresponding deterministic algorithm.

Let $t_0 := i_0 := 0$. Suppose inductively that stages $1, \ldots, j-1$ have already been defined, so that the current prefix of X^∞ has length t_{j-1}. Since the chain is nested, this prefix can be extended to a total enumeration $X^{(j)}$ of L^j. Choose and fix such an enumeration so that every element of $L^j \cap L_j^\infty$ has appeared by time $t_{j-1} + j$.

By the assumed hallucination rate guarantee on $X^{(j)}$ with respect to L^j, we have

$$H_{\mathcal{A}}^{\mathcal{L},L^j,X^{(j)}}(t) = o\left(\frac{D^{-1}(t)}{t}\right).$$

Since the number of distinct hallucinated elements is at most the number of hallucinating rounds, we have

$$\mathbb{E}\left[\frac{|\mathcal{A}_R(X^{(j)})_t \setminus L^j|}{D^{-1}(t)}\right] \le \frac{t}{D^{-1}(t)} \cdot H_{\mathcal{A}}^{\mathcal{L},L^j,X^{(j)}}(t) = o(1).$$

Furthermore, by Definition 1, $\liminf_{i\to\infty} \mu_i(L^j, L^\infty, \infty) = 0$. Then, since $D^{-1}(D(i)) = i$ and $D(i) \to \infty$, we may choose a sufficiently large index $i_j > i_{j-1}$ and set $t_j := D(i_j) \ge t_{j-1} + j$ such that

$$\frac{|L^j \cap L_{i_j}^\infty|}{i_j} \le 2^{-(j+1)} \qquad \text{and} \qquad \mathbb{E}\left[\frac{|\mathcal{A}_R(X^{(j)})_{t_j} \setminus L^j|}{i_j}\right] \le 2^{-(2j+1)}.$$

We then let X^∞ agree with $X^{(j)}$ on the interval $\{t_{j-1}+1, \ldots, t_j\}$. This recursively defines an infinite sequence X^∞. Since $\bigcup_{j\ge1} L^j = L^\infty$, every element of L^∞ belongs to $L^j \cap L_j^\infty$ for all sufficiently large j, and hence appears at some finite stage. Thus, X^∞ is a total enumeration of L^∞.

Now fix j. Since X^∞ and $X^{(j)}$ agree through time t_j, for every fixed seed R, the generator produces identical outputs on both runs up to time t_j. Therefore, $\mathcal{A}_R(X^\infty)_{t_j} = \mathcal{A}_R(X^{(j)})_{t_j}$. Hence,

$$\begin{aligned}
\mu_{i_j}^{\mathrm{pfx}}(\mathcal{A}_R(X^\infty), L^\infty, D) &= \frac{|\mathcal{A}_R(X^\infty)_{D(i_j)} \cap L_{i_j}^\infty|}{i_j} \\
&= \frac{|\mathcal{A}_R(X^\infty)_{t_j} \cap L_{i_j}^\infty|}{i_j} \\
&\le \frac{|L^j \cap L_{i_j}^\infty|}{i_j} + \frac{|\mathcal{A}_R(X^{(j)})_{t_j} \setminus L^j|}{i_j} \\
&\le 2^{-(j+1)} + \frac{|\mathcal{A}_R(X^{(j)})_{t_j} \setminus L^j|}{i_j},
\end{aligned}$$

where it suffices to show that the second term is eventually small almost surely. To bound the second term, we consider the event E_j that the generator outputs more than $2^{-(j+1)} i_j$ distinct elements outside L^j when run on $X^{(j)}$ by time t_j. Then, by Markov's inequality, we have

$$\Pr(E_j) = \Pr\left(\frac{|\mathcal{A}_R(X^{(j)})_{t_j} \setminus L^j|}{i_j} > 2^{-(j+1)}\right) \le \frac{2^{-(2j+1)}}{2^{-(j+1)}} = 2^{-j}.$$

Since this probability is exponentially decreasing in j, the first Borel–Cantelli lemma implies that only finitely many of the events E_j occur. Hence, almost surely, for all sufficiently large j,

$$\frac{|\mathcal{A}_R(X^{(j)})_{t_j} \setminus L^j|}{i_j} \leq 2^{-(j+1)}.$$

Therefore, almost surely, for all sufficiently large j,

$$\mu_{i_j}^{\mathrm{pfx}}(\mathcal{A}_R(X^\infty), L^\infty, D) \leq 2^{-(j+1)} + 2^{-(j+1)} = 2^{-j}.$$

Since $i_j > i_{j-1}$, this bound holds along a subsequence tending to infinity, and we can therefore conclude that $\liminf_{i\to\infty} \mu_i^{\mathrm{pfx}}(\mathcal{A}_R(X^\infty), L^\infty, D) = 0$ almost surely. Finally, since D is monotone non-decreasing, we have

$$\mu_i(\mathcal{A}_R(X^\infty), L^\infty, D) \leq \mu_i^{\mathrm{pfx}}(\mathcal{A}_R(X^\infty), L^\infty, D).$$

Thus,

$$\Pr\Big(\liminf_{i\to\infty} \mu_i(\mathcal{A}(X^\infty), L^\infty, D) = 0\Big) = 1.$$

Taking $K = L^\infty$ and $E = X^\infty$ proves the claim. □

Theorem 1 (Impossibility for Eventually Consistent Generators)**.** *There exists a language family $\mathcal{L}$ such that, for every deadline function $D : \mathbb{N} \to \mathbb{N}$ and every (potentially randomized) eventually consistent generator $\mathcal{A}$, there exist a target language $K \in \mathcal{L}$ and a total enumeration $E \in \mathcal{E}(K)$ such that*

$$\Pr\Big(\liminf_{i\to\infty} \mu_i(\mathcal{A}(E), K, D) = 0\Big) = 1.$$

It follows that $\mu_{low}(\mathcal{A}, D) = 0$ almost surely for every deadline function D and every eventually-consistent generator $\mathcal{A}$.

Proof. This follows from observing that eventual consistency implies the condition in Lemma 1. □

Theorem 2 (Impossibility for Linear-Time Generators)**.** *There exists a language family $\mathcal{L}$ such that, for every deadline function $D(i) = O(i)$ and every generator $\mathcal{A}$ with vanishing hallucination rate, there exist a target language $K \in \mathcal{L}$ and a total enumeration $E \in \mathcal{E}(K)$ such that*

$$\Pr\Big(\liminf_{i\to\infty} \mu_i(\mathcal{A}(E), K, D) = 0\Big) = 1.$$

It follows that $\mu_{low}(\mathcal{A}, D) = 0$ almost surely whenever $D(i) = O(i)$ and the generator $\mathcal{A}$ has a vanishing hallucination rate.

Proof. This follows immediately from Lemma 1 by taking $D(i) = O(i)$. □

D A randomized algorithm with optimal lower density guarantees

In this section, we discuss an algorithm that converts density guarantees in the limit into timely element-wise density guarantees with controlled hallucination. We will in fact do this for a general data model, extending our reduction beyond the case where the adversary totally enumerates the target language to the generator. The reduction applies whenever the data model admits a deterministic eventually consistent generator with the desired density in the limit. We start by defining the more general framework we consider.

We consider the same two-player game between an adversary and a generator where both players have access to $\mathcal{L}$ and U. A data model $\mathfrak{D}$ specifies a nonempty set $\mathfrak{D}(K) \subseteq U^{\mathbb{N}}$ of admissible observation sequences for each infinite language $K \subseteq U$. The adversary first selects a target language $K \in \mathcal{L}$ and generates an observation sequence $X = (x_1, x_2, \ldots)$, where $X \in \mathfrak{D}(K)$. The data model is held fixed as $\mathcal{L}$ varies in the density definitions below. The precise structure of $\mathfrak{D}(K)$ depends on the model, and we give formal instantiations later in this section.

The generator observes the sequence X sequentially. At time t, after having observed $x_{1:t}$, the generator outputs $o_t = \mathcal{A}_t(x_{1:t}) \in U \setminus \{x_1, \ldots, x_t\}$. The generator may repeat its own previous

outputs, provided they have not been observed from the adversary. The generator may be randomized, and we suppress its internal randomness in the notation $\mathcal{A}_t(x_{1:t})$. All probabilities and expectations are taken over the generator's internal randomness.

For each fixed instance $(\mathcal{L}, K, X)$, the hallucination rate is defined as before. Eventual consistency is also defined as before, with the requirement imposed on every admissible sequence $X \in \mathfrak{D}(K)$.

The generator's objective, as before, is to maximize the density of its output within the target language, counting distinct output strings. For deterministic generators, we define the upper and lower element-wise densities as

$$\mu_{up}(\mathcal{A}, D) = \inf \left\{ \inf_{K \in \mathcal{L}} \inf_{X \in \mathfrak{D}(K)} \limsup_{i \to \infty} \mu_i(\mathcal{A}(X), K, D) : \mathcal{L} \text{ is countable} \right\},$$

$$\mu_{low}(\mathcal{A}, D) = \inf \left\{ \inf_{K \in \mathcal{L}} \inf_{X \in \mathfrak{D}(K)} \liminf_{i \to \infty} \mu_i(\mathcal{A}(X), K, D) : \mathcal{L} \text{ is countable} \right\},$$

and analogously define the upper and lower prefix-wise densities as

$$\mu^{\text{pfx}}_{up}(\mathcal{A}, D) = \inf \left\{ \inf_{K \in \mathcal{L}} \inf_{X \in \mathfrak{D}(K)} \limsup_{i \to \infty} \mu^{\text{pfx}}_i(\mathcal{A}(X), K, D) : \mathcal{L} \text{ is countable} \right\},$$

$$\mu^{\text{pfx}}_{low}(\mathcal{A}, D) = \inf \left\{ \inf_{K \in \mathcal{L}} \inf_{X \in \mathfrak{D}(K)} \liminf_{i \to \infty} \mu^{\text{pfx}}_i(\mathcal{A}(X), K, D) : \mathcal{L} \text{ is countable} \right\}.$$

We suppress the dependence on $\mathfrak{D}$ in this notation. For randomized generators, density guarantees are understood almost surely for each fixed instance. For example, $\mu_{low}(\mathcal{A}, D) \geq \rho$ means that, for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $X \in \mathfrak{D}(K)$,

$$\Pr\Big(\liminf_{i \to \infty} \mu_i(\mathcal{A}(X), K, D) \geq \rho\Big) = 1,$$

and analogously for the other density guarantees.

The framework above is intentionally agnostic to how the adversary generates the observation sequence. We now describe several natural instantiations.

Total enumeration. We recover the classical model of [16], where the adversary reveals only elements of the target language. Formally, for each $K \in \mathcal{L}$, define $\mathfrak{D}_{\text{enum}}(K)$ to be the set of all sequences X such that $x_t \in K$ for every t, and every element of K appears at least once in X. Thus, X is a total enumeration of K, potentially with repetitions. This recovers the standard setting in which the generator receives only truthful observations and eventually sees the entire target language.

Partial enumeration. A natural relaxation is to allow the adversary to reveal only part of the target language. For each $\alpha \in [0, 1]$ and $K \in \mathcal{L}$, define the following data model, following [18]:

$$\mathfrak{D}_{\text{part},\alpha}(K) := \left\{ X \in U^{\mathbb{N}} : x_t \in K \; \forall t, \quad |\{x_t : t \in \mathbb{N}\}| = \infty, \quad \mu_{low}(X, K, \infty) \geq \alpha \right\}.$$

Equivalently, for every $K \in \mathcal{L}$, the sequence $X \in \mathfrak{D}_{\text{part},\alpha}(K)$ is an enumeration, with possible repetitions, of an infinite subset of K whose lower density in K is at least α. In this model, the generator receives only truthful observations, but may never observe large portions of the target language. The parameter α controls how much of the target language must be revealed in the limit.

Finite contamination. Another natural relaxation is to allow the adversary both to omit finitely many target strings and to insert finitely many distinct non-target strings. For each $K \in \mathcal{L}$, following [20], define

$$\mathfrak{D}_{\text{fin-cont}}(K) := \left\{ X \in U^{\mathbb{N}} : |\{x_t : t \in \mathbb{N},\ x_t \notin K\}| < \infty, \quad |K \setminus \{x_t : t \in \mathbb{N}\}| < \infty \right\}.$$

Equivalently, $X \in \mathfrak{D}_{\text{fin-cont}}(K)$ is an enumeration, with possible repetitions, of an infinite set whose symmetric difference with K is finite. In this model, the generator may observe finitely many distinct false positives, possibly repeated infinitely often, and may permanently miss finitely many elements of the target language.

Hybrid models. The framework also accommodates mixtures of the above behaviors. For example, one may consider sequences that enumerate only an infinite subset $S \subseteq K$ while also allowing finitely many distinct non-target strings. Such models interpolate between partial information and adversarial corruption, and are useful for studying the robustness of generation guarantees beyond the idealized enumeration setting.

D.1 Reduction between density metrics

In this section we provide a proof of the reduction from achieving lower density under the element-wise metric to achieving it under the prefix-wise metric. We start with the lemma statement and break it down into subproblems as per the discussion in Section 3.1. We will prove the lemma after establishing the auxiliary constructions. This reduction is data model agnostic and therefore generalizes to the framework above.

Throughout, for a monotone nondecreasing unbounded function $F : \mathbb{N} \to \mathbb{N}$, we use the generalized inverse

$$F^{-1}(t) := \min\{n \in \mathbb{N} : F(n) \geq t\}.$$

We now provide the generalization of Lemma 2 to the general framework.

Lemma 4 (Reduction Between Density Metrics for General Data Models)**.** *Fix any data model $\mathfrak{D}$, density $\rho \in [0, 1]$, and feasible profile (D, H). There exists a feasible profile $(D_{\mathrm{pfx}}, H_{\mathrm{pfx}})$ such that the following holds.*

Suppose a generator $\mathcal{A}$ achieves $\mu^{\mathrm{pfx}}_{low}(\mathcal{A}, D_{\mathrm{pfx}}) \geq \rho$ almost surely for each fixed instance and satisfies the following: for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $X \in \mathfrak{D}(K)$, there exists a finite time T_{pfx} such that

$$H^{\mathcal{L},K,X}_{\mathcal{A}}(t) \leq H_{\mathrm{pfx}}(t) \qquad \forall t \geq T_{\mathrm{pfx}}.$$

Then $\mathcal{A}$ achieves $\mu_{low}(\mathcal{A}, D) \geq \rho$ almost surely for each fixed instance and satisfies the following: for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $X \in \mathfrak{D}(K)$, there exists a finite time T such that

$$H^{\mathcal{L},K,X}_{\mathcal{A}}(t) \leq H(t) \qquad \forall t \geq T.$$

The following are auxiliary lemmas, and their corresponding proofs, used to prove the above lemma.

Lemma 5. *Let $f_1, f_2, \ldots : \mathbb{N} \to \mathbb{N}$ be monotone nondecreasing functions satisfying $f_m(i) = o(i)$ for every fixed $m \in \mathbb{N}$. Then there exists a monotone nondecreasing function $s : \mathbb{N} \to \mathbb{N}$ such that $s(i) \to \infty$, $s(i) = o(i)$, and, for every fixed $m \in \mathbb{N}$, $s(i) \geq f_m(i)$ for all sufficiently large i.*

Proof. For each $k \in \mathbb{N}$, since $f_m(i) = o(i)$ for every $m \leq k$, we may choose a strictly increasing sequence of thresholds N_k such that $N_k \geq k^2$ and $f_m(i) \leq i/k$ for every $m \leq k$ and every $i \geq N_k$.

For $i < N_1$, define $s(i) := 1$. For $i \in [N_k, N_{k+1})$, define

$$s(i) := \max\{k, f_1(i), \ldots, f_k(i)\}.$$

Then s is monotone nondecreasing. Moreover, for $i \in [N_k, N_{k+1})$ we have $s(i) \geq k$, so $s(i) \to \infty$. Also $s(i) \leq \max\{k, i/k\}$, and since $i \geq N_k \geq k^2$, we have

$$\frac{s(i)}{i} \leq \max\left\{\frac{k}{i}, \frac{1}{k}\right\} \leq \frac{1}{k}.$$

Since $k \to \infty$, it follows that $s(i) = o(i)$. Finally, for every fixed m, whenever $i \geq N_m$, the interval containing i has index at least m, and hence $s(i) \geq f_m(i)$. □

Lemma 6. *Let (D, H) be a feasible profile. Then there exists a monotone nondecreasing super-linear function $D_{\mathrm{pfx}} : \mathbb{N} \to \mathbb{N}$ such that, defining*

$$r(i) := \max\big(\{0\} \cup \{j \in [i] : D(j) < D_{\mathrm{pfx}}(i)\}\big)$$

and

$$\tau(t) := \min\{n \in \mathbb{N} : D_{\mathrm{pfx}}(n) \geq t\} = D^{-1}_{\mathrm{pfx}}(t),$$

we have $r(i) = o(i)$ and $\tau(t)/t = o(H(t))$.

Proof. Since D is super-linear, for every fixed $m \in \mathbb{N}$, the function $i \mapsto D^{-1}(mi)$ is monotone nondecreasing and $o(i)$. Applying Lemma 5 to these functions gives a monotone nondecreasing function $g : \mathbb{N} \to \mathbb{N}$ such that $g(i) = o(i)$ and g eventually dominates each of them. Thus, for every fixed m, we have $D(g(i)) \geq mi$ for all sufficiently large i, and hence $D(g(i))/i \to \infty$.

Now, for each $m \in \mathbb{N}$, define

$$q_m(n) := 1 + \max\left\{D^{-1}(t) : t \in \mathbb{N},\ \left\lfloor \frac{tH(t)}{m} \right\rfloor \le n\right\},$$

with the convention that the maximum of the empty set is 0. Feasibility gives $D^{-1}(t) = o(tH(t))$ and $tH(t) \to \infty$. Consequently, the maximum defining $q_m(n)$ is over a finite set, and q_m is monotone nondecreasing.

We next verify that $q_m(n) = o(n)$ for every fixed m. Fix any $\varepsilon > 0$. For all sufficiently large t,

$$D^{-1}(t) \le \varepsilon \frac{tH(t)}{m}.$$

If t occurs in the maximum defining $q_m(n)$, then $tH(t)/m < n + 1$. Thus, these sufficiently large values of t contribute at most $\varepsilon(n+1)$, while the remaining finitely many values contribute a bounded quantity. Hence, for some finite constant $C_{m,\varepsilon}$ independent of n,

$$q_m(n) \le 1 + \max\{C_{m,\varepsilon}, \varepsilon(n+1)\}.$$

Dividing by n and using that $\varepsilon > 0$ was arbitrary gives $q_m(n) = o(n)$.

Now, apply Lemma 5 to the countable family $\{g, q_1, q_2, \ldots\}$. Thus, there exists a monotone nondecreasing function $s : \mathbb{N} \to \mathbb{N}$ such that $s(i) \to \infty$, $s(i) = o(i)$, and s eventually dominates g and every q_m. Now define

$$D_{\mathrm{pfx}}(i) := D(s(i)).$$

Since s and D are monotone nondecreasing, D_{pfx} is monotone nondecreasing. Moreover, since s eventually dominates g, we have

$$\frac{D_{\mathrm{pfx}}(i)}{i} = \frac{D(s(i))}{i} \ge \frac{D(g(i))}{i} \longrightarrow \infty,$$

so D_{pfx} is super-linear.

We next show that $r(i) = o(i)$. For any $j \in \mathbb{N}$, if $D(j) < D_{\mathrm{pfx}}(i) = D(s(i))$, then monotonicity of D gives $j < s(i)$. Therefore, $r(i) \le s(i) = o(i)$.

Finally, we prove $\tau(t)/t = o(H(t))$. Fix any $\alpha > 0$, and choose $m \in \mathbb{N}$ such that $1/m \le \alpha$. Since $tH(t) \to \infty$, the quantity $\lfloor \alpha t H(t) \rfloor$ tends to infinity. Moreover,

$$\left\lfloor \frac{tH(t)}{m} \right\rfloor \le \lfloor \alpha t H(t) \rfloor,$$

so t occurs in the maximum defining $q_m(\lfloor \alpha t H(t) \rfloor)$. Since s eventually dominates q_m, for all sufficiently large t,

$$s(\lfloor \alpha t H(t) \rfloor) \ge q_m(\lfloor \alpha t H(t) \rfloor) > D^{-1}(t).$$

Therefore,

$$D_{\mathrm{pfx}}(\lfloor \alpha t H(t) \rfloor) = D\big(s(\lfloor \alpha t H(t) \rfloor)\big) \ge t.$$

Hence $\tau(t) \le \lfloor \alpha t H(t) \rfloor$ for all sufficiently large t. Since $\alpha > 0$ was arbitrary, $\tau(t)/t = o(H(t))$. □

Lemma 7. *Let D_{pfx} be monotone nondecreasing and super-linear, and define $\tau(t) = D_{\mathrm{pfx}}^{-1}(t)$ as in Lemma 6. Suppose H is a monotone nonincreasing vanishing rate satisfying $\tau(t)/t = o(H(t))$. Then there exists a monotone nonincreasing vanishing rate H_{pfx} such that $H_{\mathrm{pfx}}(t) = \omega(D_{\mathrm{pfx}}^{-1}(t)/t)$ and $H_{\mathrm{pfx}}(t) \le H(t)$ for all sufficiently large t.*

Proof. Since $\tau(t)/t = o(H(t))$, $tH(t)/\tau(t) \to \infty$. Define

$$a(t) := \max\left\{1, \inf_{u \ge t} \frac{uH(u)}{\tau(u)}\right\}.$$

Then a is monotone nondecreasing and $a(t) \to \infty$.

Define $H_{\mathrm{pfx}}(t) := H(t)/\sqrt{a(t)}$. Since H is monotone nonincreasing and a is monotone nondecreasing, H_{pfx} is monotone nonincreasing. Moreover, $H_{\mathrm{pfx}}(t) \le H(t)$ for every t, and hence $H_{\mathrm{pfx}}(t) \to 0$.

For all sufficiently large t, the maximum with 1 in the definition of $a(t)$ is unnecessary, so $a(t) \leq tH(t)/\tau(t)$. Therefore,

$$\frac{H_{\mathrm{pfx}}(t)}{\tau(t)/t} = \frac{tH(t)}{\tau(t)\sqrt{a(t)}} \geq \sqrt{a(t)} \longrightarrow \infty.$$

Since $\tau(t) = D_{\mathrm{pfx}}^{-1}(t)$, this proves $H_{\mathrm{pfx}}(t) = \omega(D_{\mathrm{pfx}}^{-1}(t)/t)$. □

Finally, we provide the proof of Lemma 4.

Proof of Lemma 4. First apply Lemma 6 to obtain D_{pfx} such that $r(i) = o(i)$ and $\tau(t)/t = o(H(t))$. Then apply Lemma 7 to obtain a monotone nonincreasing vanishing hallucination rate H_{pfx} with $H_{\mathrm{pfx}}(t) = \omega(D_{\mathrm{pfx}}^{-1}(t)/t)$ and $H_{\mathrm{pfx}}(t) \leq H(t)$ for all sufficiently large t. Thus $(D_{\mathrm{pfx}}, H_{\mathrm{pfx}})$ is a feasible profile.

Let $\mathcal{A}$ be any generator satisfying the hypotheses, and fix any countable collection $\mathcal{L}$, target language $K \in \mathcal{L}$, and observation sequence $X \in \mathfrak{D}(K)$. By assumption, there exists a finite time T_{pfx} such that

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H_{\mathrm{pfx}}(t) \qquad \forall t \geq T_{\mathrm{pfx}}.$$

Choose a finite time T_0 such that $H_{\mathrm{pfx}}(t) \leq H(t)$ for every $t \geq T_0$. Setting $T := \max\{T_{\mathrm{pfx}}, T_0\}$, we immediately get

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H(t) \qquad \forall t \geq T.$$

It remains to prove the density guarantee. For any i, every element among the first i elements of K that is generated by time $D_{\mathrm{pfx}}(i)$ is generated by its element-wise deadline under D, except possibly those among the first $r(i)$ elements. Indeed, if $j \in [i]$ and $j > r(i)$, then $D(j) \geq D_{\mathrm{pfx}}(i)$ by definition of $r(i)$. Therefore, for every realization of the output sequence,

$$\mu_i(\mathcal{A}(X), K, D) \geq \mu_i^{\mathrm{pfx}}(\mathcal{A}(X), K, D_{\mathrm{pfx}}) - \frac{r(i)}{i}.$$

Taking lower limits, using $r(i) = o(i)$, and applying the assumed prefix-wise density guarantee for this fixed instance gives

$$\liminf_{i\to\infty} \mu_i(\mathcal{A}(X), K, D) \geq \liminf_{i\to\infty} \mu_i^{\mathrm{pfx}}(\mathcal{A}(X), K, D_{\mathrm{pfx}}) \geq \rho$$

almost surely. Since $(\mathcal{L}, K, X)$ was arbitrary, this establishes the required density guarantee almost surely for each fixed instance and completes the proof. □

D.2 Achieving lower density under the prefix metric

In this section we discuss an algorithm that reduces obtaining prefix-wise density guarantees to obtaining density guarantees in the limit. We first prove a convexification lemma that allows the algorithm to use a discrete-convex deadline without restricting the feasible profiles in our statement. We then provide the algorithm, restate the black-box reduction lemma, provide a couple of key ingredients necessary for the proof, and prove the result.

Lemma 8 (Rate-Preserving Convexification)**.** *Let (D, H) be a feasible profile. Then there exists a monotone nondecreasing, discrete-convex, super-linear function $\widetilde{D} : \mathbb{N} \to \mathbb{N}$ such that $\widetilde{D}(i) \leq D(i)$ for every $i \in \mathbb{N}$ and*

$$\widetilde{D}^{-1}(t) = o(tH(t)).$$

In particular, $(\widetilde{D}, H)$ is a feasible profile and $\widetilde{D}(m) - \widetilde{D}(m-1) \longrightarrow \infty$.

Proof. We start by defining

$$I(t) := D^{-1}(t) = \min\{n \in \mathbb{N} : D(n) \geq t\}.$$

Feasibility implies $I(t) = o(tH(t))$ and $tH(t) \to \infty$. Together with monotonicity of H, this also implies $H(t) > 0$ for every t. To account for plateaus and specify the extension to real arguments, define, for $u \geq 0$,

$$J(u) := \max\big(\{0\} \cup \{n \in \mathbb{N} : D(n) \leq u\}\big).$$

Since D is integer-valued and unbounded,

$$J(u) = I(\lfloor u \rfloor + 1) - 1.$$

Extend H to real arguments by $h(u) := H(\max\{1, \lfloor u \rfloor\})$. For $u \geq 1$, putting $n = \lfloor u \rfloor$, feasibility and the monotonicity of H give

$$\frac{J(u)}{uh(u)} \leq \frac{I(n+1)}{uH(n)} \leq 2\,\frac{I(n+1)}{(n+1)H(n+1)} \longrightarrow 0.$$

Thus $J(u) = o(uh(u))$.

Let C be the least concave majorant of J on $[0, \infty)$, given explicitly by

$$C(u) := \sup\left\{ \sum_{j=1}^{k} \lambda_j J(u_j) : \begin{array}{l} k \in \mathbb{N},\ \lambda_j \geq 0,\ u_j \geq 0, \\ \sum_{j=1}^{k} \lambda_j = 1, \quad \sum_{j=1}^{k} \lambda_j u_j = u \end{array} \right\}.$$

This function is finite. Indeed, superlinearity of D gives

$$M := \sup_{n \in \mathbb{N}} \frac{n}{D(n)} < \infty,$$

and hence $J(u) \leq Mu$ and $C(u) \leq Mu$. Combining admissible convex combinations shows that C is concave, and Jensen's inequality shows that it is the least concave majorant of J. Monotonicity of J also implies monotonicity of C: translating every u_j by the same nonnegative amount cannot decrease the corresponding value.

Moreover, $C(0) = 0$, and C is continuous on $[0, \infty)$, including at zero since $0 \leq C(u) \leq Mu$. Since $C \geq J$ is unbounded, concavity and monotonicity imply that C is strictly increasing.

We next bound C. For $u > 0$, define

$$\varepsilon(u) := \sup_{v \geq u} \frac{J(v)}{vh(v)},$$

so that $\varepsilon(u) \to 0$. Consider any convex combination in the definition of $C(u)$. The contribution from indices with $u_j \leq u$ is at most $J(u) \leq \varepsilon(u)uh(u)$. For the remaining indices, monotonicity of h gives

$$\sum_{u_j > u} \lambda_j J(u_j) \leq \varepsilon(u)h(u) \sum_{u_j > u} \lambda_j u_j \leq \varepsilon(u)uh(u).$$

Taking the supremum yields

$$C(u) \leq 2\varepsilon(u)uh(u) = o(uh(u)).$$

Let $G := C^{-1}$ be the ordinary continuous inverse. Since C is increasing and concave, G is increasing and convex, with $G(0) = 0$. Also,

$$C(D(n)) \geq J(D(n)) \geq n,$$

and therefore $G(n) \leq D(n)$ for every $n \in \mathbb{N}$. To make the interpolation explicit, let F be the piecewise-linear interpolation through $(0, 0)$ and the points $(n, D(n))$. The inequalities at integer arguments and convexity of G imply $G(u) \leq F(u)$ between consecutive integers as well. Since $h(u) \to 0$, we have $C(u) = o(u)$, and hence G is super-linear.

To obtain integer-valued deadlines, define

$$\ell_m := \lfloor G(m) - G(m-1) \rfloor, \qquad S(n) := \sum_{m=1}^{n} \ell_m, \qquad \widetilde{D}(n) := \max\{1, S(n)\}.$$

Convexity of G makes the nonnegative integers ℓ_m nondecreasing, so S is monotone nondecreasing and discrete convex. Taking the maximum with the constant 1 preserves these properties on positive indices: the piecewise-linear interpolation of S is convex and monotone nondecreasing, and so is its maximum with the constant 1. Restricting this function to positive integer arguments gives $\widetilde{D}$. Moreover,

$$G(n) - n \leq S(n) \leq G(n) \leq D(n).$$

Since $D(n) \geq 1$, it follows that $\widetilde{D}(n) \leq D(n)$. Superlinearity of G also implies superlinearity of S and $\widetilde{D}$.

Finally, superlinearity of G gives $S(n) \geq G(n)/2$ for all sufficiently large n. For sufficiently large integer t, set $n_t := \lceil C(2t) \rceil$. Then $n_t \to \infty$ and $G(n_t) \geq G(C(2t)) = 2t$, so

$$\widetilde{D}(n_t) \geq S(n_t) \geq \frac{G(n_t)}{2} \geq t.$$

Consequently,

$$\widetilde{D}^{-1}(t) \leq \lceil C(2t) \rceil \leq C(2t) + 1 = o(tH(t)).$$

Here we used $C(2t) = o(2tH(2t))$, $H(2t) \leq H(t)$, and $tH(t) \to \infty$. Thus $(\widetilde{D}, H)$ is feasible.

Since the ℓ_m are nondecreasing and $S(m)/m \to \infty$, we have $\ell_m \to \infty$. As $\widetilde{D} = S$ for all sufficiently large arguments, its successive differences also tend to infinity. □

Now, we present the algorithm that is capable of achieving timely density in the more general setting.

Algorithm 2 Speculative Blackbox Generation

Parameters: Feasible profile (D, H), blackbox $\mathcal{B}$, collection $\mathcal{L}$
Choose $d \leq D$ using Lemma 8; set $d(0) := 0$
Define $m(t) := \min\{m \in \mathbb{N} : t \leq d(m)\} = d^{-1}(t)$
Choose unbounded nondecreasing $C : \mathbb{N} \to \mathbb{N}$ such that
$\sum_{k \leq m(t)} C(k) = o(tH(t))$ and $C(m) = o(d(m) - d(m-1))$
for $m = 1, 2, \ldots$ **do**
  $\ell_m \leftarrow d(m) - d(m-1)$
  if $\ell_m = 0$ **then**
    $\delta_m \leftarrow 0$
    continue
  $\delta_m \leftarrow \min\{1, 10C(m)/\ell_m\}$
  for $t = d(m-1) + 1, \ldots, d(m)$ **do**
    Observe x_t
    $b_t \leftarrow \mathcal{B}(x_{1:t})$ // Advance the blackbox on every round
    Sample $\text{decision}_t \sim \text{Bern}(\delta_m)$ and $\text{choice}_t \sim \text{Unif}(\{1, \ldots, C(m)\})$
    if $\text{decision}_t = 0$ **then**
      $o_t \leftarrow b_t$
    else
      $o_t \leftarrow \text{NextUnused}_t(L^{\text{choice}_t})$
    Generate o_t

Lemma 9 (Black-Box Reduction for General Data Models)**.** *Fix any data model $\mathfrak{D}$, density $\rho \in [0, 1]$, and feasible profile (D, H). For any deterministic generator $\mathcal{B}$ that is eventually consistent under $\mathfrak{D}$ and satisfies $\mu_{low}(\mathcal{B}, \infty) = \rho$, there exists a randomized algorithm $\mathcal{A} = \mathcal{A}_{\mathcal{B}}$ achieving $\mu_{low}^{\text{pfx}}(\mathcal{A}, D) \geq \rho$ almost surely for each fixed instance and satisfying the following: for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $X \in \mathfrak{D}(K)$, there exists a finite time T such that*

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H(t) \qquad \forall t \geq T.$$

Furthermore, $\mathcal{A}$ only makes black-box calls to $\mathcal{B}$.

Proof of Lemma 9. Fix any countable collection $\mathcal{L}$, any $K \in \mathcal{L}$, and any admissible sequence $X \in \mathfrak{D}(K)$. We analyze the algorithm in Algorithm 2 on the input sequence X. We use the deadline $d := \widetilde{D} \leq D$ supplied by Lemma 8, and set $d(0) := 0$ for the epoch notation below. We know that (d, H) is feasible and $d(m) - d(m-1) \to \infty$.

We first justify the existence of the auxiliary function C. Define

$$a(m) := \max\{1, d(m) - d(m-1)\} \qquad \text{and} \qquad b(m) := \inf_{t \in \mathbb{N}:\, m(t) \geq m} \frac{tH(t)}{m}.$$

We have $a(m) \to \infty$. We also claim that $b(m) \to \infty$. Indeed, since $tH(t) = \omega(d^{-1}(t))$, for every $M > 0$ there exists T_M such that for all $t \geq T_M$, $tH(t) \geq Md^{-1}(t)$. Now, if $m(t) \geq m$, then $t > d(m-1)$ and $d^{-1}(t) = m(t) \geq m$. Therefore, for all sufficiently large m, $t \geq T_M$ and $tH(t)/m \geq M$ uniformly over all t satisfying $m(t) \geq m$. Taking the infimum over such t yields $b(m) \geq M$. Since $M > 0$ was arbitrary, $b(m) \to \infty$.

Applying Lemma 10, there exists a monotone nondecreasing function $C : \mathbb{N} \to \mathbb{N}$ such that $C(m) \to \infty$, $C(m) = o(a(m))$, and $C(m) = o(b(m))$. The first little-o condition implies $\delta_m \to 0$. Moreover, since C is monotone nondecreasing,

$$\sum_{k \leq m(t)} C(k) \leq m(t)C(m(t)).$$

By definition, $b(m(t)) \leq tH(t)/m(t)$. Thus, since $C(m) = o(b(m))$, $\sum_{k \leq m(t)} C(k) = o(tH(t))$.

Now, we prove the hallucination guarantee. Since $\mathcal{B}$ is deterministic and eventually consistent on X, it produces outputs outside K on only finitely many rounds; denote this finite quantity by

$$N_{\mathcal{B}}(X, K) := \sum_{s=1}^{\infty} \mathbf{1}\{b_s \notin K\}.$$

At each round s, a hallucination by $\mathcal{A}$ is either a retained blackbox hallucination or occurs on a speculative round. Recall that $m(t) = \min\{m \in \mathbb{N} : t \leq d(m)\}$ is the index of the epoch containing time t. Thus,

$$\begin{aligned} tH_{\mathcal{A}}^{\mathcal{L},K,X}(t) &= \mathbb{E}\left[\sum_{s=1}^{t} \mathbf{1}\{o_s \notin K\}\right] \\ &\leq N_{\mathcal{B}}(X, K) + \sum_{m \leq m(t)} \delta_m(d(m) - d(m-1)) \\ &\leq N_{\mathcal{B}}(X, K) + 10 \sum_{m \leq m(t)} C(m). \end{aligned}$$

By the choice of C, the second term is $o(tH(t))$ and therefore there exists some time t_1 such that for all $t \geq t_1$ the second term is at most $tH(t)/2$. Moreover, since $tH(t) = \omega(d^{-1}(t))$ and $d^{-1}(t) \to \infty$, we have $tH(t) \to \infty$. Since $N_{\mathcal{B}}(X, K) < \infty$, there exists some t_2 such that for all $t \geq t_2$ the first term is also at most $tH(t)/2$. Therefore, for any $t \geq \max\{t_1, t_2\}$, we have

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H(t).$$

This gives the required finite time $T = T(\mathcal{L}, K, X)$ for each fixed instance.

It remains to prove the density guarantee. We begin by bounding the probability that the generator makes fewer than i speculations from K by time $d(i)$. Fix $j \in \mathbb{N}$ such that $K = L^j$. Since $C(m) \to \infty$, $d(m) - d(m-1) \to \infty$, and $\delta_m \to 0$, there exists m_K such that for all $m \geq m_K$, we have $j \leq C(m)$, $d(m) - d(m-1) > 0$, and $\delta_m = 10C(m)/(d(m) - d(m-1))$. Moreover, since m_K is fixed, there exists $i_K \geq m_K$ such that for all $i \geq i_K$, $10(i - m_K + 1) \geq 5i$.

For each round t in epoch m, let Y_t be the indicator that $\text{decision}_t = 1$ and $\text{choice}_t = j$. Then for all $m \geq m_K$ and all t in epoch m,

$$\Pr(Y_t = 1) = \frac{\delta_m}{C(m)} = \frac{10}{d(m) - d(m-1)}.$$

Hence, by linearity of expectation and our choice of i_K, for all $i \geq i_K$,

$$\mathbb{E}\left[\sum_{t=1}^{d(i)} Y_t\right] \geq \sum_{m=m_K}^{i} \frac{10}{d(m) - d(m-1)} \cdot (d(m) - d(m-1)) = 10(i - m_K + 1) \geq 5i.$$

Define $S_i := \{\sum_{t=1}^{d(i)} Y_t \geq i\}$. Since the variables $\{Y_t\}_{t \leq d(i)}$ are independent, a Chernoff bound yields a constant $c > 0$ such that for all $i \geq i_K$,

$$\Pr(S_i^c) \leq e^{-ci}.$$

On S_i, every element of K_i has appeared either in the adversary's sequence or in the generator's output by time $d(i)$. Indeed, whenever an element of K_i remains unused by both players, a speculation from $L^j = K$ outputs such an element. Each of the at least i such speculations removes one remaining element unless none remain, while other observations and outputs cannot increase the number remaining. Therefore, on S_i, $K_i \subseteq \{x_1, \ldots, x_{d(i)}\} \cup \mathcal{A}(X)_{d(i)}$.

Now fix any $0 < \epsilon < 1$. For each i, let U_i be the set of elements of K_i eventually output by $\mathcal{B}$. Since $\mu_{low}(\mathcal{B}, \infty) = \rho$, there exists $i_1(\epsilon)$ such that for all $i \geq i_1(\epsilon)$, $|U_i| \geq (\rho - \epsilon/2)i$. For each $u \in U_i$, let $\tau(u)$ denote the first round at which $\mathcal{B}$ outputs u. Because $\mathcal{B}$ is deterministic and X is fixed, both U_i and these first occurrence times are deterministic. Partition U_i into

$$U_i^{\leq} := \{u \in U_i : \tau(u) \leq d(i)\} \qquad \text{and} \qquad U_i^{>} := U_i \setminus U_i^{\leq}.$$

If $u \in U_i^{>}$, then the adversary cannot have revealed u by time $d(i)$, since otherwise $\mathcal{B}$ could not validly output it at time $\tau(u) > d(i)$. Consequently, on S_i, every element of $U_i^{>}$ belongs to $\mathcal{A}(X)_{d(i)} \cap K_i$.

Since $\delta_m \to 0$, there exists $m(\epsilon)$ such that $\delta_m \leq \epsilon/2$ for all $m \geq m(\epsilon)$. Since $d(m(\epsilon) - 1)$ is finite, there exists $i_2(\epsilon)$ such that for all $i \geq i_2(\epsilon)$, $d(m(\epsilon) - 1) \leq \epsilon i/2$. Fix any deterministic $i \geq \max\{i_K, i_1(\epsilon), i_2(\epsilon)\}$. We will show that there exists a constant $c_\epsilon > 0$ such that

$$\Pr\left(\mu_i^{\text{pfx}}(\mathcal{A}(X), K, d) < \rho - 2\epsilon\right) \leq e^{-ci} + e^{-c_\epsilon i}.$$

It remains to account for the elements of $U_i^{\leq}$. Partition this set into

$$U_i^{Spec} := \{u \in U_i^{\leq} : u \in \mathcal{A}(X)_{\tau(u)-1}\} \qquad \text{and} \qquad U_i^{BB} := U_i^{\leq} \setminus U_i^{Spec}.$$

By definition, if $u \in U_i^{Spec}$, then u already appears among the generator's first $\tau(u) - 1$ outputs, and hence $u \in \mathcal{A}(X)_{d(i)} \cap K_i$. Let $T_i := \{\tau(u) : u \in U_i^{BB}\}$. Since distinct elements have distinct first occurrence times, we have $|T_i| = |U_i^{BB}|$. Moreover, if $t \in T_i$ and the generator retains the blackbox output at time t, then $o_t = u \in K_i$ and this element is not already in $\mathcal{A}(X)_{t-1}$ by definition of U_i^{BB}. Therefore, each such round contributes a distinct element of K_i to the generator's output.

We split T_i into early and late rounds in order to isolate the finite prefix of epochs where the speculation probabilities may still be large. Define

$$T_i^{late} := \{t \in T_i : m(t) \geq m(\epsilon)\}, \qquad T_i^{early} := T_i \setminus T_i^{late}.$$

Since T_i^{early} is contained in the first $m(\epsilon) - 1$ epochs, by our choice of $i_2(\epsilon)$ we have $|T_i^{early}| \leq d(m(\epsilon) - 1) \leq \epsilon i/2$.

Now, for each $t \leq d(i)$, define the indicator random variable $Z_t := \mathbf{1}\{t \in T_i^{late}\} \cdot \mathbf{1}\{\text{decision}_t = 0\}$, which is 1 exactly when a late blackbox output counted by T_i is retained by the generator. Since the contributions counted by $\sum_{t=1}^{d(i)} Z_t$ correspond to distinct elements of K_i and are disjoint from U_i^{Spec} and $U_i^{>}$, on S_i we have

$$|\mathcal{A}(X)_{d(i)} \cap K_i| \geq |U_i^{>}| + |U_i^{Spec}| + \sum_{t=1}^{d(i)} Z_t.$$

Let $\mathcal{F}_{t-1}$ denote the natural filtration containing the history up to immediately before the retention coin decision_t is flipped. In particular, the blackbox output at time t has already been determined at this point. Since $U_i^{\leq}$ and τ are deterministic, membership in T_i^{late} depends only on whether the corresponding element has appeared among the generator's outputs before time t. Thus, the event $\{t \in T_i^{late}\}$ is $\mathcal{F}_{t-1}$-measurable, while decision_t is sampled independently of $\mathcal{F}_{t-1}$. Therefore,

$$\mathbb{E}[Z_t \mid \mathcal{F}_{t-1}] = \mathbf{1}\{t \in T_i^{late}\}(1 - \delta_{m(t)}) \geq \left(1 - \frac{\epsilon}{2}\right)\mathbf{1}\{t \in T_i^{late}\},$$

where the inequality follows because $t \in T_i^{late}$ implies $m(t) \geq m(\epsilon)$.

Define $M_t := Z_t - \mathbb{E}[Z_t \mid \mathcal{F}_{t-1}]$. Then $\{M_t\}_{t=1}^{d(i)}$ is a martingale difference sequence with respect to $\{\mathcal{F}_t\}$, with $|M_t| \leq 1$ almost surely. Moreover,

$$\text{Var}(M_t \mid \mathcal{F}_{t-1}) = \text{Var}(Z_t \mid \mathcal{F}_{t-1}) \leq \mathbb{E}[Z_t \mid \mathcal{F}_{t-1}] \leq \mathbf{1}\{t \in T_i^{late}\}.$$

Thus the predictable quadratic variation satisfies

$$\sum_{t=1}^{d(i)} \mathrm{Var}(M_t \mid \mathcal{F}_{t-1}) \le |T_i^{late}| \le |T_i| = |U_i^{BB}| \le i.$$

Applying Freedman's inequality (Theorem 4) to the martingale difference sequence $\{-M_t\}_{t=1}^{d(i)}$ with $R = 1$, $\eta = \epsilon i/2$, and $\sigma^2 = i$, we obtain

$$\Pr\left(\sum_{t=1}^{d(i)} M_t \le -\frac{\epsilon}{2} i\right) \le \exp(-c_\epsilon i)$$

for some constant $c_\epsilon > 0$. On the complementary event,

$$\sum_{t=1}^{d(i)} Z_t \ge \sum_{t=1}^{d(i)} \mathbb{E}[Z_t \mid \mathcal{F}_{t-1}] - \frac{\epsilon}{2} i \ge \left(1 - \frac{\epsilon}{2}\right) |T_i^{late}| - \frac{\epsilon}{2} i.$$

Since $|T_i^{late}| = |U_i^{BB}| - |T_i^{early}| \ge |U_i^{BB}| - d(m(\epsilon) - 1)$, on the intersection of this event with S_i we obtain

$$\begin{aligned}
|\mathcal{A}(X)_{d(i)} \cap K_i| &\ge |U_i^{>}| + |U_i^{Spec}| + \left(1 - \frac{\epsilon}{2}\right)(|U_i^{BB}| - d(m(\epsilon) - 1)) - \frac{\epsilon}{2} i \\
&= |U_i| - \frac{\epsilon}{2}|U_i^{BB}| - \left(1 - \frac{\epsilon}{2}\right) d(m(\epsilon) - 1) - \frac{\epsilon}{2} i \\
&\ge |U_i| - \epsilon i - d(m(\epsilon) - 1) \\
&\ge \left(\rho - \frac{3\epsilon}{2}\right) i - d(m(\epsilon) - 1) \\
&\ge (\rho - 2\epsilon) i,
\end{aligned}$$

where the second inequality follows from $|U_i^{BB}| \le i$, the third follows from $|U_i| \ge (\rho - \epsilon/2)i$, and the final from $d(m(\epsilon) - 1) \le \epsilon i/2$. Combining this with the bound on $\Pr(S_i^c)$ proves that, for every deterministic $i \ge \max\{i_K, i_1(\epsilon), i_2(\epsilon)\}$,

$$\Pr\left(\mu_i^{\mathrm{pfx}}(\mathcal{A}(X), K, d) < \rho - 2\epsilon\right) \le e^{-ci} + e^{-c_\epsilon i}.$$

These probabilities are summable in i. Thus, by the first Borel–Cantelli lemma, with probability 1, for all sufficiently large i, $\mu_i^{\mathrm{pfx}}(\mathcal{A}(X), K, d) \ge \rho - 2\epsilon$. Applying this argument to $\epsilon = 1/n$ for every integer $n \ge 2$ and taking the intersection of the resulting probability-one events gives

$$\liminf_{i \to \infty} \mu_i^{\mathrm{pfx}}(\mathcal{A}(X), K, d) \ge \rho$$

almost surely. Finally, since $d(i) \le D(i)$ for every i,

$$\mu_i^{\mathrm{pfx}}(\mathcal{A}(X), K, D) \ge \mu_i^{\mathrm{pfx}}(\mathcal{A}(X), K, d).$$

Therefore, $\mathcal{A}$ achieves the required lower prefix-wise density under D almost surely for each fixed instance, together with the pointwise hallucination guarantee proved above. □

Lemma 10. *Let $a, b : \mathbb{N} \to \mathbb{R}_+$ satisfy $a(i) \to \infty$ and $b(i) \to \infty$. Then there exists a monotone nondecreasing function $s : \mathbb{N} \to \mathbb{N}$ such that $s(i) \to \infty$, $s(i) = o(a(i))$, and $s(i) = o(b(i))$.*

Proof. For each $k \in \mathbb{N}$, since $a(i) \to \infty$ and $b(i) \to \infty$, choose N_k sufficiently large so that $N_{k+1} > N_k$ and

$$a(i) \ge k^2 \qquad \text{and} \qquad b(i) \ge k^2$$

for every $i \ge N_k$. Define $s(i) := 1$ for $i < N_1$, and for $i \in [N_k, N_{k+1})$, define $s(i) := k$. Then s is monotone nondecreasing and $s(i) \to \infty$. Moreover, for $i \in [N_k, N_{k+1})$, we have $s(i) = k$, while $a(i) \ge k^2$ and $b(i) \ge k^2$. Hence

$$\frac{s(i)}{a(i)} \le \frac{1}{k} \qquad \text{and} \qquad \frac{s(i)}{b(i)} \le \frac{1}{k}.$$

Since $k \to \infty$, it follows that $s(i) = o(a(i))$ and $s(i) = o(b(i))$. □

Theorem 4 (Theorem 1.6 in [8]). *Let $\{M_t\}_{t=1}^T$ be a martingale difference sequence adapted to a filtration $\{\mathcal{F}_t\}_{t=0}^T$ such that $|M_t| \leq R$ almost surely for all t. Define*

$$W_k := \sum_{t=1}^{k} \mathrm{Var}(M_t \mid \mathcal{F}_{t-1}).$$

Then, for all $\eta, \sigma^2 > 0$,

$$\Pr\left(\exists k \leq T : \sum_{t=1}^{k} M_t \geq \eta \text{ and } W_k \leq \sigma^2\right) \leq \exp\left(-\frac{\eta^2/2}{\sigma^2 + R\eta/3}\right).$$

D.3 Proof of main theorem and extensions to other data models

In this section we provide examples of how our results in Lemma 4 and Lemma 9 allow us to generalize to other data models.

In fact, we will use this as an opportunity to formally prove Theorem 3 by proving a more general result for the partial enumeration model of [18]. Recall the partial enumeration model from Appendix D. In [18], they show the following:

Theorem 5 (Theorem 1.6 of [18]). *There exists a deterministic eventually consistent generator $\mathcal{B}_{\mathrm{KW}}$ such that for every $\alpha \in [0,1]$, under the data model $\mathfrak{D}_{\mathrm{part},\alpha}$, $\mu_{low}(\mathcal{B}_{\mathrm{KW}}, \infty) \geq \alpha/2$.*

We will now use this to get the following timely generation guarantee.

Theorem 6 (Timely Dense Generation from Partial Enumeration). *Fix any $\alpha \in [0,1]$ and any feasible profile (D, H). Under the data model $\mathfrak{D}_{\mathrm{part},\alpha}$, there exists a randomized generation algorithm $\mathcal{A} = \mathcal{A}_{\mathcal{B}_{\mathrm{KW}}}$ such that $\mu_{low}(\mathcal{A}, D) \geq \alpha/2$ almost surely for each fixed instance and, for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $X \in \mathfrak{D}_{\mathrm{part},\alpha}(K)$, there exists a finite time T such that*

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H(t) \qquad \forall t \geq T.$$

Proof. Fix $\alpha \in [0,1]$ and a feasible profile (D, H). By Theorem 5, there exists a deterministic eventually consistent generator $\mathcal{B}_{\mathrm{KW}}$ such that, under $\mathfrak{D}_{\mathrm{part},\alpha}$, $\mu_{low}(\mathcal{B}_{\mathrm{KW}}, \infty) \geq \alpha/2$. Let $\rho_\alpha := \mu_{low}(\mathcal{B}_{\mathrm{KW}}, \infty) \geq \alpha/2$.

Apply Lemma 4 with data model $\mathfrak{D}_{\mathrm{part},\alpha}$, density parameter ρ_α, and target feasible profile (D, H). This yields a feasible profile $(D_{\mathrm{pfx}}, H_{\mathrm{pfx}})$ such that any generator satisfying $\mu_{low}^{\mathrm{pfx}}(\cdot, D_{\mathrm{pfx}}) \geq \rho_\alpha$ almost surely for each fixed instance and hallucination rate eventually bounded by H_{pfx} on each fixed instance also satisfies $\mu_{low}(\cdot, D) \geq \rho_\alpha$ almost surely for each fixed instance and hallucination rate eventually bounded by H on each fixed instance.

Now apply Lemma 9 with data model $\mathfrak{D} = \mathfrak{D}_{\mathrm{part},\alpha}$, density parameter ρ_α, feasible profile $(D_{\mathrm{pfx}}, H_{\mathrm{pfx}})$, and black-box generator $\mathcal{B}_{\mathrm{KW}}$. We obtain a randomized algorithm $\mathcal{A} = \mathcal{A}_{\mathcal{B}_{\mathrm{KW}}}$ such that $\mu_{low}^{\mathrm{pfx}}(\mathcal{A}, D_{\mathrm{pfx}}) \geq \rho_\alpha$ almost surely for each fixed instance and, for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $X \in \mathfrak{D}_{\mathrm{part},\alpha}(K)$, there exists a finite time $T_{\mathrm{pfx}} = T_{\mathrm{pfx}}(\mathcal{L}, K, X)$ such that

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H_{\mathrm{pfx}}(t) \qquad \forall t \geq T_{\mathrm{pfx}}.$$

Therefore, by Lemma 4, $\mu_{low}(\mathcal{A}, D) \geq \rho_\alpha \geq \alpha/2$ almost surely for each fixed instance and, for every such instance, there exists a finite time $T = T(\mathcal{L}, K, X)$ such that

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H(t) \qquad \forall t \geq T.$$

This concludes the proof. □

We can now prove the main result in the main body of our paper.

Proof of Theorem 3. Every total enumeration of K belongs to $\mathfrak{D}_{\mathrm{part},1}(K)$. Therefore, applying Theorem 6 with $\alpha = 1$ gives the result. □

Another example of when we can directly apply our reduction is the contamination framework of [20]. We will provide an explicit example of the finite contamination data model; however, the reduction also applies to other contamination models whenever a deterministic eventually consistent generator satisfies the lower-density hypothesis of Lemma 9. In particular, any such guarantee of lower density ρ in the limit can be converted into a timely lower-density guarantee of at least ρ under any feasible profile (D, H).

The work in [20] combined with the work in [18] implies the following guarantee.

Theorem 7 (Theorem 6.18 of [20] + Theorem 1.6 of [18])**.** *There exists a deterministic eventually consistent generator $\mathcal{B}_{\mathrm{MVYZ}}$ such that, under the data model $\mathfrak{D}_{\text{fin-cont}}$, $\mu_{low}(\mathcal{B}_{\mathrm{MVYZ}}, \infty) \geq 1/2$.*

With this, we get the following timely generation guarantee.

Theorem 8 (Timely Dense Generation from Finite Contamination)**.** *Fix any feasible profile (D, H). Under the data model $\mathfrak{D}_{\text{fin-cont}}$, there exists a randomized generation algorithm $\mathcal{A} = \mathcal{A}_{\mathcal{B}_{\mathrm{MVYZ}}}$ such that $\mu_{low}(\mathcal{A}, D) \geq 1/2$ almost surely for each fixed instance and, for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $X \in \mathfrak{D}_{\text{fin-cont}}(K)$, there exists a finite time T such that*

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H(t) \qquad \forall t \geq T.$$

Proof. Fix a feasible profile (D, H). By Theorem 7, there exists a deterministic eventually consistent generator $\mathcal{B}_{\mathrm{MVYZ}}$ such that, under $\mathfrak{D}_{\text{fin-cont}}$, we have $\mu_{low}(\mathcal{B}_{\mathrm{MVYZ}}, \infty) \geq 1/2$. Let $\rho := \mu_{low}(\mathcal{B}_{\mathrm{MVYZ}}, \infty) \geq 1/2$.

Apply Lemma 4 with data model $\mathfrak{D}_{\text{fin-cont}}$, density parameter ρ, and target feasible profile (D, H). This yields a feasible profile $(D_{\mathrm{pfx}}, H_{\mathrm{pfx}})$ such that any generator satisfying $\mu_{low}^{\mathrm{pfx}}(\cdot, D_{\mathrm{pfx}}) \geq \rho$ almost surely for each fixed instance and hallucination rate eventually bounded by H_{pfx} on each fixed instance also satisfies $\mu_{low}(\cdot, D) \geq \rho$ almost surely for each fixed instance and hallucination rate eventually bounded by H on each fixed instance.

Now apply Lemma 9 with data model $\mathfrak{D} = \mathfrak{D}_{\text{fin-cont}}$, density parameter ρ, feasible profile $(D_{\mathrm{pfx}}, H_{\mathrm{pfx}})$, and black-box generator $\mathcal{B}_{\mathrm{MVYZ}}$. We obtain a randomized algorithm $\mathcal{A} = \mathcal{A}_{\mathcal{B}_{\mathrm{MVYZ}}}$ such that $\mu_{low}^{\mathrm{pfx}}(\mathcal{A}, D_{\mathrm{pfx}}) \geq \rho$ almost surely for each fixed instance and, for every countable collection $\mathcal{L}$, every $K \in \mathcal{L}$, and every $X \in \mathfrak{D}_{\text{fin-cont}}(K)$, there exists a finite time $T_{\mathrm{pfx}} = T_{\mathrm{pfx}}(\mathcal{L}, K, X)$ such that

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H_{\mathrm{pfx}}(t) \qquad \forall t \geq T_{\mathrm{pfx}}.$$

Therefore, by Lemma 4, we conclude that $\mu_{low}(\mathcal{A}, D) \geq \rho \geq 1/2$ almost surely for each fixed instance and, for every such instance, there exists a finite time $T = T(\mathcal{L}, K, X)$ such that

$$H_{\mathcal{A}}^{\mathcal{L},K,X}(t) \leq H(t) \qquad \forall t \geq T,$$

concluding the proof. □

Our results extend to additional data models admitting suitable deterministic eventually consistent blackbox generators, and we defer exploration of these extensions to future work.

E Optimal linear-time upper density via eventually consistent generators

While the impossibility results of Section 2 establish that there exist simple language families $\mathcal{L}_{chain}$ for which no consistent generator can achieve density across all finite prefixes of every target language $K \in \mathcal{L}_{chain}$, this limitation can be circumvented under a weaker notion that requires density only infinitely often at isolated indices. We formally define this notion, first introduced by [17], below.

Definition 5 (Upper Density)**.** *The upper density of a generation algorithm $\mathcal{A}$ is defined as*

$$\mu_{up}(\mathcal{A}, D) = \inf \left\{ \inf_{K \in \mathcal{L}} \inf_{E \in \mathcal{E}(K)} \limsup_{i \to \infty} \mu_i(\mathcal{A}(E), K, D) : \mathcal{L} \text{ is countable} \right\}.$$

Under this notion, it suffices for the generator to achieve high density along an infinite subsequence of indices, rather than maintaining density uniformly over all sufficiently large prefixes. Interestingly this relaxation suffices for eventually-consistent generators to obtain the optimal density in timely fashion.

Algorithm 3 Generation via Consistent Guessing (GCG)

Initialize $m \leftarrow 1$, $G \leftarrow [\,]$, and $\hat{L}^0 \leftarrow L^1$ // $G = (g_1, \dots, g_{|G|})$; g_m is its m-th element
Define $\alpha_m \leftarrow 1/2 - 2^{-m}$ for all $m \in \mathbb{N}$, and $D(i) = i$
for $t = 1, 2, \dots$ **do**
    $\hat{L}^t \leftarrow \mathsf{Accurate}(e_{1:t})$
    if $\hat{L}^t \supsetneq \hat{L}^{t-1}$ **then**
        Append $\hat{L}^t$ to G
    if $m \leq |G|$ **then**
        Output $\mathsf{OnTimeUnused}(g_m)$ // first element of g_m not yet late and not output/observed
    else
        Output $\mathsf{OnTimeUnused}(\hat{L}^t)$ // first element of $\hat{L}_t$ not yet late and not output/observed
    if $|G| \geq m$ and $\mu_t(\mathsf{GCG}(E), g_m, D) \geq \alpha_m$ **then**
        $m \leftarrow m + 1$

Theorem 9 (Separation of Upper and Lower Densities)**.** *Let $D(i) = i$ be the identity function. There exists a deterministic eventually consistent generator $\mathcal{A}$ such that $\mu_{up}(\mathcal{A}, D) = 1/2$.*

This result is strong in that the smallest deadline one can hope to obtain optimal density within is the identity function. This result also implies the first quantitative separation between the upper density and lower density metrics.

We begin with a result from [17]. They show that for any collection of languages $\mathcal{L}$, there exists a simple, deterministic algorithm $\mathsf{Accurate} = \{\mathsf{Accurate}_t : U^t \to \mathcal{L}\}_{t=1}^{\infty}$ with the following property. For any target language $K \in \mathcal{L}$ and any total enumeration $E = (e_1, e_2, \dots) \in \mathcal{E}(K)$ of K, if we denote the output at time t by $\hat{L}^t = \mathsf{Accurate}(e_{1:t})$, then:

1. $\hat{L}^t = K$ for infinitely many t, and
2. there exists a finite time T such that for all $t \geq T$, $\hat{L}^t \subseteq K$.

That is, the algorithm eventually only produces guesses that are consistent with the target language, and moreover, it guesses the target language correctly infinitely often.

Using this procedure, Kleinberg and Wei [17] construct a generation algorithm $\mathcal{A}$ satisfying $\mu_{up}(\mathcal{A}, \infty) = 1/2$, which can easily be shown to be optimal. We now show that a simple modification of this construction yields a stronger result: one can obtain an algorithm GCG, given in Algorithm 3, such that $\mu_{up}(\mathsf{GCG}, D) = \frac{1}{2}$ even under the identity deadline function $D(i) = i$.

The generator maintains an ordered list G of candidate languages produced over time. Whenever $\mathsf{Accurate}$ outputs a guess that is a strict superset of its previous guess, this guess is appended to G. Intuitively, these strict superset transitions correspond to moments where the learner expands its current hypothesis and therefore may expose new elements worth generating from. Thus, at any time we may write $G = (g_1, \dots, g_{|G|})$, where g_m denotes the m-th element whenever $m \leq |G|$.

In addition to G, the generator maintains a stage index $m \in \mathbb{N}$ that determines from which candidate language it generates. When $m \leq |G|$, the generator generates from g_m. When $m > |G|$, it instead generates directly from the current guess $\hat{L}^t$ produced by $\mathsf{Accurate}$.

For any language $L \subseteq U$, let $\mathsf{OnTimeUnused}(L)$ denote the first element of L (under the canonical ordering of U) that is not yet late (according to the deadline function) and has neither previously been output by the generator nor previously been observed from the adversary. At each round t, the generator outputs $\mathsf{OnTimeUnused}(g_m)$ when $m \leq |G|$, and $\mathsf{OnTimeUnused}(\hat{L}^t)$ otherwise.

The progression of the stage index m is governed by a sequence of thresholds $\{\alpha_m\}_{m \in \mathbb{N}}$ defined by $\alpha_m = 1/2 - 2^{-m}$. Whenever $m \leq |G|$, the generator monitors its empirical density with respect to the current candidate g_m. If at time t the condition $\mu_t(\mathsf{GCG}(E), g_m, D) \geq \alpha_m$ is satisfied, then the generator increments $m \leftarrow m + 1$, retiring the current candidate even if no successor has yet been appended. In this way, the generator proceeds through the list G, spending sufficient time generating from each candidate before advancing, and falls back to the current guess whenever the list is exhausted.

Proof. We recall the guarantees of the algorithm Accurate above. For any target language $K \in \mathcal{L}$ and any total enumeration $E \in \mathcal{E}(K)$, the sequence of guesses $\{\hat{L}^t\}_{t\in\mathbb{N}}$ satisfies that

1. $\hat{L}^t = K$ for infinitely many t, and
2. there exists a finite time T such that for all $t \geq T$, $\hat{L}^t \subseteq K$.

The algorithm GCG maintains two objects: a list $G = (g_1, \ldots, g_{|G|})$ of candidate languages which is initialized to be empty and a stage index $m \in \mathbb{N}$ which is initialized to be 1. The list G is constructed from the guesses output by Accurate as follows: whenever at some round t we have $\hat{L}^t \supsetneq \hat{L}^{t-1}$, the guess $\hat{L}^t$ is appended to G. Thus, G consists precisely of those guesses produced by Accurate that are strict supersets of its previous guess.

At each round, if $m > |G|$, the generator outputs the first element of $\hat{L}^t$ that is both not yet late and has not previously been output by the generator or observed from the adversary. Otherwise, it outputs the first element of g_m with this same property. Whenever $m \leq |G|$, the stage index m is incremented once the generator has achieved the density threshold $\mu_t(\mathsf{GCG}(E), g_m, D) \geq \alpha_m$, where $\alpha_m = \frac{1}{2} - 2^{-m}$.

We first show that every language appended to G is eventually retired. Fix any m and suppose the algorithm begins generating from g_m at some time $T_m + 1$. Suppose for the sake of contradiction that g_m is never retired. Then for all $t > T_m$, round t would be in stage m, and the generator would output the earliest element of g_m that is both not yet late and has not previously been generated by either player. We now argue that the generator achieves $\mu_t(\mathsf{GCG}(E), g_m, D) \geq \alpha_m$ after finitely many rounds.

We first observe that no element outside the first T_m elements of g_m becomes late before being generated by one of the two players. Indeed, fix any $j > T_m$ and consider the j-th element of g_m. If this element has not been generated by either player before round j and is not observed at round j, then it remains available to the generator. At this round, every preceding element of g_m is already late, while the j-th element is still on time. Thus, it is the earliest available element that is not yet late, and the generator outputs it. Consequently, every such element is generated by one of the two players no later than its deadline.

Now let i_n denote the index of the element of g_m output by the generator at round $T_m + n$. Since this element is generated on time, we have $i_n \geq T_m + n$. Consequently, by time $T_m + n$, every element among the first i_n elements of g_m either belongs to the first T_m elements or has already been generated by one of the two players. Since each player has made at most $T_m + n$ outputs, it follows that

$$i_n \leq T_m + (T_m + n) + (T_m + n) = 3T_m + 2n.$$

Moreover, the n outputs generated at rounds $T_m + 1, \ldots, T_m + n$ are distinct elements of $(g_m)_{i_n}$ and have all been generated on time. Since $i_n \geq T_m + n$, all of these outputs have already been generated by time i_n. It follows that

$$\mu_{i_n}(\mathsf{GCG}(E), g_m, D) \geq \frac{n}{i_n} \geq \frac{n}{3T_m + 2n}.$$

Since $\alpha_m < \frac{1}{2}$ is fixed and $n/(3T_m + 2n)$ converges to $\frac{1}{2}$, there exists some finite n for which the density threshold is reached at time i_n, a contradiction. Therefore, every language, once reached, is retired in finite time. Since the generator proceeds through G in order, it follows by induction on m that every language appended to G is eventually reached and retired.

The same argument shows that whenever, from some finite time onward, the generator outputs the earliest element of a fixed language L that is both not yet late and has not previously been generated by either player, we have

$$\limsup_{i\to\infty} \mu_i(\mathsf{GCG}(E), L, D) \geq \tfrac{1}{2}. \tag{1}$$

We now analyze the behavior of the algorithm via three cases.

G remains empty: In this case, there never exists a time t for which $\hat{L}^t \supsetneq \hat{L}^{t-1}$, as otherwise G would be nonempty. Since there are infinitely many rounds t for which $\hat{L}^t = K$, and since T is finite, there must also be infinitely many rounds after time T for which this is true. Furthermore, since for

all $t \geq T$ we have $\hat{L}^t \subseteq K$, it cannot be the case that there exists some $t \geq T$ for which $\hat{L}^t \subsetneq K$, as otherwise there would exist a first later round $t' > t$ for which $\hat{L}^{t'} = K$, implying $\hat{L}^{t'} \supsetneq \hat{L}^{t'-1}$, a contradiction to the fact that G remains empty. As such, we must have $\hat{L}^t = K$ for all $t \geq T$.

It follows that for every round $t \geq T$, the generator outputs the earliest element of K that is both not yet late and has not previously been output by the generator or observed from the adversary. Therefore, by (1),

$$\limsup_{i \to \infty} \mu_i(\mathsf{GCG}(E), K, D) \geq \tfrac{1}{2}.$$

G grows indefinitely: In this case, $\hat{L}^t$ does not converge to a fixed language, and the list G grows without bound. We first claim that K appears infinitely often in G. Indeed, since there are infinitely many rounds t for which $\hat{L}^t = K$ and for all $t \geq T$ we have $\hat{L}^t \subseteq K$, it follows that whenever the sequence transitions from a strict subset of K to K, we must have $\hat{L}^t \supsetneq \hat{L}^{t-1}$, and hence K is appended to G. Since $\hat{L}^t$ does not converge to a fixed language, these transitions occur infinitely often, and so K appears infinitely often in G.

Since every language appended to G is retired in finite time and K appears infinitely often in G, there exists an infinite sequence of stages $\{m_j\}_{j \in \mathbb{N}}$ such that $g_{m_j} = K$. At each such stage m_j, the generator achieves density at least α_{m_j} with respect to K. Finally, since α_{m_j} goes to $\frac{1}{2}$ as j goes to infinity, and the corresponding retirement times tend to infinity, it follows that

$$\limsup_{i \to \infty} \mu_i(\mathsf{GCG}(E), K, D) \geq \tfrac{1}{2}.$$

G is nonempty and stops growing after some finite time: Suppose G is nonempty and there exists some $m^\star$ such that $g_{m^\star}$ is the last element ever appended to G. Let $T^\star$ denote the time at which $g_{m^\star}$ is appended to G. Since there are infinitely many rounds t for which $\hat{L}^t = K$, and since $T^\star$ is finite, there must also be infinitely many rounds after time $T^\star$ for which this is true. Furthermore, since for all $t \geq T$ we have $\hat{L}^t \subseteq K$, it cannot be the case that there exists some $t > \max\{T, T^\star\}$ for which $\hat{L}^t \subsetneq K$, as otherwise there would exist a first later round $t' > t$ for which $\hat{L}^{t'} = K$, implying $\hat{L}^{t'} \supsetneq \hat{L}^{t'-1}$ and hence that some additional language is appended to G, a contradiction. As such, we must have $\hat{L}^t = K$ for all sufficiently large t.

By the argument above, every language appended to G is retired in finite time. In particular, $g_1, \ldots, g_{m^\star}$ are all retired in finite time, and so the generator eventually has $m = m^\star + 1 > |G|$ and generates directly from the current guess $\hat{L}^t$. Thus, from some finite time onward, the generator outputs the earliest element of K that is both not yet late and has not previously been generated by either player. Exactly as in the case where G remains empty, (1) implies that

$$\limsup_{i \to \infty} \mu_i(\mathsf{GCG}(E), K, D) \geq \tfrac{1}{2}.$$

As we have now argued that $\limsup_{i \to \infty} \mu_i(\mathsf{GCG}(E), K, D) \geq \frac{1}{2}$ in each case, and since the target language and its total enumeration were arbitrary, we have $\mu_{up}(\mathsf{GCG}, D) \geq \frac{1}{2}$. Together with the upper bound of $\frac{1}{2}$ noted above, this proves the density guarantee.

It remains to show that the generator is eventually consistent. Since T is finite, only finitely many languages are appended to G before time T. By the argument above, each of these languages is retired in finite time. Furthermore, every language appended to G at or after time T is a subset of K. Thus, after time T and after all languages appended before time T have been retired, the generator either generates from some $g_m \subseteq K$ or directly from the current guess $\hat{L}^t \subseteq K$. Therefore, every subsequent output belongs to K, completing the proof. $\square$

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We prove everything we state as a result in our abstract and additionally prove several additional results as well.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the final section we discuss the following limitation of our work: an ideal measure of timeliness measures the asymptotic time required to guarantee dense generation from our algorithm with respect to a preference ordering on both strings and languages. This would be ideal and leads to our Golden-Ratio Barrier for Timely Language Generation conjecture.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We provide a proof of every original result we present and cite the results we use that were a product of other research efforts.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [N/A]

Justification: This is a theory paper and therefore is devoid of experiments.

Guidelines:

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

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

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

   Answer: [N/A]

   Justification: This is a theory paper and therefore is devoid of data and code.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [N/A]

   Justification: This is a theory paper and therefore is devoid of experiments.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [N/A]

   Justification: This is a theory paper and therefore is devoid of experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [N/A]

   Justification: This is a theory paper and therefore is devoid of experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [N/A]

   Justification: This is a theory paper and therefore is devoid any ethical concerns and will almost certainly have no societal impact as defined in the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This is a theory paper and therefore will almost certainly have no broader societal impact as defined in the NeurIPS Code of Ethics.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This is a theory paper and therefore is devoid any data and code.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [N/A]

    Justification: This is a theory paper and therefore is devoid any assets.

    Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: This is a theory paper and therefore is devoid any assets.

    Guidelines:

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

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

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

    Answer: [N/A].

    Justification: This is a theory paper. There was no crowdsourcing conducted.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This is a theory paper; we did not have study participants.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A].

Justification: This is a theory paper; we have no need for LLMs in experiments because there are no experiments. LLM useage was for the purpose of editing/writing only.

Guidelines:

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