# DETERMINED BUT UNREACHABLE: CRYPTOGRAPHIC BARRIER TO LANGUAGE GENERATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Can examples determine a language while leaving the production of even one new word out of reach? We show that this happens exactly when one-way functions exist. With one-way functions, attaching to every word a statistically binding but computationally hiding commitment to the key of a pseudorandom-function graph gives an efficiently presented family in which one example determines the language except with probability $2^{-\Omega(n)}$, yet every efficient generator has negligible probability of producing an unseen valid word within any polynomial horizon. With exponentially secure primitives the barrier extends to total running time $2^{\Omega(n)}$, which is tight up to the constant in the exponent. For the converse we view a family through its sampling map, which sends a hidden description and coins to the examples: the posterior is the projection of the uniform measure on a fibre, and a point of the fibre whose language lies inside the target enumerates fresh words forever. Classical universal posterior sampling in the absence of one-way functions, stated jointly with the hidden target and padded across lengths, then yields efficient generators that match every generator allowed to draw posterior samples, at infinitely many lengths. Hence one-way functions exist if and only if posterior sampling helps generation, and hard instances for generation and for identification exist under the same condition. Since the Bayesian generator idealizes in-context learning as implicit posterior inference, the result also shows that this idealization can be computationally out of reach even when a single example suffices statistically.

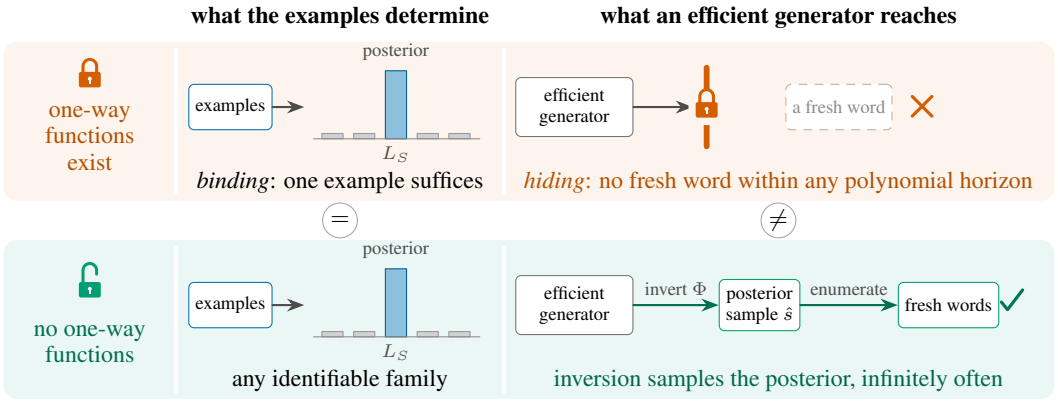



Figure 1: Determined but unreachable. In both rows the examples determine the target language; only computation differs. If one-way functions exist (top), attaching to every word a statistically binding, computationally hiding commitment to the key makes one example fix the language except with probability $2^{-\lfloor n/5 \rfloor}$, yet every efficient generator has negligible probability of producing a fresh word within any polynomial horizon (Theorem 5.5). If they do not (bottom), inverting the sampling map samples the posterior at infinitely many lengths, so efficient generators match posterior-sampling ones up to a polynomial factor at those lengths (Theorem 4.2). The two rows are the two sides of Theorem 6.1.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

# 1 INTRODUCTION

A generator sees examples from an unknown language and must produce a valid word absent from the data. Without computational limits, Kleinberg & Mullainathan (2024) showed that generation in the limit succeeds for every countable collection of languages, even when identification is impossible; later work studied closure dimension (Raman et al., 2025), cumulative mistakes (Kleinberg et al., 2026), and space constraints (Flammarion et al., 2026). We study the opposite regime, in which polynomially many examples from a known prior and source already identify the language, and ask whether computation alone can block generation. It can, exactly when one-way functions exist: then one example can determine a language that no efficient algorithm can extend by a single word within any polynomial horizon (Figure 1). The barrier is average-case and concerns time, since the same language is generated efficiently in the limit.

**Theorem 1.1** (Cryptographic barrier, informal)**.** *Uniform one-way functions exist if and only if there is a family of languages, with efficient membership, enumeration and sampling, in which*

*(i) polynomially many examples determine the target language with high probability, yet*

*(ii) every efficient generator has negligible probability of producing even one valid word it has not seen, within any polynomial number of rounds.*

*When one-way functions exist, the family can be chosen so that a single example determines the language except with probability $2^{-\Omega(n)}$. The equivalence persists if (ii) only requires every efficient generator to fail with constant probability, or if identification replaces generation in (ii). With (ii) unchanged, it also persists if (i) is relaxed to the success, with inverse-polynomial probability, of an efficient generator that may draw polynomially many samples from the posterior given its first examples. Thus one-way functions exist exactly when posterior sampling helps generation.*

**The relative point of view.** Why should the answer be one-wayness? Treat a family of languages as a single object over its space of descriptions. A hidden description $s$ and random coins $r$ produce a transcript of examples $z = \Phi(s, r)$ through a public, efficient *sampling map* $\Phi$. The descriptions that could have produced $z$, weighted by their likelihoods, form exactly the projection to descriptions of the uniform measure on the fibre $\Phi^{-1}(z)$ (Lemma 2.2). Whether the examples determine the language depends on the *shape* of fibres: over a typical transcript, the fibre should lie over a single language (Figure 2). Whether a generator can use this information depends on *access* to fibres: a generator holding a point $(\hat{s}, \hat{r})$ of the fibre, that is, an explanation of the examples by a description and coins, whose language lies inside the target, can enumerate $L_{\hat{s}}$ and produce fresh valid words forever. Maps whose fibres cannot be reached efficiently are one-way functions. The question thus becomes whether a fibre can lie over a single language and still be out of reach, and a statistically binding but computationally hiding commitment is exactly such a fibre.

**How the proof goes.** The converse combines two classical facts, that a uniform preimage of a sampler with its coins is a posterior draw (Naor & Rothblum, 2006, proof of Theorem 4.2) and that without one-way functions samplable models admit efficient approximate conditional sampling, known as universal extrapolation (Impagliazzo & Levin, 1990; Ostrovsky & Wigderson, 1993; Hirahara & Nanashima, 2023), with a simulation theorem that replaces the posterior draws of a running generator by inversions of the sampling map, jointly with the hidden target (Theorem 3.1), and a padding lemma that moves inversion to infinitely many description lengths (Lemma 4.1). The direct part attaches to every word a statistically binding but computationally hiding commitment to a key, together with a rare word that reveals the key. This makes any keyed family posterior-identifiable while leaving generation as hard as producing a fresh word of the untagged family or recovering its key (Lemmas 5.3 and 5.4), and a pseudorandom-function graph gives the witness (Theorem 5.5).

**What is new.**

- The equivalence itself, with ordinary one-way functions, a witness identifiable from one example, a single-fresh-word criterion, and both strong and weak gaps (Theorem 6.1). In sample-complexity terms, one example suffices information-theoretically while every efficient generator needs superpolynomially many to succeed with non-negligible probability.

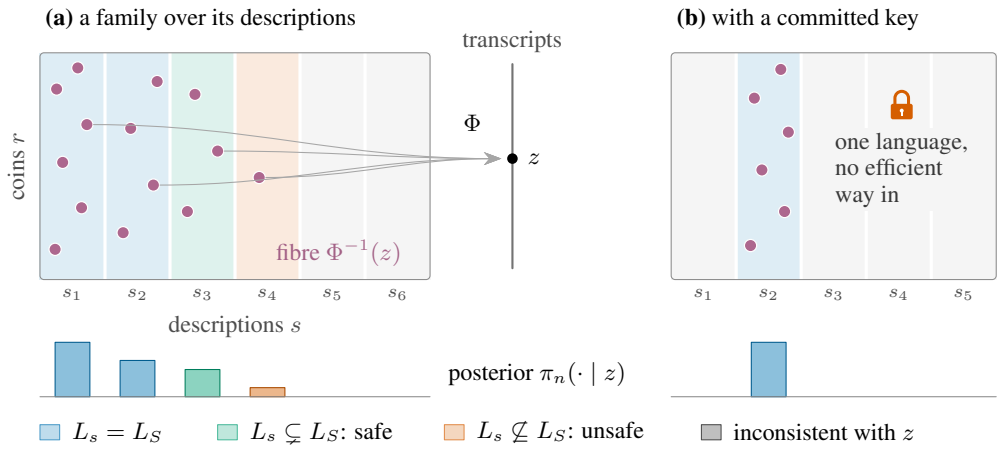



Figure 2: The relative picture. (a) A hidden description and random coins produce a transcript through the public sampling map Φ. Colours show how each description's language relates to the target $L_S$. The fibre over a transcript $z$ (dots), projected to descriptions, is exactly the posterior (bars; Lemma 2.2). The examples determine the language when the fibre lies over a single language, and a generator that holds any point of the fibre over a blue or green column enumerates fresh valid words forever. (b) When every word carries a statistically binding, computationally hiding commitment to the key (Section 5), one example places the whole fibre over the target's language except with probability $2^{-\lfloor n/5 \rfloor}$, yet no efficient algorithm reaches it (Theorem 5.5).

- A converse for every generator that may draw posterior samples and every success criterion (Theorems 3.1 and 4.2), which shows that a strong gap in $\mathcal{K}_{\text{post}}$ makes the family's own padded sampling map distributionally one-way (Remark 4.4).
- A tagging construction that makes any keyed family identifiable without making it easier to generate, a hardness proof by hidden checks whose concrete-security form is tight up to the constant in the exponent (Lemmas 5.3 and 5.4 and Proposition E.1), and the same threshold for identification (Corollary 6.3).

Proving such a barrier unconditionally would prove that one-way functions exist, and hence $\mathrm{P} \neq \mathrm{NP}$; in the terminology of Impagliazzo (1995), the threshold is the border between Pessiland and Minicrypt.

**Related work.** Learning to generate is classically hard under cryptographic assumptions (Kearns et al., 1994; Naor, 1996; Xiao, 2010), and Naor & Rothblum (2006) relate learning to impersonate a secret process to *almost* one-way functions, hard at infinitely many lengths, a quantifier arrangement dual to ours, so neither equivalence implies the other. These works measure closeness in distribution, allow repeated outputs, or lack statistical identifiability; our witness is identifiable from one example, success is a single fresh word, and a matching converse holds under ordinary one-way functions. That inference in samplable models reduces to inversion is classical (Impagliazzo & Luby, 1989; Impagliazzo & Levin, 1990; Ostrovsky & Wigderson, 1993; Hirahara & Nanashima, 2023); in Bayesian language, universal extrapolation is posterior sampling. Goldreich (1990) characterizes one-way functions by samplable ensembles that are statistically far but computationally close, a template our equivalence follows, and Voitovych et al. (2026) show that conditional-sampling oracles help online learning. One-way functions also separate computational from statistical sample complexity in PAC learning (Decatur et al., 1999) and make posterior sampling for diffusion models intractable (Gupta et al., 2024). For generation in the limit, Arenas et al. (2026) show prohibitive sample requirements for regular and context-free languages, Jimenez et al. (2026) give a strong-RSA barrier on finite domains and, under injective one-way functions, a family generatable but not learnable in polynomial time, Li et al. (2026) give a random-oracle mistake separation, and Kalavasis et al. (2025); Anastasopoulos et al. (2026) compare generation with identification. Our equivalence holds in the standard model, with an average-case i.i.d. experiment; Section L compares these results in detail.

## 2 FAMILIES AND THEIR SAMPLING MAPS

All algorithms are uniform; $\langle\cdot\rangle$ is a canonical, injective, linear-length tuple encoding (Section K), $\mathsf{U}_n$ is the uniform law on $\{0,1\}^n$, and a function is negligible if it is eventually below $n^{-c}$ for every $c > 0$. A family consists of pairs $(L_s, D_s)$ for $s \in \{0,1\}^n$, $n \geq 1$, with $L_s$ infinite and $\mathrm{supp}(D_s) = L_s$. The experiment draws $S \sim \mathsf{U}_n$ and then $X_1, X_2, \ldots$ independently from $D_S$. Several descriptions may name the same language, and the prior is uniform on descriptions, which loses little generality among efficiently samplable priors, since a family can be re-indexed by the prior sampler's coins.

**Definition 2.1** (Presented family)**.** A family is presented when the following algorithms and bounds are public.

1. Membership in $L_s$ is decidable in time $\mathrm{poly}(|s| + |w|)$.
2. $\mathsf{Enum}(s,1), \mathsf{Enum}(s,2), \ldots$ lists $L_s$ without duplicates, the $j$th word in time $\mathrm{poly}(n + j)$.
3. $\mathsf{Smp}(s)$ halts almost surely with law $D_s$, and every terminating run producing $w$ uses at most $b(n + |w|)$ steps and random bits, for a known nondecreasing polynomial $b$.
4. A polynomial-time computable cutoff $\kappa(n,q) = \mathrm{poly}(n+q)$ satisfies $\mathbb{P}_{D_s}[|W| > \kappa(n,q)] \leq 1/q$ for every $s \in \{0,1\}^n$ and integer $q \geq 2$.

As in Kleinberg & Mullainathan (2024), a generator can test membership in every candidate language; only the target's description is hidden. The converse never uses membership, so efficient membership only strengthens the witness.

**Fresh generation.** Write $V_t = \{X_1, \ldots, X_t\}$. A generator outputs $A_t = A(1^n, X_{1:t}; \omega)$ with a persistent random tape $\omega$ and no feedback, and it is efficient if each invocation takes time polynomial in $B_t = n + t + \sum_{j \leq t} |X_j|$. Success means $A_t \in L_S \setminus V_t$, and for a positive integer-valued polynomial $H$,

$$\mathrm{Succ}_{A,H}(n) = \mathbb{P}[\exists t \leq H(n) : A_t \in L_S \setminus V_t]. \tag{1}$$

A family has a *strong polynomial-horizon gap* if this probability is negligible for every efficient $A$ and polynomial $H$, and a *weak gap* if, for some constant $\epsilon > 0$, it is at most $1 - \epsilon$ at all large $n$ for every efficient $A$ and $H$. Hardness is proved for the weakest criterion, one fresh word, and the converse for the strongest, permanent correctness, so the strong-gap equivalence holds for every criterion in between (applied, for (b) of Theorem 6.1, to premise and gap alike).

### 2.1 THE SAMPLING MAP AND ITS FIBRES

Fix a polynomially bounded, polynomial-time computable sample count $N = N(n)$ and cutoff $\kappa = \kappa(n)$, and put $b_\kappa = b(n + \kappa)$. Let $\mathsf{Smp}_{\leq\kappa}(s; r)$ run $\mathsf{Smp}(s)$ on exactly $b_\kappa$ random bits for at most $b_\kappa$ steps and return its word if that word has length at most $\kappa$, and $\perp$ otherwise. The *sampling map* on $d(n) = n + Nb_\kappa$ bits is

$$\Phi_n(s, r_1, \ldots, r_N) = \big(\mathsf{Smp}_{\leq\kappa}(s; r_1), \ldots, \mathsf{Smp}_{\leq\kappa}(s; r_N)\big). \tag{2}$$

A transcript is *short* if no entry is $\perp$. With shared tapes, $\Phi_n(U)$ for $U \sim \mathsf{U}_{d(n)}$ agrees with $X \sim D_S^{\otimes N}$ unless an original word is longer than $\kappa$, an event of probability $\tau_n \leq N \sup_s \mathbb{P}_{D_s}[|W| > \kappa]$.

**Lemma 2.2** (Fibres are posteriors)**.** *For every short transcript $z$ and every description $s$,*

$$\#\{r : \Phi_n(s,r) = z\} = 2^{Nb_\kappa} \prod_{j=1}^{N} D_s(z_j), \qquad \mathbb{P}[S = s,\ \Phi_n(U) = z] = \mathbb{P}[S = s,\ X = z]. \tag{3}$$

*Hence the description of a uniformly random point of $\Phi_n^{-1}(z)$ has the posterior law*

$$\pi_n(s \mid z) = \frac{\prod_{j=1}^{N} D_s(z_j)}{\sum_{u \in \{0,1\}^n} \prod_{j=1}^{N} D_u(z_j)}. \tag{4}$$

*Proof.* Every terminating run that produces a word of length at most $\kappa$ fits the budget $b_\kappa$, so extending its coin prefix to all $b_\kappa$-bit strings preserves its probability exactly; the count follows from independence of the coin blocks, and both probabilities equal $2^{-n} \prod_j D_s(z_j)$ (Section F). □

The identity is exact: it neither rounds probabilities nor assumes a likelihood oracle, and it does not condition the prior on short outputs. The observation is standard (Naor & Rothblum, 2006, proof of Theorem 4.2); see also Bogdanov & Trevisan (2006, Section 5.2) and Hirahara & Nanashima (2023).

### 2.2 Generators that may sample the posterior

Conditional on the examples, write $S^*$ for an independent posterior draw.

**Definition 2.3** (Posterior-sampling generators)**.** A posterior-sampling generator consists of polynomially bounded, polynomial-time computable $N_0(n) \geq 1$ and $q(n)$ and an efficient algorithm $A$ that acts as an ordinary generator before round $N_0$, receives $q$ independent draws $S_1^*, \ldots, S_q^*$ from $\pi_n(\cdot \mid X_{1:N_0})$ once and for all at round $N_0$, and afterwards outputs $A_t = A(1^n, X_{1:t}, S_{1:q}^*; \omega)$. A presented family belongs to $\mathcal{K}_{\mathrm{post}}$ if some posterior-sampling generator and polynomial $H$ satisfy $\mathrm{Succ}_{A,H}(n) \geq n^{-\gamma}$ for a constant $\gamma$ and all large $n$.

Every efficient generator is a posterior-sampling generator with $q = 0$, so $\mathcal{K}_{\mathrm{post}}$ contains every efficiently generatable family; the question is whether it contains more. The simplest posterior-sampling generator takes one draw.

**Proposition 2.4** (Bayesian generation)**.** *Let the* Bayesian generator *draw $S^*$ once from $\pi_n(\cdot \mid X_{1:N})$ and, at each $t \geq N$, output the first word of* $\mathsf{Enum}(S^*, \cdot)$ *outside $V_t$. Under the i.i.d. source,* $\mathbb{P}[\forall t \geq N : A_t \in L_S \setminus V_t] = \mathbb{P}[L_{S^*} \subseteq L_S]$.

Section F.4 gives the proof. Since $(S, S^*)$ is exchangeable given $X$ and distinct languages fail containment in some direction,

$$\mathbb{P}[L_{S^*} \not\subseteq L_S] \leq \mathbb{P}[L_{S^*} \neq L_S] \leq 2\mathbb{P}[L_{S^*} \not\subseteq L_S]. \tag{5}$$

**Definition 2.5** (Posterior-identifiable families)**.** A presented family belongs to $\mathcal{K}_{\mathrm{pid}}$ if a positive integer-valued, polynomially bounded and polynomial-time computable $N(n, p)$ satisfies $\mathbb{P}[L_{S^*} \neq L_S] \leq 1/p$ for $X \sim D_S^{\otimes N(n,p)}$, every $n \geq 1$ and every integer $p \geq 2$.

The posterior draw errs at most twice as often as the Bayes-optimal identifier (Proposition F.1), so $\mathcal{K}_{\mathrm{pid}}$ consists exactly of the presented families that some rule, efficient or not, identifies with error $1/p$ from $\mathrm{poly}(n, p)$ examples, condition (i) of Theorem 1.1, and with $N_0 = N(n, 2)$ the Bayesian generator gives $\mathcal{K}_{\mathrm{pid}} \subseteq \mathcal{K}_{\mathrm{post}}$. Three examples frame the premise (Sections I and J).

**Example 2.6** (Hidden parities)**.** For $a \in \{0,1\}^n$ let $L_a = \{\langle 1^n, x, \langle a, x_{1:n}\rangle\rangle : |x| \geq n\}$, with the inner product over $\mathbb{F}_2$ and $x \sim \mathsf{U}_{n+\ell}$ for a geometric $\ell$. The posterior errs only if the observed prefixes fail to span $\mathbb{F}_2^n$ (probability at most $2^{n-N}$), so this family $\mathcal{C}_{\mathrm{par}}$ lies in $\mathcal{K}_{\mathrm{pid}}$ with $N(n, p) = n + \lceil \log_2 p \rceil$; identification needs $\Theta(n)$ samples, but generation needs one, since extending an observed $x$ preserves its label.

**Example 2.7** (Easy generation where Bayesian generation fails)**.** For $L_s = A_n \cup \{u_{n,s}\}$, with a public enumerable core $A_n$ and a source emitting a distinct special word $u_{n,s} \notin A_n$ with probability $2^{-n}$ and otherwise an $s$-independent core word, enumerating unseen core words succeeds from the first example ($q = 0$), yet $\mathbb{P}[L_{S^*} \not\subseteq L_S] = (1 - 2^{-n})^{N+1}$: the family lies in $\mathcal{K}_{\mathrm{post}}$, far outside $\mathcal{K}_{\mathrm{pid}}$.

**Example 2.8** (Missing information)**.** For $c_n = \langle 1^n \rangle$ and $L_s = \{c_n\} \cup \{\langle 1^n, s, i\rangle : i \in \{0,1\}^*\}$ with $c_n$ emitted with probability $1 - 2^{-n}$, every generator, however powerful, has $\mathrm{Succ}_{A,H}(n) \leq 2H(n)2^{-n}$, since until another word appears the examples are independent of $s$ and every fresh valid word names $s$. Without a statistical premise, a gap need not involve computation; the family lies outside $\mathcal{K}_{\mathrm{post}}$.

## 3 From fibres to generators

A distributional inverter returns an approximately uniform point of the fibre over its input, in total variation distance TV:

$$\mathrm{TV}\big((\Phi_n(U), U), (\Phi_n(U), \mathsf{Inv}(1^n, \Phi_n(U)))\big) \leq \eta_n, \qquad U \sim \mathsf{U}_{d(n)}. \tag{6}$$

**Theorem 3.1** (Simulation). *Let $A$ be a posterior-sampling generator with prefix $N_0$ and $q$ draws, and let $\Phi_n$ be the sampling map for $N_0(n)$ examples, inverted as in (6). Let $A^{\mathsf{Inv}}$ replace the $i$th draw by the first $n$ bits of $\mathsf{Inv}(1^n, X_{1:N_0})$ with fresh coins and, from round $N_0$ on, output $\perp$ if a prefix example is longer than $\kappa$ or an inverse has the wrong length. Then $A^{\mathsf{Inv}}$ is efficient, and the joint laws of target, examples and outputs under $A$ and $A^{\mathsf{Inv}}$ are within total variation $q\eta_n + \tau_n$, also when later examples are chosen from $L_S$ by a rule that sees the target, the examples and the outputs.*

*Proof.* The trajectory is a fixed function of $(S, X_{1:N_0})$, the draws $T$, and randomness independent of $T$ given $(S, X_{1:N_0})$, so it suffices to compare the laws of $(S, X_{1:N_0}, T)$. In both runs $T$ comes from a kernel depending only on $x = X_{1:N_0}$: the posterior product $K_x^{\otimes q}$, or the inverter product $Q_x^{\otimes q}$ on short $x$ and a point mass at $\perp$ otherwise. Transcripts that are not short contribute at most $\tau_n$. On short $x$, $\mathrm{TV}(K_x^{\otimes q}, Q_x^{\otimes q}) \le q\,\mathrm{TV}(K_x, Q_x)$, and by Lemma 2.2 weights and posteriors coincide with those of $\Phi_n(U)$, so these terms sum to at most $q\eta_n$ by projecting (6) to descriptions (Section F). □

The comparison must be joint with the target: for a uniform bit $S$, both $S$ and $1 - S$ have the right marginal law, yet only one agrees with the target. For the Bayesian generator, Theorem 3.1 and Proposition 2.4 give an efficient generator permanently successful with probability at least $1 - \alpha_n - \eta_n - \tau_n$, where $\alpha_n = \mathbb{P}[L_{S^*} \not\subseteq L_S]$. The draws are of the hidden description: a resampled *example* is fresh only with vanishing probability, since $D_S(V_t) \to 1$, so extrapolating the stream of examples does not help, while extrapolating a sequence that appends an enumeration of the language would reproduce the Bayesian generator for a fixed horizon (Section F).

## 4 Without one-way functions

For a polynomial-time $f$ and a uniform PPT inverter $I$, let

$$\mathrm{err}_{f,I}(m) = \mathrm{TV}\big((f(U_m), U_m), (f(U_m), I(1^m, f(U_m)))\big). \tag{7}$$

Impagliazzo & Luby (1989) proved that one-way functions exist exactly when some $f$ and $c > 0$ satisfy $\mathrm{err}_{f,I}(m) \ge m^{-c}$ for every $I$ and all large $m$ (Naor & Rothblum, 2006, Section 2.1). Without one-way functions, each $f$ and $c$ therefore admit one inverter with error below $m^{-c}$ at infinitely many lengths $m$ (Section B.2); with Lemma 2.2, this is universal extrapolation in Bayesian form (Impagliazzo & Levin, 1990; Ostrovsky & Wigderson, 1993; Hirahara & Nanashima, 2023). These good lengths need not include the lengths $d(n)$ of our sampling maps, so we place each map at every length of a block.

**Lemma 4.1** (Padding across blocks). *Let polynomial-time maps $\Phi_n$ have input lengths $d(n) \le n^D$ for $n \ge n_0$, let $\rho(m) = \lfloor m^{1/D} \rfloor$ and $J_n = \rho^{-1}(n) = \{n^D, \dots, (n+1)^D - 1\}$, and define, on $\{0,1\}^m$ for $m \ge n_0^D$ and with a distinct format at smaller lengths,*

$$f(x) = \langle 1^m, 1^{\rho(m)}, \Phi_{\rho(m)}(x_{1:d(\rho(m))}) \rangle. \tag{8}$$

*(i) For $m \in J_n$, the first $d(n)$ bits of a distributional inverse of $f$ at length $m$ invert $\Phi_n$ with no larger error. (ii) If one inverter has error below $m^{-c}$ at infinitely many $m$, then for infinitely many $n$ some $m \in J_n$ yields an inverter for $\Phi_n$ with error below $n^{-c}$.*

*Proof.* Padding multiplies every fibre of $\Phi_n$ by $2^{m-d(n)}$ and the field $1^m$ separates lengths, so (i) is data processing; for (ii), the finite blocks partition all large integers and $m \ge n$ (Section G). □

Padding of this kind is standard; here one inverter serves every block at once, and the generator never needs to know which length is good.

**Theorem 4.2** (Collapse). *Suppose uniform one-way functions do not exist, and let $A$ be a posterior-sampling generator with prefix $N_0$ and $q$ draws for a presented family.*

1. *Let success be any event determined by the target, the examples and the outputs, such as a fresh valid word within a horizon or permanent correctness. If $A$ succeeds with probability*

*at least $n^{-\gamma}$ at all large $n$, then some efficient $A'$ and integer $D \geq 1$ satisfy*

$$\mathbb{P}[A' \text{ succeeds}] \geq \frac{1}{4n^{\gamma}((n+1)^D - n^D)} \qquad \text{for infinitely many } n. \tag{9}$$

2. *If $A$ is permanently successful from round $N_0$ with probability at least $1-\epsilon(n)$ at all large $n$, then for every $c > 0$ some efficient $A''$ and $D \geq 1$ satisfy $\mathrm{Succ}_{A'',H''}(n) \geq 1-\epsilon(n)-2n^{-c}$ for infinitely many $n$, where $H'' = N_0 + (n+1)^D - n^D$.*

*Proof.* Choose the cutoff and the inversion exponent of Lemma 4.1 so that $\tau_n$ and $q\eta_n$ are at most $n^{-\gamma}/4$ in (i) and $n^{-c}$ in (ii). In (i), $A'$ picks $m \in J_n$ with probability at least $1/(2|J_n|)$ each and runs $A^{\mathsf{Inv}}$ at length $m$; at a good length Theorem 3.1 gives success at least $n^{-\gamma}/2$. In (ii), round $N_0 + j$ runs $A^{\mathsf{Inv}}$ at the $j$th length of $J_n$ and outputs that run's output for the round; at the good length that output is fresh and valid with probability at least $1 - \epsilon(n) - 2n^{-c}$ (Section G). □

**Corollary 4.3.** *Without uniform one-way functions, no family in $\mathcal{K}_{\mathrm{post}}$ has a strong polynomial-horizon gap, and no family in $\mathcal{K}_{\mathrm{pid}}$ has a weak gap.*

*Proof.* The first claim is Theorem 4.2(i). For the second, given $\epsilon > 0$ take $p > 2/\epsilon$ and apply Theorem 4.2(ii) with $c = 1$ to the Bayesian generator with $N_0 = N(n,p)$, which by Proposition 2.4, (5) and Definition 2.5 is permanently successful with probability at least $1 - 1/p$; success is then at least $1 - 1/p - 2/n > 1 - \epsilon$ at infinitely many $n$. □

*Remark* 4.4 (The sampling map is distributionally one-way). The proof shows more: if a family in $\mathcal{K}_{\mathrm{post}}$ has a strong gap, then the padded sampling map $f$ of Lemma 4.1, built as in Section G from a posterior-sampling generator that witnesses membership in $\mathcal{K}_{\mathrm{post}}$, is distributionally one-way: every uniform PPT inverter has error at least $m^{-(\gamma+e+2)}$ at all large $m$, where $q(n) \leq n^e$. Generation from sufficient data is hard only if inverting the family's own data-generating process in distribution is hard. The map need not be one-way in the stronger sense of finding some preimage (Section H).

## 5 With one-way functions: rigidification

Assume one-way functions exist, so that pseudorandom generators $G_\lambda : \{0,1\}^\lambda \to \{0,1\}^{3\lambda}$, $E_\lambda : \{0,1\}^\lambda \to \{0,1\}^{\lambda^2}$ and pseudorandom functions exist (Håstad et al., 1999; Goldreich et al., 1986). A pseudorandom-function graph makes fresh generation a prediction at an unseen input, but it is not identifiable: different keys may agree on every likely example. We restore identifiability for any keyed family with the commitment idea of Naor (1991) (Figure 5). Choose independent $k, z \sim \mathsf{U}_\lambda$ and $R \sim \mathsf{U}_{3\lambda}$, split $E_\lambda(z)$ into blocks $\rho_1, \ldots, \rho_\lambda \in \{0,1\}^\lambda$, and set

$$C_j = G_\lambda(\rho_j) \oplus k_j R, \qquad P = (R, C_1, \ldots, C_\lambda), \tag{10}$$

where $0R = 0^{3\lambda}$ and $1R = R$: a zero bit places $C_j$ in $\mathrm{im}(G_\lambda)$, a one bit in its shift by $R$.

**Lemma 5.1** (Binding and hiding). *Let $\Delta_\lambda = \{G_\lambda(a) \oplus G_\lambda(b) : a, b \in \{0,1\}^\lambda\}$. Then $\mathbb{P}_R[R \in \Delta_\lambda] \leq 2^{-\lambda}$, and outside this event a tag of the form (10) admits at most one key, even for correlated openings. With $\widetilde{P} = (R, U_1, \ldots, U_\lambda)$ for independent uniform $U_j \in \{0,1\}^{3\lambda}$, every uniform PPT test $T$ has $|\mathbb{P}[T(1^\lambda, k, R, P) = 1] - \mathbb{P}[T(1^\lambda, k, R, \widetilde{P}) = 1]| \leq \mathrm{negl}(\lambda)$.*

Binding uses only $|\mathrm{im}\, G_\lambda| \leq 2^\lambda$; hiding replaces $E_\lambda(z)$ and then each $G_\lambda(\rho_j)$ by uniform strings, and the test may generate any key-dependent data (Section C). The secret $(k, R, z)$ has $5\lambda$ bits although the tag has $3\lambda(\lambda+1)$.

**Definition 5.2** (Keyed families and their rigidification). A *keyed family* assigns to each key $k \in \{0,1\}^\lambda$ an infinite language $L_k$ with a full-support source $D_k$, presented with $k$ as the description. Its *rigidification* $\mathcal{K}^\sharp$ has, at length $n \geq 5$ with $\lambda = \lfloor n/5 \rfloor$, descriptions $s = (k, R, z, \mathrm{pad})$ with $n - 5\lambda \leq 4$ ignored padding bits, the tag $P$ of (10), and

$$L^\sharp_s = \{\langle 1^n, P, 0, w\rangle : w \in L_k\} \cup \{w^\sharp_s\}, \qquad w^\sharp_s = \langle 1^n, P, 1, k\rangle; \tag{11}$$

its source returns the *revealing word* $w_s^\sharp$ with probability $\delta_n = 2^{-\lambda}$ and otherwise $\langle 1^n, P, 0, w\rangle$ with $w \sim D_k$. Lengths $n < 5$ use a fixed public language.

**Lemma 5.3** (Rigidification). *For every keyed family $\mathcal{K}$, the family $\mathcal{K}^\sharp$ is presented, and for every $n \geq 5$ and $N \geq 1$,*

$$\mathbb{P}[L^\sharp_{S^*} \neq L^\sharp_S] \leq \delta_n(1-\delta_n)^N \leq \frac{1}{eN}. \tag{12}$$

*Hence $\mathcal{K}^\sharp \in \mathcal{K}_{\mathrm{pid}}$ with $N(n,p) = p$, and after one example its Bayesian generator succeeds at every round with probability at least $1 - 2^{-\lfloor n/5\rfloor}$.*

*Proof.* Presentation is routine (Section D). Posterior-supported descriptions reproduce the observed tag, and $L^\sharp_s$ depends only on $(n, P, k)$. Off $R \in \Delta_\lambda$ the tag fixes $k$, and a revealing sample fixes $k$ even on it; an error needs a bad mask (probability at most $\delta_n$) and no revelation in $N$ samples (probability $(1-\delta_n)^N$ given every description), and $\delta(1-\delta)^N \leq \delta e^{-N\delta} \leq 1/(eN)$. The last claim follows with Proposition 2.4 and (5). □

The revealing word resolves the rare bad masks, which repeating the tag cannot; it is needed because Definition 2.5 demands confidence $1 - 1/p$ for every $p$ at each fixed $n$, which $2^{-\lambda}$ alone does not give. Computationally, the tag changes success probabilities only negligibly:

**Lemma 5.4** (Hardness transfer). *Let $\mathcal{K}$ be keyed with cutoff $\kappa(\lambda, q) = \mathrm{poly}(\lambda + \log q)$. For every efficient generator $A$ for $\mathcal{K}^\sharp$ and polynomial $h$ there are a negligible $\nu$ and an efficient $B$ with*

$$\mathrm{Succ}_{A,h}(n) \leq h\delta_n + 2^{-2\lambda-1} + \nu(n) + \mathrm{Win}_B(n), \tag{13}$$

*where $\mathrm{Win}_B(n)$ is the probability that $B$, receiving one example of $D_k$ per round for a uniform key $k$ and outputting one candidate per round, produces within $h(n)$ rounds a word of $L_k$ not yet shown or the key $k$.*

*Proof sketch.* Coupling removes revelations at cost $h\delta_n$ (the revealing word stays in the target), capping sample lengths at $\kappa(\lambda, h2^{2\lambda+1}) = \mathrm{poly}(\lambda)$ costs $2^{-2\lambda-1}$, and the bounded experiment is a polynomial-time test given $k$ and $R$, so Lemma 5.1 replaces $P$ by $\widetilde{P}$ at negligible cost. With $\widetilde{P}$, the experiment is the game of the $B$ that samples $\widetilde{P}$ itself, wraps each example, runs $A$ and unwraps: a success is a fresh tagged word of $L_k$ or the revealing word, which names $k$ (Section E). □

Let $\mathcal{G}$ be the pseudorandom-function graph, $L_k = \{\langle i, F_k(i)\rangle : i \in \{0,1\}^*\}$ with $F_k : \{0,1\}^* \to \{0,1\}^{2\lambda}$ and a geometric index length, whose cutoff is $O(\lambda + \log q)$, and let $\mathcal{C}_{\mathrm{tag}} = \mathcal{G}^\sharp$ (Figure 6).

**Theorem 5.5** (Polynomial-horizon hardness). *For every uniform efficient generator $A$ and polynomial horizon $H$, $\mathrm{Succ}_{A,H}(n)$ for $\mathcal{C}_{\mathrm{tag}}$ is negligible.*

*Proof.* Apply Lemma 5.4 for each residue of $n$ modulo 5, and bound $\mathrm{Win}_B$ by a distinguisher $D^O$ with *hidden checks*: it answers example indices with $O$, runs $B$, and checks each candidate with one query $B$ never sees (accepting if one passes), a word $\langle i, y\rangle$ at an unshown index by $O(i) = y$ and a key $k'$ by $O(u) = F_{k'}(u)$ at a fresh index $u$. Under $F_k$ every win passes, since each index carries one word; under a random function each checked value is uniform given $B$'s view, so $D$ accepts with probability at most $h2^{-2\lambda}$. PRF security thus gives $\mathrm{Win}_B(n) \leq h2^{-2\lambda} + \mathrm{negl}(\lambda)$ and

$$\mathrm{Succ}_{A,H}(5\lambda + r) \leq h2^{-\lambda} + 2^{-2\lambda-1} + \nu(n) + h2^{-2\lambda} + \mathrm{negl}(\lambda) = \mathrm{negl}(\lambda). \tag{14}$$

□

An unbounded generator recovers $k$ from one example in time $2^\lambda\,\mathrm{poly}(\lambda)$ off the bad-mask event, and an efficient one can wait for the revealing word, of mean waiting time $2^\lambda$, and then enumerate (Figure 8); it succeeds eventually on every complete enumeration (Section D). With $s(\lambda)$-secure primitives the bound becomes explicit: for $s(\lambda) \geq 2^{c\lambda}$ with a constant $c \in (0, 1/2]$, every generator running in total time $2^{c\lambda/3}$ succeeds with probability $2^{-\Omega(\lambda)}$, which key search shows to be tight up to the constant in the exponent (Proposition E.1).

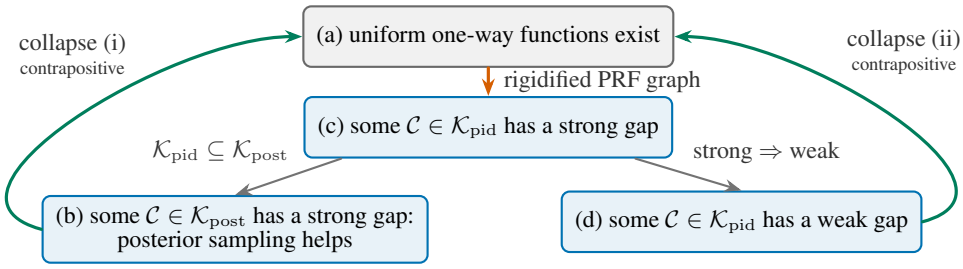



Figure 3: How the four statements of Theorem 6.1 are linked: the rigidified PRF graph gives (a)⇒(c) (orange; Lemma 5.3 and Theorem 5.5), inclusions give (c)⇒(b) and (c)⇒(d), and the collapse closes both cycles (green; Corollary 4.3).

## 6 The barrier

**Theorem 6.1** (Cryptographic barrier)**.** *The following are equivalent: (a) uniform one-way functions exist; (b) posterior sampling helps generation: some family in $\mathcal{K}_{\mathrm{post}}$ has a strong polynomial-horizon gap; (c) some family in $\mathcal{K}_{\mathrm{pid}}$ has a strong polynomial-horizon gap; (d) some family in $\mathcal{K}_{\mathrm{pid}}$ has a weak gap. Under (a), $\mathcal{C}_{\mathrm{tag}}$ witnesses (c) and moreover: (1) after one example its Bayesian generator succeeds at every round with probability at least $1 - 2^{-\lfloor n/5 \rfloor}$; (2) after $N \geq 1$ examples its posterior identification error is at most $1/(eN)$; (3) an efficient generator succeeds eventually on every complete enumeration of every target.*

*Proof.* (a)⇒(c) by Lemma 5.3 and Theorem 5.5; (c)⇒(b) as $\mathcal{K}_{\mathrm{pid}} \subseteq \mathcal{K}_{\mathrm{post}}$; (c)⇒(d) trivially; (b)⇒(a) and (d)⇒(a) by Corollary 4.3. Items (1)–(2) are Lemma 5.3; (3) is in Section D (Figure 3). □

*Remark* 6.2 (A universal witness). The statistical properties of $\mathcal{C}_{\mathrm{tag}}$ hold for every efficiently computable instantiation of $G_\lambda$, $E_\lambda$ and $F$, because binding uses only the size of the image of $G_\lambda$ (Lemma 5.1). Instantiating them from Levin's universal one-way function (Levin, 1987), through hardness amplification and the explicit constructions of Håstad et al. (1999) and Goldreich et al. (1986) (see Goldreich, 2001), therefore gives one explicit family in $\mathcal{K}_{\mathrm{pid}}$ that has a strong polynomial-horizon gap if and only if one-way functions exist, and if and only if it has a weak gap.

Under (a), Example H.1 gives a hard family in $\mathcal{K}_{\mathrm{post}}$ that the Bayesian generator cannot handle (Section H). Let $\mathrm{Succ}^{\mathrm{id}}_{A,H}(n) = \mathbb{P}[\exists t \leq H(n) : A_t \in \{0,1\}^n,\ L_{A_t} = L_S]$, with identification gaps defined as for generation.

**Corollary 6.3** (Identification)**.** *Uniform one-way functions exist if and only if some family in $\mathcal{K}_{\mathrm{pid}}$ has a strong identification gap, and if and only if some family in $\mathcal{K}_{\mathrm{pid}}$ has a weak identification gap.*

*Proof.* An identifier yields a generator with at least its success, by enumerating its output's language; conversely, the proof of Theorem 4.2(ii) applies to the identifier that outputs its posterior draw from round $N(n,p)$ on, correct with probability at least $1 - 1/p$ (Section G). □

## 7 Scope and open directions

The barrier concerns the specified prior, source and public algorithms, and its lower bound is average-case over polynomial horizons. The Bayesian generator idealizes in-context learning as implicit posterior inference (Xie et al., 2022); the theorem shows that this idealization can be computationally out of reach even when one example suffices statistically. The premise $\mathcal{K}_{\mathrm{post}}$ is relative to posterior samples, and an intrinsic, information-theoretic description of it remains open, as do adversarial presentations and worst-case targets (Li et al., 2026), natural classes such as regular languages (Section L), and a dual equivalence for almost one-way functions, in which inverters at all large lengths would make the padding unnecessary. When the examples already suffice, the only thing that can keep unseen words out of efficient reach is the one-wayness of the process that produced them.

## AI USE STATEMENT

The original research ideas and initial manuscript draft were developed by the authors. AI tools were used to assist with checking mathematical details and the correctness of arguments, filling in routine proof steps, drawing TikZ figures, and language polishing. The authors take full responsibility for the content and correctness of the final manuscript.

## REPRODUCIBILITY STATEMENT

The model and assumptions appear in Section 2. The main text gives the general principles, the construction, and both directions of the equivalence; Sections B to K provide the cryptographic interfaces, exact sampling calculations, and proof details. The source package contains the LaTeX files, bibliography, supplied conference style files, and a compilation script. The results are mathematical; no finite computation is used as evidence of asymptotic cryptographic security.

## ETHICS STATEMENT

This work studies an abstract generation problem under standard cryptographic assumptions. It does not train models, collect personal data, or evaluate human participants. The synthetic construction is not an attack on an implemented system, and the theorem makes no claim about the behavior of trained language models.

## REFERENCES

Antonios Anastasopoulos, Giuseppe Ateniese, and Evgenios M. Kornaropoulos. Safe language generation in the limit, 2026.

Marcelo Arenas, Pablo Barceló, Luis Cofré, and Alexander Kozachinskiy. Language generation in the limit: Complexity barriers and implications for learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. arXiv:2511.05759.

Eric Binnendyk, Marco Carmosino, Antonina Kolokolova, R Ramyaa, and Manuel Sabin. Learning with distributional inverters. In *Proceedings of the 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pp. 90–106, 2022. URL https://proceedings.mlr.press/v167/binnendyk22a.html.

Avrim Blum. Separating distribution-free and mistake-bound learning models over the Boolean domain. *SIAM Journal on Computing*, 23(5):990–1000, 1994.

Avrim Blum, Merrick L. Furst, Michael Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology – CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pp. 278–291, 1993.

Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Foundations and Trends in Theoretical Computer Science*, 2(1):1–106, 2006. doi: 10.1561/0400000004.

Scott Decatur, Oded Goldreich, and Dana Ron. Computational sample complexity. *SIAM Journal on Computing*, 29(3):854–879, 1999.

Nicolas Flammarion, Chirag Pabbaraju, Hristo Papazov, Miltiadis Stouras, and Ola Svensson. Space-efficient language generation in the limit. In *Proceedings of the 39th Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 2477–2502, 2026.

Oded Goldreich. A note on computational indistinguishability. *Information Processing Letters*, 34 (6):277–281, 1990. doi: 10.1016/0020-0190(90)90010-U.

Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001. doi: 10.1017/CBO9780511546891.

Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. doi: 10.1145/6490.6503.

Shivam Gupta, Ajil Jalal, Aditya Parulekar, Eric Price, and Zhiyang Xun. Diffusion posterior sampling is computationally intractable. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17020–17059, 2024. URL https://proceedings.mlr.press/v235/gupta24a.html.

Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. doi: 10.1137/S0097539793244708.

Shuichi Hirahara and Mikito Nanashima. Learning in Pessiland via inductive inference. In *Proceedings of the 64th IEEE Symposium on Foundations of Computer Science*, pp. 447–457, 2023.

Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference*, pp. 134–147, 1995. doi: 10.1109/SCT.1995.514853.

Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pp. 812–821, 1990.

Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pp. 230–235, 1989. doi: 10.1109/SFCS.1989.63483.

Héctor Jimenez, Alexander Kozachinskiy, and Vicente Opazo. Polynomial-time mistake-bounded language generation, 2026. URL https://arxiv.org/abs/2606.16077v2. Version 2.

Alkis Kalavasis, Anay Mehrotra, and Grigoris 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*, pp. 1732–1743, 2025. doi: 10.1145/3717823.3718108.

Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pp. 273–282, 1994. doi: 10.1145/195058.195155.

Jon Kleinberg and Sendhil Mullainathan. Language generation in the limit. In *Advances in Neural Information Processing Systems*, volume 37, pp. 66058–66079, 2024. doi: 10.52202/079017-2111. URL https://arxiv.org/abs/2404.06757.

Jon Kleinberg, Charlotte Peale, and Omer Reingold. Mistake-bounded language generation, 2026. URL https://arxiv.org/abs/2605.10809.

Leonid A. Levin. One way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987. doi: 10.1007/BF02579323.

Xiaoyu Li, Andi Han, Dai Shi, Jiaojiao Jiang, and Junbin Gao. On computational hardness of mistake-bounded language generation: A random-oracle query separation, 2026. URL https://arxiv.org/abs/2608.05029.

Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4:151–158, 1991. doi: 10.1007/BF00196774.

Moni Naor. Evaluation may be easier than generation. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 74–83, 1996. doi: 10.1145/237814.237833.

Moni Naor and Guy N. Rothblum. Learning to impersonate. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 649–656, 2006. URL https://www.wisdom.weizmann.ac.il/~naor/PAPERS/acd.pdf. Author's full version.

Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings of the 2nd Israel Symposium on Theory and Computing Systems*, pp. 3–17, 1993. doi: 10.1109/ISTCS.1993.253489.

Vinod Raman, Jiaxun Li, and Ambuj Tewari. Generation through the lens of learning theory. In *Proceedings of the 38th Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pp. 4740–4776, 2025. URL https://proceedings.mlr.press/v291/raman25a.html.

Sasha Voitovych, Abhishek Shetty, Noah Golowich, and Alexander Rakhlin. Learning with simulators: No regret in a computationally bounded world, 2026.

David Xiao. Learning to create is as hard as learning to appreciate. In *Proceedings of the 23rd Conference on Learning Theory*, pp. 516–528, 2010.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.02080.

## CONTENTS OF THE APPENDIX

**A Notation and dependencies** **13**

**B Cryptographic conventions and imported results** **16**

B.1 Polynomial stretch and a fixed-key variable domain . . . . . 16

B.2 The exact negation used in the converse . . . . . 16

**C Expanded-opening tags and uniform hybrids** **17**

**D Rigidification: presentation, confidence, and two generators** **18**

**E Bounded simulation and the hardness interface** **19**

E.1 Sharpness under exponential security . . . . . 21

**F Fibres, simulation, and the Bayesian generator** **22**

F.1 Exact short-output probabilities . . . . . 22

F.2 Keeping the hidden target in the error bound . . . . . 23

F.3 From a description to permanent generation . . . . . 23

F.4 The Bayesian generator as the exact case . . . . . 24

F.5 Bayes-risk characterization of posterior identifiability . . . . . 24

**G Padding and collapse** **24**

**H More on the barrier** **26**

**I The hidden-parity family** **27**

**J A presented common-core family** **27**

**K Canonical encoding and probability spaces** **28**

**L Comparison with computational generation and impersonation** **29**

## A NOTATION AND DEPENDENCIES

Table 1 collects the notation of the main text and the place where each symbol is defined. Figure 4 shows how the numbered results depend on one another; Figure 3, next to Theorem 6.1, shows how they combine into its four statements.

Table 1: Notation of the main text, grouped by topic.

| Symbol | Meaning | Defined in |
|---|---|---|
| *Families and generation* | | |
| $\langle\cdot\rangle$, $\mathsf{U}_n$ | canonical tuple encoding; uniform law on $\{0,1\}^n$ | Section 2 |
| negl, TV | negligible function; total variation distance | Section 2, (6) |
| $n$, $s \in \{0,1\}^n$ | description length; a description | Section 2 |
| $L_s$, $D_s$ | language named by $s$ and its source, with $\mathrm{supp}(D_s) = L_s$ | Section 2 |
| $S$, $X_1, X_2, \ldots$ | hidden target description $S \sim \mathsf{U}_n$; examples drawn i.i.d. from $D_S$ | Section 2 |
| $\mathsf{Enum}$, $\mathsf{Smp}$, $b$, $\kappa(n,q)$ | enumerator; sampler; step and coin bound; cutoff with tail $\le 1/q$ | Definition 2.1 |
| $V_t$, $A_t$, $\omega$ | the first $t$ examples; output at round $t$; persistent random tape | (1) |
| $B_t$ | accumulated input length $n + t + \sum_{j \le t} \lvert X_j \rvert$ | (1) |
| $H$, $\mathrm{Succ}_{A,H}(n)$ | horizon; probability of a fresh valid word within $H(n)$ rounds | (1) |
| strong gap | $\mathrm{Succ}_{A,H}$ negligible for all efficient $A$ and polynomial $H$ | Section 2 |
| weak gap | for one $\epsilon > 0$, $\mathrm{Succ}_{A,H} \le 1 - \epsilon$ at all large $n$, for all such $A$, $H$ | Section 2 |
| *The sampling map* | | |
| $N$, $\kappa$ | sample count and length cutoff of the sampling map | (2) |
| $b_\kappa$, $d(n)$ | coins per example, $b(n+\kappa)$; input length $n + Nb_\kappa$ | (2) |
| $\mathsf{Smp}_{\le\kappa}$, $\Phi_n$ | sampler on $b_\kappa$ coins, $\perp$ beyond length $\kappa$; the sampling map | (2) |
| $r$, $U$ | coins $(r_1, \ldots, r_N)$; uniform input $U \sim \mathsf{U}_{d(n)}$ of $\Phi_n$ | (2) |
| $z$, $\Phi_n^{-1}(z)$ | a transcript and its fibre | Lemma 2.2 |
| $\tau_n$ | probability that one of the $N$ examples is longer than $\kappa$ | (2) |
| $\pi_n(\cdot \mid z)$ | posterior on descriptions given the transcript $z$ | (4) |
| *Posterior sampling* | | |
| $S^*$, $S_1^*, \ldots, S_q^*$ | independent draws from the posterior given the examples | Definition 2.3 |
| $N_0$, $q$ | round of the posterior draws and their number | Definition 2.3 |
| $\mathcal{K}_{\mathrm{post}}$ | families generatable with posterior samples | Definition 2.3 |
| $\gamma$ | exponent in $\mathrm{Succ}_{A,H}(n) \ge n^{-\gamma}$ at all large $n$ | Definition 2.3 |
| $\mathcal{K}_{\mathrm{pid}}$, $N(n,p)$ | posterior-identifiable families; examples that bring the error to $1/p$ | Definition 2.5 |
| $\alpha_n$ | $\mathbb{P}[L_{S^*} \not\subseteq L_S]$, failure of permanent Bayesian generation | Section 3 |
| $\mathcal{C}_{\mathrm{par}}$ | the hidden-parity family | Example 2.6 |
| *Inversion and padding* | | |
| $\mathsf{Inv}$, $\eta_n$ | distributional inverter of $\Phi_n$ and its error | (6) |
| $A^{\mathsf{Inv}}$ | the generator $A$ with its posterior draws replaced by inversions | Theorem 3.1 |
| $\mathrm{err}_{f,I}(m)$ | distributional inversion error of $I$ for $f$ at input length $m$ | (7) |
| $D$, $\rho(m)$, $J_n$ | padding degree; $\lfloor m^{1/D} \rfloor$; the block $\{n^D, \ldots, (n+1)^D - 1\}$ | Lemma 4.1 |
| $f$ | padded sampling map | (8) |
| $\epsilon(n)$ | failure probability of permanent success in Theorem 4.2(ii) | Theorem 4.2 |
| $e$ | exponent with $q(n) \le n^e$; in (12) and Theorem 6.1, Euler's number | Remark 4.4 |
| *Rigidification and the witness* | | |
| $\lambda$ | security parameter $\lfloor n/5 \rfloor$ | Definition 5.2 |
| $G_\lambda$, $E_\lambda$ | pseudorandom generators into $\{0,1\}^{3\lambda}$ and $\{0,1\}^{\lambda^2}$ | Section 5 |
| $F_k$ | pseudorandom function with key $k$ | Section 5 |
| $k$, $R$, $z$, $\rho_j$ | key, mask, seed, and openings, the blocks of $E_\lambda(z)$ | (10) |
| $C_j$, $P$ | commitment blocks and the tag | (10) |
| $\Delta_\lambda$, $\widetilde{P}$ | bad masks; the key-independent tag | Lemma 5.1 |
| $\mathcal{K}$, $\mathcal{K}^\sharp$, $L_k$, $D_k$ | keyed family; its rigidification; language and source of key $k$ | Definition 5.2 |
| $L_s^\sharp$, $w_s^\sharp$, $\delta_n$ | rigidified language; revealing word and its probability $2^{-\lambda}$ | (11) |
| $\mathcal{G}$, $\mathcal{C}_{\mathrm{tag}}$ | pseudorandom-function graph and the witness $\mathcal{G}^\sharp$ | Section 5 |
| $h$, $\nu$ | horizon and negligible loss in the hardness transfer | Lemma 5.4 |
| $\mathrm{Win}_B(n)$ | chance that $B$ outputs a fresh word of $L_k$, or $k$, within $h$ rounds | Lemma 5.4 |
| $s(\lambda)$ | security level of the primitives | Proposition E.1 |
| $\mathrm{Succ}_{A,H}^{\mathrm{id}}(n)$ | probability of naming the target language within $H(n)$ rounds | Section 6 |

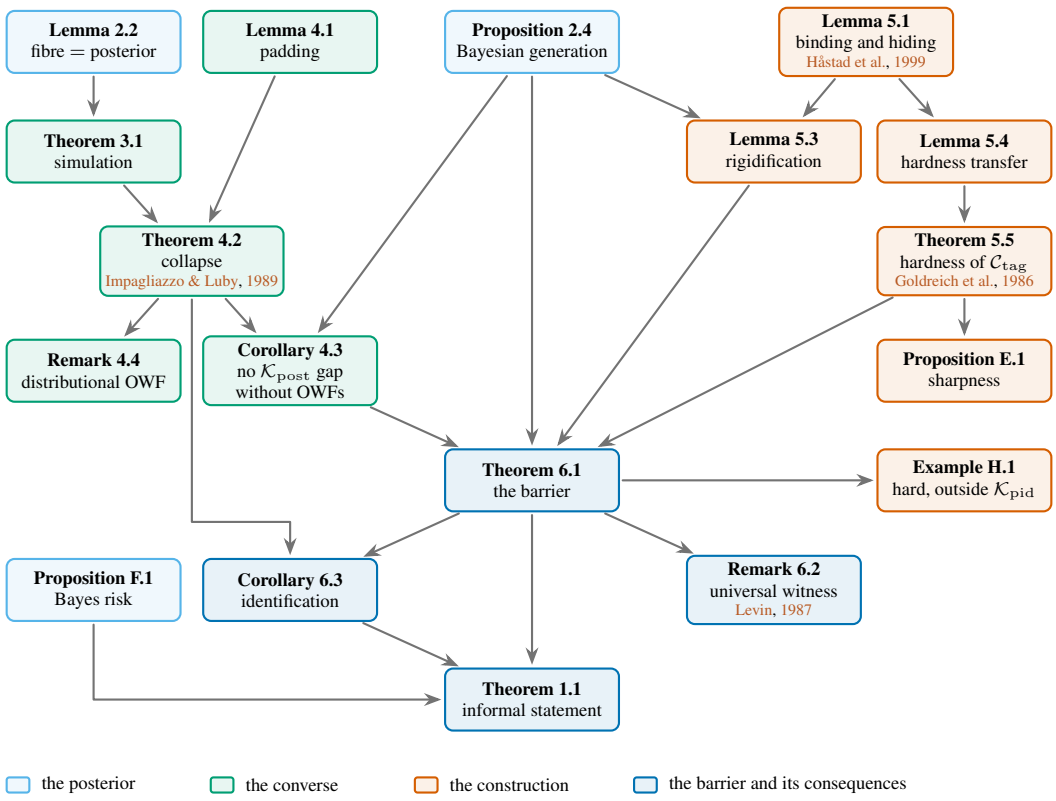



Figure 4: Dependencies among the numbered results. Each arrow is a use in a proof, and a result depends on another exactly when it can be reached from it along the arrows; uses implied by longer paths are not all drawn. A box whose proof imports a published theorem names it, and every box links to its statement.

# B CRYPTOGRAPHIC CONVENTIONS AND IMPORTED RESULTS

A one-way function is a uniformly polynomial-time map $f : \{0,1\}^* \to \{0,1\}^*$ such that every uniform PPT $I$ has negligible probability of outputting $y \in \{0,1\}^m$ with $f(y) = f(U_m)$ on input $(1^m, f(U_m))$. Including the input length in the function output makes this fixed-length-preimage convention equivalent, at the level of existence, to the usual formulation. Security is an all-sufficiently-large-length statement; its threshold may depend on the adversary and the desired inverse-polynomial bound.

We import two existence/equivalence theorems. First, one-way functions imply pseudorandom generators (Håstad et al., 1999), and pseudorandom generators imply pseudorandom functions (Goldreich et al., 1986). Second, distributionally one-way functions exist if and only if one-way functions exist (Impagliazzo & Luby, 1989); Naor & Rothblum (2006, Section 2.1) state the definition and equivalence with explicit length quantifiers. We do not reprove these foundational cryptographic theorems. All other arguments needed for the stated construction and converse are supplied here or in the main text.

For Proposition E.1 we use a quantitative notion. A primitive is $s(\lambda)$-*secure* if, for all sufficiently large $\lambda$, every uniform algorithm running in time at most $s(\lambda)$ distinguishes it from its ideal counterpart (uniform strings, or a uniformly random function) with advantage at most $1/s(\lambda)$. A distinguisher that tries $s$ seeds of a length-increasing generator with a $\lambda$-bit seed has advantage about $s2^{-\lambda}$, so $s(\lambda)$-security forces $s(\lambda) \le 2^{\lambda/2}\,\mathrm{poly}(\lambda)$.

## B.1 POLYNOMIAL STRETCH AND A FIXED-KEY VARIABLE DOMAIN

From a doubling PRG $T_\lambda : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ one obtains polynomial stretch by iterating a state/output split. Starting with $s_0 = z$, write $T_\lambda(s_{j-1}) = (s_j, v_j)$ and concatenate $v_1, \dots, v_\lambda$. This gives an explicit candidate $E_\lambda(z) \in \{0,1\}^{\lambda^2}$. A hybrid replaces each state expansion by an independent uniform state/output pair. At a chosen step the distinguisher knows the prior outputs and uses its challenge for that expansion; the remaining computation uses the state half of the challenge. Telescoping over $\lambda$ steps gives pseudorandomness. The same argument provides the constant stretch $3\lambda$ used for $G_\lambda$. Security need hold only asymptotically; at $\lambda = 1$ the specified expander may be any total map of the required lengths. This does not affect any cryptographic conclusion, and the statistical bounds do not require security at that length.

For completeness, a fixed key can handle every finite PRF input, rather than only one prescribed input length. Let $T_0, T_1$ be the two halves of a doubling PRG, and let $Q_\lambda$ be a doubling PRG used for final output. Use the prefix-free code $c(i) = 1^{|i|}0i$. Starting from $s_\varnothing = k$, set $s_{ub} = T_b(s_u)$ along $c(i)$ and output

$$F_k(i) = Q_\lambda(s_{c(i)}).$$

Evaluation uses $O(|i| + 1)$ PRG calls and outputs $2\lambda$ bits. The same doubling PRG can serve both roles. Prefix-freeness ensures that an output leaf is not later an ancestor of another output leaf. The GGM tree hybrid (Goldreich et al., 1986), followed by a terminal-output hybrid, replaces the values on any polynomially explored set of paths by independent uniform outputs, with repeated inputs receiving the same output. The number of explored nodes is bounded by the total number of input bits queried by the adversary. Accidental collisions among independent internal states, if separated out explicitly, have probability at most a polynomial times $2^{-\lambda}$ and can be charged to the hybrid error. Thus this is a variable-domain PRF with one $\lambda$-bit key, under ordinary uniform security. There is no unit-cost access to superpolynomial-length indices.

## B.2 THE EXACT NEGATION USED IN THE CONVERSE

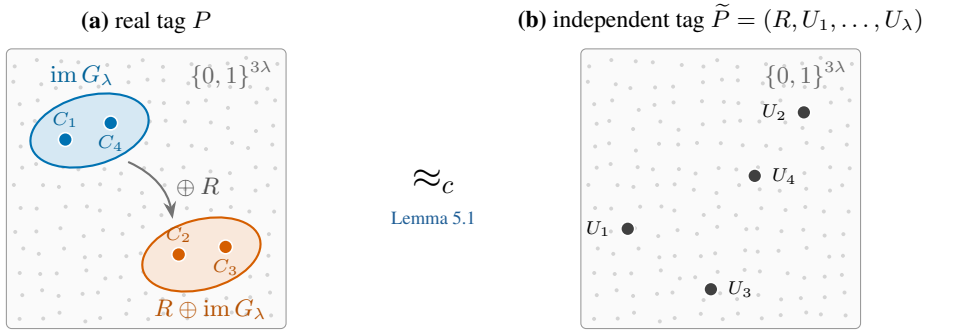
In terms of (7), distributional one-wayness has the following quantifiers:

$$\exists f\ \exists c > 0\ \forall I \in \mathsf{PPT}\ \exists m_0\ \forall m \ge m_0 : \quad \mathrm{err}_{f,I}(m) \ge m^{-c}.$$

The absence of one-way functions, together with the imported equivalence, implies

$$\forall f\ \forall c > 0\ \exists I \in \mathsf{PPT}\ \forall m_0\ \exists m \ge m_0 : \quad \mathrm{err}_{f,I}(m) < m^{-c}.$$

Fix $f$ and $c$. The resulting single inverter has error below $m^{-c}$ at infinitely many lengths $m$. Neither the inverter nor its polynomial running-time exponent is chosen separately for each length. No




$k_j = 0$ puts $C_j$ in $\operatorname{im} G_\lambda$ and $k_j = 1$ in its shift (here $k = 0110$). The two sets meet only if $R \in \Delta_\lambda$ (probability $\leq 2^{-\lambda}$), so exhaustive search over seeds reads off $k$.

Uniform strings, independent of $k$. No efficient test tells the real tag from this one, even when given $k$ and $R$ and free to generate key-dependent data.

Figure 5: The tag of Section 5 is binding for unbounded observers and hiding for efficient ones. (a) Each block $C_j = G_\lambda(\rho_j) \oplus k_j R$ lies in the image of the length-tripling generator $G_\lambda$ or in that image shifted by the public mask $R$; each set holds at most $2^\lambda$ of the $2^{3\lambda}$ strings. (b) Replacing every block by an independent uniform string changes the behaviour of any efficient test only negligibly. The first picture gives statistical identifiability; the second, combined with pseudorandomness of $F_k$, gives hardness.

all-sufficiently-large-length algorithm follows by reversing these quantifiers. Lemma 4.1 is designed for exactly this weaker conclusion.

## C Expanded-opening tags and uniform hybrids

This appendix gives the reduction interface for Lemma 5.1; Figure 5 pictures the two halves of the lemma. Consider any uniform PPT experiment $\mathsf{T}$ that samples $k, R$ with the indicated laws, forms a tag, generates arbitrary additional data using $k, R, P$ and fresh coins, and returns a bit. Its computation outside tag formation must not read the secret seed $z$ or the openings. The bounded experiment in the proof of Lemma 5.4 is such an experiment: the key is known to the tag-hiding simulator, and neither the source nor membership in the virtual language uses an opening.

**First hybrid: expand the openings.** On input a challenge $W \in \{0,1\}^{\lambda^2}$, an $E$-distinguisher samples $k, R$, splits $W$ into $\lambda$ openings, computes (10), and runs $\mathsf{T}$ with the resulting data. It can check any predicate computable from $k$ and the transcript. If $W = E_\lambda(z)$, this is the compact-description experiment. If $W$ is uniform, the openings are independent uniform seeds. No seed inversion or rejection sampling occurs. Therefore the two acceptance probabilities differ negligibly.

**Second hybrid: hide every key bit.** Let $p_j$ be the acceptance probability with the first $j$ values $G_\lambda(\rho_i)$ replaced by independent uniforms and the remaining ones unchanged, always retaining the xor with $k_i R$. Thus $p_0$ is the independent-opening experiment and $p_\lambda$ the uniform-mask endpoint. A single $G$-distinguisher draws $b = \lceil \log_2 \lambda \rceil$ bits. If their integer value is at least $\lambda$, it outputs zero. Otherwise it selects the corresponding coordinate $j$, makes all earlier values uniform, generates all later seeds itself, places its challenge at $j$, and runs $\mathsf{T}$. Its signed difference in acceptance probabilities is

$$2^{-b} \sum_{j=1}^{\lambda} (p_{j-1} - p_j) = 2^{-b}(p_0 - p_\lambda).$$

Taking absolute values, PRG security bounds $|p_0 - p_\lambda|$ by $2^b$ times a negligible function, which is negligible because $2^b < 2\lambda$. The distinguisher samples and knows $k$ even though its input challenge seed is unknown.

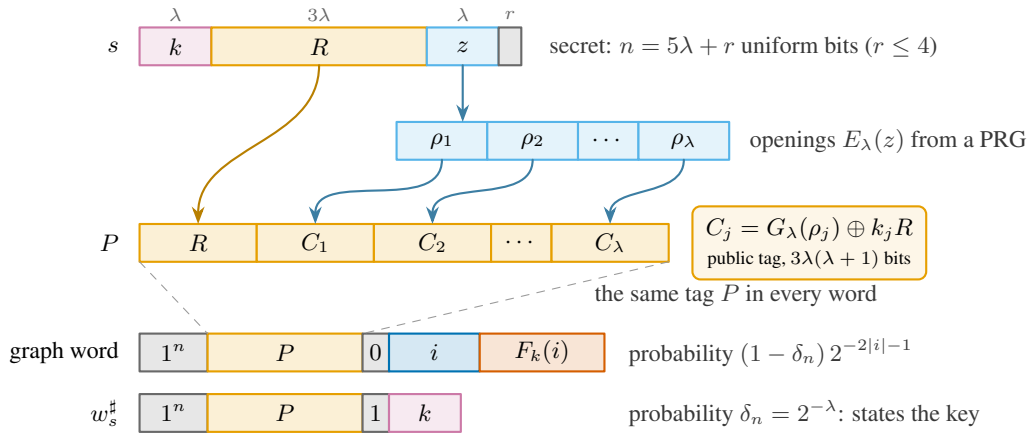



Figure 6: Anatomy of the witness $\mathcal{C}_{\mathrm{tag}} = \mathcal{G}^\sharp$ (Definition 5.2). The secret is only $5\lambda$ bits, yet every word carries the public tag $P$, whose blocks commit to the key bits with openings expanded from $z$. A graph word is a tagged point of the PRF graph, and the rare revealing word states the key; the language depends only on $(n, P, k)$.

At the final endpoint the pre-xor values are independent uniform $3\lambda$-bit strings. Conditional on $k, R$, xoring by $k_j R$ preserves that joint uniform law. The resulting $P = (R, U_1, \ldots, U_\lambda)$ is independent of $k$, as required by the later reduction. The two hybrid errors add. The public tag is not required to be openable at intermediate endpoints: the virtual language (11) is defined for every bit string $P$ of the correct length.

**Why expanding randomness preserves statistical binding.** The bad set $\Delta_\lambda$ depends only on the image of $G_\lambda$, not the law of the opening vector. If $R \notin \Delta_\lambda$, no coordinate admits both a zero opening and a one opening even when every possible seed is allowed. Restricting to opening vectors in the image of $E_\lambda$ cannot create a second key. Thus binding holds pointwise for arbitrary correlations, including those of the compressed description. Hiding uses the pseudorandomness of $E$; binding does not. This is the reason the seed-expansion step can shorten descriptions without changing the posterior bound, and the reason binding holds for *every* length-tripling map $G_\lambda$, secure or not.

## D RIGIDIFICATION: PRESENTATION, CONFIDENCE, AND TWO GENERATORS

Figure 6 shows the anatomy of the witness. Fix a keyed family $\mathcal{K}$ with algorithms $\mathrm{Mem}_\mathcal{K}, \mathsf{Enum}_\mathcal{K}, \mathsf{Smp}_\mathcal{K}$, pathwise bound $b_\mathcal{K}$ and cutoff $\kappa_\mathcal{K}$. Fix $n \ge 5$, set $\lambda = \lfloor n/5 \rfloor$, and parse $s = (k, R, z, \mathrm{pad})$. The lengths are $\lambda, 3\lambda, \lambda, n - 5\lambda$. Every sufficiently large description length is covered; for a uniform description the active bits are independent and uniform. Descriptions with different padding or seeds may give the same semantic pair $(P, k)$. The family and posterior always keep this multiplicity.

**Presentation.** To decide membership given $s$, compute $E_\lambda(z)$ and the $\lambda$ values $G_\lambda(\rho_j)$ forming $P$, and parse the candidate word canonically. Reject an incorrect length field, tag, arity, or branch marker. For a word $\langle 1^n, P, 0, w\rangle$ run $\mathrm{Mem}_\mathcal{K}(k, w)$; for the revealing word compare its key field with $k$. This takes time polynomial in $n + |w|$. Set $\mathsf{Enum}^\sharp(s, 1) = w_s^\sharp$ and $\mathsf{Enum}^\sharp(s, j+1) = \langle 1^n, P, 0, \mathsf{Enum}_\mathcal{K}(k, j)\rangle$; the branches are disjoint and the encoding is injective, so the enumeration is duplicate-free and inherits the time bound of $\mathsf{Enum}_\mathcal{K}$ up to the $O(\lambda^2)$ cost of the tag. The sampler chooses the revealing branch when $\lambda$ fair coins are all zero, an event of probability exactly $2^{-\lambda}$, and otherwise wraps a run of $\mathsf{Smp}_\mathcal{K}(k)$. A terminating run uses at most $\lambda + b_\mathcal{K}(\lambda + |w|)$ coins and polynomial time in $n + |w|$, and its output has length at most $c(n^2 + |w|)$ for a fixed $c$; conversely the encoded word reveals $n$, so a nondecreasing polynomial $b^\sharp$ bounds every terminating run in terms of $n + |w^\sharp|$. A cutoff $\kappa^\sharp(n, q) = c(n^2 + \kappa_\mathcal{K}(\lambda, q))$ has tail at most $1/q$. At lengths $1 \le n < 5$, take $L_s = \{\langle 1^n, i\rangle : i \in \{0,1\}^*\}$ for every $s$, with the geometric index law below; its posterior error is zero.

**Confidence for the actual compressed prior.** On any transcript of positive probability, every posterior-supported description must match its $n$ and $P$. Off $R \in \Delta_\lambda$ the key is unique by Lemma 5.1. Any revealing sample likewise fixes the key even on that event. In either case all posterior-supported descriptions define the same language, irrespective of the multiplicity of seeds and padding and irrespective of the keyed family. Consequently

$$\{L^\sharp_{S^*} \neq L^\sharp_S\} \subseteq \{R \in \Delta_\lambda\} \cap \{\text{no revealing sample among the first } N\}.$$

The last event has probability $(1-\delta_n)^N$ conditional on every description, and $R$ is uniform in the description prior. This proves the first bound in (12). The inequality $xe^{-x} \leq 1/e$ for $x \geq 0$ proves the second. This calculation holds for every integer $N \geq 1$ and length $n$, not just for a fixed confidence as $n$ tends to infinity.

**The witness.** For the pseudorandom-function graph $\mathcal{G}$, let $i_1, i_2, \ldots$ enumerate $\{0,1\}^*$ in length-lexicographic order, including the empty string, and set $\mathsf{Enum}_\mathcal{G}(k, j) = \langle i_j, F_k(i_j)\rangle$. The $j$th index has length $O(\log(j+1))$ and is computable in time polynomial in $j$. The sampler reads fair coins until the first zero, lets $\ell$ be the number of preceding ones, so that $\mathbb{P}[\ell = r] = 2^{-r-1}$, and reads $\ell$ new bits as the index; each graph word with index length $r$ has mass $2^{-2r-1}$ under $D_k$, so every word of $L_k$ has positive mass. A terminating run uses $2\ell+1$ random bits plus polynomial time in $\lambda + \ell$, and its output has length $O(\lambda+\ell)$, so $\kappa_\mathcal{G}(\lambda, q) = O(\lambda + \log q)$ has tail at most $1/q$. The language is infinite because different indices give different encoded words regardless of PRF collisions. In $\mathcal{C}_{\mathrm{tag}}$ a graph word with index $i$ has mass $(1-\delta_n)2^{-2|i|-1}$ and the revealing word has mass $\delta_n$; the lengths of all words are $O(n^2 + |i|)$.

**Finite one-sample generation.** An unbounded algorithm first parses $n, P$. For $n < 5$ it uses the public language. Otherwise, for each coordinate $j$, it searches all $a \in \{0,1\}^\lambda$ and $b \in \{0,1\}$ for $G_\lambda(a) = C_j \oplus bR$, choosing the first admissible $b$. Each coordinate has at least one choice for a genuine tag, because the actual expanded opening is admissible. Off the bad-mask event, each bit is uniquely the actual bit $k_j$. If the first word is revealing, the algorithm can simply use its key instead. Its exhaustive preprocessing is bounded by $2^\lambda \operatorname{poly}(\lambda)$, not by an uncomputable language-equality test. The algorithm freezes its candidate and subsequently checks the graph words at the first $t+1$ indices against $V_t$. There are at most $t$ distinct observed words and the candidate graph words are distinct, so at least one is unseen; when the key is correct, that output is also in the target. One event, the correctness of this candidate, certifies all future outputs without a union over infinitely many rounds. The algorithm may err on bad masks, whose prior probability is at most $2^{-\lambda}$.

**Efficient generation in the limit.** The waiting algorithm outputs $\perp$ before the first revealing sample and thereafter uses its key. On any complete adversarial enumeration the revealing word appears at a finite time, after which every output is fresh and valid by the same counting argument. The input-length running-time bound permits parsing each prefix and enumerating at most $t+1$ candidate graph words. Under the stochastic source, the first revelation time is geometric with mean $1/\delta_n$ and exceeds $h$ with probability $(1-\delta_n)^h$, so the waiting algorithm succeeds within $h$ rounds with probability $1-(1-\delta_n)^h \leq h\delta_n$. This exhibits an eventual algorithm without contradicting the polynomial-horizon lower bound. A full-support i.i.d. stream is itself a complete enumeration almost surely: each word has positive mass, so the probability of never seeing it is zero, and a countable union of null events is null. The finite-horizon theorem nevertheless remains tied to the stated source and prior, not every adversarial stream.

## E Bounded simulation and the hardness interface

This appendix proves Lemma 5.4, specializes it to $\mathcal{C}_{\mathrm{tag}}$, and quantifies it for Proposition E.1. The reduction first uses a simulator that knows the key and then an algorithm $B$ that does not, and finally a distinguisher with hidden checks (Figure 7). We spell out running times and the sets of examples each algorithm sees, so that freshness is preserved when the experiments are connected.

**Coupling away revelations.** At each round couple the original source to a source without revelations by using the same Bernoulli revelation choice and the same example of $D_k$ when the choice is not revelation. Until a revelation, the observed streams and the persistent generator state agree. Their

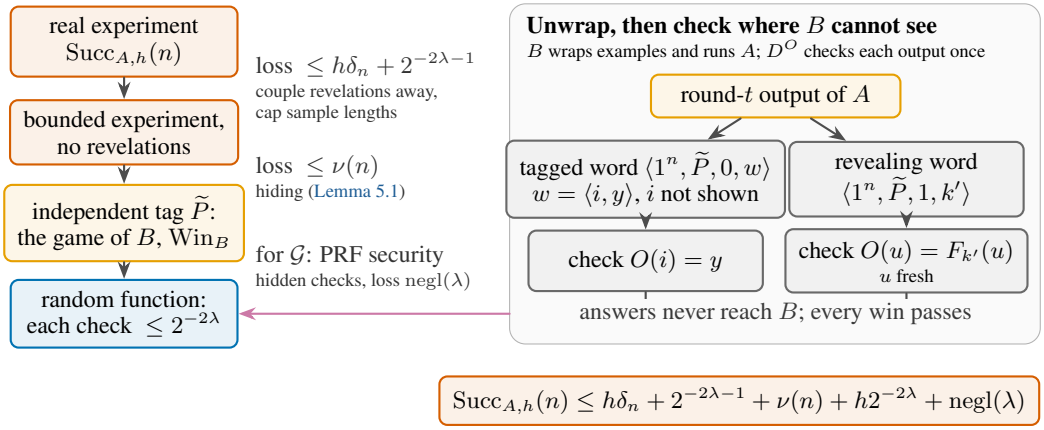



Figure 7: The proof of Lemma 5.4 and Theorem 5.5 as a chain of experiments, with $h = H(n)$. Coupling removes revelations and caps sample lengths; hiding replaces the tag by one that does not depend on the key, after which the experiment is a game that $B$ plays from examples of $L_k$ alone. For the pseudorandom-function graph, a distinguisher checks every candidate with one oracle query whose answer $B$ never sees: every win passes under $F_k$, while under a random function each check passes with probability at most $2^{-2\lambda}$.

horizon-success events can therefore differ only if there was a revelation, with probability at most $h\delta_n$. The coupled experiment still tests membership in the full target language, which contains the revealing word; this retains success from a generator that guesses the key before seeing it.

**A bounded sampler.** Let $\kappa^* = \kappa_{\mathcal{K}}(\lambda, h2^{2\lambda+1})$, which is polynomial in $\lambda + \log h$ by assumption, and replace each sample by $\mathsf{Smp}_{\le\kappa^*}$ as in Section 2: exactly $b_{\mathcal{K}}(\lambda + \kappa^*)$ coins, the same step bound, and $\perp$ unless a word of length at most $\kappa^*$ is produced. Short words keep their exact probabilities, and each sample is $\perp$ with probability at most $2^{-2\lambda-1}/h$, so some sample among the first $h$ is $\perp$ with probability at most $2^{-2\lambda-1}$. The bounded experiment counts any $\perp$ as failure, even when a successful output occurred earlier; the event lost in this way has probability at most $2^{-2\lambda-1}$. For $\mathcal{G}$ one can equivalently read at most $a+1$ stopping coins with $a = 2\lambda + \lceil \log_2 h \rceil$ and abort if all are ones, which retains the probability $2^{-\ell-1}$ of every permitted length. With either implementation the bounded experiment has a fixed polynomial running-time bound beyond the generator's own time, and a simulator that knows $k$ can test success at every round by running $\mathrm{Mem}_{\mathcal{K}}(k, \cdot)$ and comparing key fields. Hence it is a test of the kind covered by Lemma 5.1, and replacing $P$ by $\widetilde{P}$ changes its success probability $p$ to some $p_{\mathrm{hid}}$ with $|p - p_{\mathrm{hid}}| \le \nu(n)$, where $\nu(n)$ is the loss of the reductions for the residue of $n$ modulo 5.

**The algorithm $B$.** Only after both tag hybrids is the tag independent of the key. The algorithm $B$, on input $1^n$, samples $\widetilde{P}$ itself and plays the game of Lemma 5.4: in each round it wraps the new example $w$ as $\langle 1^n, \widetilde{P}, 0, w\rangle$, stops (and fails from then on) if the example is longer than $\kappa^*$, runs the generator on the wrapped examples, and unwraps its output. From $\langle 1^n, \widetilde{P}, 0, w\rangle$ it outputs the candidate word $w$, from $\langle 1^n, \widetilde{P}, 1, k'\rangle$ with $|k'| = \lambda$ it outputs the candidate key $k'$, and otherwise it outputs nothing. A successful output is fresh and valid in the target. In the first case $w \in L_k$ and $w$ differs from every example shown, because the tag is the same in every word; in the second case $k' = k$. Every run of the bounded experiment with $\widetilde{P}$ that succeeds at some round and never aborts is therefore a run in which $B$ wins before it stops, so $p_{\mathrm{hid}} \le \mathrm{Win}_B(n)$. Collecting the coupling, bounding and hiding losses gives (13). No round is preselected, and no output of the generator is ever checked by $B$.

**The witness: hidden checks.** For $\mathcal{G}$, a distinguisher $D^O$ with oracle $O$ runs the game for $B$ without knowing any key. It draws each example index $i$ from the geometric index law, shows $B$ the example $\langle i, O(i)\rangle$, and records the set $Q$ of shown indices. After each round it checks $B$'s candidate with at most one further oracle query, whose answer is used only by $D$: a candidate word $\langle i, y\rangle$ with $i \notin Q$

passes if $O(i) = y$; a candidate key $k' \in \{0,1\}^\lambda$ passes if $O(u) = F_{k'}(u)$, where $u$ is the first index in length-lexicographic order that is neither in $Q$ nor previously checked; any other candidate fails. $D$ outputs one if some check passes within $h$ rounds, and zero if an example is longer than $\kappa^*$ before that.

If $O = F_k$, every winning candidate passes. A word of $L_k$ not yet shown has an index outside $Q$, because the only word of $L_k$ with index $i$ is $\langle i, F_k(i)\rangle$, and it passes by definition; the correct key passes because $F_k(u) = O(u)$. Hence $\mathbb{P}[D^{F_k} = 1] \geq \mathrm{Win}_B(n)$. If $O$ is a uniformly random function, consider the check in round $t$. Its index was never shown to $B$ before round $t$, and the answers to earlier checks are never shown to $B$ at all, so the checked value is uniform and independent of $B$'s view when its candidate is fixed; the check passes with probability at most $2^{-2\lambda}$. A union bound over at most $h$ checks gives $\mathbb{P}[D^R = 1] \leq h2^{-2\lambda}$. $D$ makes at most $|Q| + h$ queries and runs in polynomial time, so PRF security gives

$$\mathrm{Win}_B(n) \leq h2^{-2\lambda} + \mathrm{negl}(\lambda).$$

The later appearance of a checked index as an example does not matter: the check was decided before it appeared.

**Security at all description lengths.** For each of the five fixed residues $r$, the reductions use $n = 5\lambda + r$. Each is a single uniform polynomial-time algorithm. There are finitely many resulting negligible functions, so their maximum is negligible. Since $\lambda = \lfloor n/5 \rfloor$, the same asymptotic property holds for every sufficiently large $n$. No reduction is given advice about a bad length, and no negligible bound is summed over an uncontrolled family of length-dependent machines.

### E.1 Sharpness under exponential security

The bound (14) is explicit enough to be quantified. Its first term, $h\delta_n$, bounds the success probability $1 - (1 - \delta_n)^h$ of the waiting strategy; the others are governed by the security of the primitives.

**Proposition E.1** (Sharpness). *Suppose that $E_\lambda$, $G_\lambda$ and $F$ are $s(\lambda)$-secure: every algorithm running in time at most $s(\lambda)$ has distinguishing advantage at most $1/s(\lambda)$, for all large $\lambda$. There is a polynomial $p$ with the following property. For every generator and every horizon $h = h(\lambda)$ computable in time $s(\lambda)$ such that the first $h$ rounds take total time $T$ with $(T + h^2)\, p(\lambda) \leq s(\lambda)$, for all sufficiently large $\lambda$ and $n = 5\lambda + r$,*

$$\mathrm{Succ}_{A,h}(n) \leq h\delta_n + 2^{-2\lambda-1} + h2^{-2\lambda} + \frac{2\lambda + 2}{s(\lambda)}.$$

*In particular, if $s(\lambda) \geq 2^{c\lambda}$ for a constant $c \in (0, 1/2]$, every generator running for total time at most $2^{c\lambda/3}$ succeeds with probability $2^{-\Omega(\lambda)}$.*

Security beyond $2^{\lambda/2}$ is impossible for a generator with a $\lambda$-bit seed, which is why $c \leq 1/2$.

*Proof.* Each of the three reductions behind Theorem 5.5 (the $E_\lambda$ and $G_\lambda$ hybrids of Section C and the hidden-check distinguisher) runs the generator once for at most $h$ rounds and adds three kinds of cost: sampling examples with the cap $a = 2\lambda + \lceil \log_2 h \rceil$ and forming the tag once, which is polynomial in $\lambda$ per round; writing the prefix $X_{1:t}$ for the $t$th invocation of a stateless generator, which is polynomial in $\lambda$ times $t$ and hence at most $h^2 \,\mathrm{poly}(\lambda)$ in total; and testing each output for membership and freshness, or checking it with one oracle query, which is polynomial in $\lambda$ and the output length, and hence at most $T \,\mathrm{poly}(\lambda)$ in total. For a suitable polynomial $p$ the running time is therefore at most $(T + h^2)\, p(\lambda) \leq s(\lambda)$, and each advantage is at most $1/s(\lambda)$. The expander reduction therefore costs $1/s(\lambda)$, the generator reduction $2^{\lceil \log_2 \lambda \rceil}/s(\lambda) \leq 2\lambda/s(\lambda)$, and the hidden-check distinguisher gives $\mathrm{Win}_B(n) \leq h2^{-2\lambda} + 1/s(\lambda)$. Substituting into (13) gives the displayed bound. When $s(\lambda) \geq 2^{c\lambda}$ with $c \leq 1/2$ and $T \leq 2^{c\lambda/3}$, we have $h \leq T$, the condition $(T + h^2)\, p(\lambda) \leq s(\lambda)$ holds for large $\lambda$, and the bound is at most $2^{-(1-c/3)\lambda} + 2^{-2\lambda-1} + 2^{c\lambda/3-2\lambda} + (2\lambda + 2)2^{-c\lambda} = 2^{-\Omega(\lambda)}$. For the matching upper bound, a generator that has seen one graph example $\langle i, F_k(i)\rangle$ can test candidate keys $k'$ by comparing $F_{k'}(i)$ with $F_k(i)$ and enumerate the graph of the first key that passes; in total time $T$ it tests about $T/\,\mathrm{poly}(\lambda)$ keys, and the true key is among them with probability about $T2^{-\lambda}/\,\mathrm{poly}(\lambda)$. The proposition is therefore tight up to the constant in the exponent. □

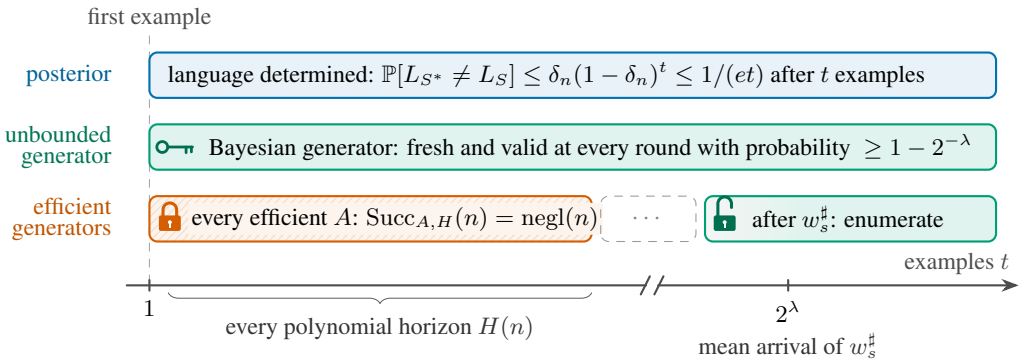



Figure 8: Three time scales in the witness $\mathcal{C}_{\mathrm{tag}}$, with $\lambda = \lfloor n/5 \rfloor$ and a schematic time axis. One example already determines the language except with probability $2^{-\lambda}$, so an unbounded generator can be correct from the first round on. Every efficient generator nevertheless has negligible probability of producing even one fresh valid word within any polynomial horizon (Theorem 5.5). The waiting strategy catches up after the revealing word $w_s^\sharp$, whose mean arrival time is $2^\lambda$; a generator whose running time grows with its input may catch up earlier, but only after superpolynomially many examples. Under exponentially secure primitives, the barrier holds for total running time exponential in $n$ (Proposition E.1).

One example thus suffices for an unbounded generator running in time $2^\lambda \,\mathrm{poly}(\lambda)$, and a generator that waits for the revealing word needs about $2^{\lfloor n/5 \rfloor}$ examples (Figure 8). Because efficiency is measured in the accumulated input length, an efficient generator may instead wait until its budget covers exhaustive key search, which happens after about $2^{\lambda/k}$ examples when its running time grows like the $k$th power of its input. Under exponential security Proposition E.1 shows that such a generator needs $2^{\Omega(\lambda/k)}$ examples, so each efficient generator needs exponentially many, but no single exponent serves all of them; under polynomial security Theorem 5.5 gives superpolynomially many.

# F Fibres, simulation, and the Bayesian generator

We expand Lemma 2.2 and Theorem 3.1, including the short-output exactness and the probability spaces used in the joint bound. The source may have description-dependent tails.

## F.1 Exact short-output probabilities

Fix $n, N, \kappa$ and a simulation bound $b_\kappa = b(n+\kappa)$, enlarging a fixed implementation constant if needed. Every terminating original run producing a word of length at most $\kappa$ uses at most this time and this number of coins. Each such run has a finite stopping prefix on the coin tape, and its probability is exactly preserved by extending that prefix to every $b_\kappa$-bit string with the same prefix. Conversely, a short word returned by the bounded simulator follows a genuine terminating path of the original source. Hence

$$\mathbb{P}_r[\mathsf{Smp}_{\le\kappa}(s;r) = w] = D_s(w) \qquad (|w| \le \kappa). \tag{15}$$

Original nontermination has probability zero, so under the shared-tape coupling the censored output is the original word when short and $\perp$ when long, almost surely. In particular, short-word probabilities are dyadic with denominator dividing $2^{b_\kappa}$. This is a consequence of the pathwise bounded sampling assumption; we neither round source probabilities nor posit an efficient likelihood oracle. For $N$ independent coin blocks, the preimage count at a short transcript $z$ and description $s$ is

$$\#\{r : \Phi_n(s,r) = z\} = 2^{Nb_\kappa} \prod_{j=1}^{N} D_s(z_j), \tag{16}$$

and dividing by $2^{d(n)} = 2^{n+Nb_\kappa}$ gives $\mathbb{P}[S = s, \Phi_n(U) = z] = 2^{-n} \prod_j D_s(z_j) = \mathbb{P}[S = s, X = z]$. This exact integer identity is the connection between uniform inversion and the full-likelihood posterior.

### F.2 KEEPING THE HIDDEN TARGET IN THE ERROR BOUND

Embed every malformed output and $\perp$ into a common countable output space. The projection of a recovered preimage first checks its full length $d(n)$ and then returns its first $n$ bits, or $\perp$ upon failure. Write $K_z = \mathcal{L}(S \mid Z = z)$ for $Z = \Phi_n(U)$, and $Q_z$ for the law of the projected inverter output on input $z$. Data processing applied to (6) gives

$$\mathrm{TV}\big((Z,S),(Z,\text{projected inverse})\big) = \sum_z \mathbb{P}[Z=z]\,\mathrm{TV}(K_z,Q_z) \le \eta_n.$$

The first $S$ here is the actual description used to generate $Z$; this is why an event involving both the recovered and the *true* description is controlled, rather than just the marginal law of a candidate.

Now consider the generator's experiment. Given the prefix $x = X_{1:N_0}$, the posterior-sampling generator draws $T = (T_1, \dots, T_q)$ from $K_x^{\otimes q}$, independently of $S$ and of every later example given $x$; the simulated generator draws $T$ from $Q_x^{\otimes q}$ when $x$ is short and sets $T = (\perp, \dots, \perp)$ otherwise, again independently of everything else given $x$. The remaining randomness, namely the generator's tape and the examples after round $N_0$, has the same conditional law given $(S, x, T)$ in both experiments; if later examples are chosen adaptively, the adversary's coins are also independent of $T$ given $(S, x)$. The two trajectories are therefore images of $(S, X_{1:N_0}, T)$ under the same kernel, except that on a prefix that is not short the simulated generator outputs $\perp$ while the original keeps running; since the trajectory records the prefix, these prefixes are charged at most their probability $\tau_n$ in total. On short prefixes, data processing reduces the claim to

$$\mathrm{TV}\big((S, X_{1:N_0}, T_{\mathrm{post}}),(S, X_{1:N_0}, T_{\mathrm{inv}})\big) = \sum_x \mathbb{P}[X_{1:N_0}=x]\ \mathrm{TV}\big(K_x^{\otimes q}, \mathcal{L}(T_{\mathrm{inv}} \mid x)\big),$$

where conditional independence of $T$ and $S$ given $x$ was used together with $\mathrm{TV}(\mu \otimes P, \mu \otimes Q) = \mathrm{TV}(P, Q)$. For short $x$, (16) identifies $\mathbb{P}[X_{1:N_0} = x]$ with $\mathbb{P}[Z = x]$ and $K_x$ with the posterior, and a hybrid over the $q$ coordinates gives $\mathrm{TV}(K_x^{\otimes q}, Q_x^{\otimes q}) \le q\,\mathrm{TV}(K_x, Q_x)$. Summing over short $x$ yields at most $q\eta_n$. Transcripts that are not short have total probability $\tau_n$ under the original experiment and contribute at most $\tau_n$. This proves Theorem 3.1. No posterior on an original transcript containing $\perp$ is defined or needed, and the argument does not replace the uniform prior by a prior conditional on short outputs, which would generally differ.

### F.3 FROM A DESCRIPTION TO PERMANENT GENERATION

For one frozen description $u$, the enumeration rule outputs, at round $t$, the first word of $\mathsf{Enum}(u, \cdot)$ outside $V_t$. At most $t$ distinct words belong to $V_t$, and the first $t+1$ enumeration outputs are distinct, so one is absent; computing them and comparing them with the transcript takes time polynomial in $B_t$. If $L_u \subseteq L_S$, that word lies in the target at every round. There is one frozen random decision and one successful event, not an infinite list of independently fallible choices. A stateless implementation can recompute the same candidate from the prefix using a persistent tape; a stateful one can store it. The event $L_u \subseteq L_S$ depends only on the prefix and the inversion coins, so it certifies every future round even if the examples after the prefix are an arbitrary, adaptively chosen sequence of target words. The i.i.d. law is used only for the prefix, through (16) and the posterior bound.

For comparison, if $D$ has full support on a countable language, then $D(V_t) \to 1$ almost surely along an i.i.d. stream. Given $\epsilon > 0$, choose a finite set of mass at least $1 - \epsilon$; all of its words eventually appear almost surely, so $\liminf_t D(V_t) \ge 1 - \epsilon$. Applying this to a countable sequence of $\epsilon$ tending to zero shows that a fresh resample from $D$ has conditional probability $1 - D(V_t) \to 0$ of being unseen. Moreover, the empirical distribution $\hat{p}_t$ of $X_{1:t}$ converges to $D$ in total variation almost surely: by the strong law and a countable intersection, $\hat{p}_t(w) \to D(w)$ for every word $w$ simultaneously, almost surely, and since both sum to one, Scheffé's lemma gives $\sum_w |\hat{p}_t(w) - D(w)| \to 0$. Resampling an observed example is therefore an asymptotically exact predictive sampler, and it never produces a fresh word. Enumeration avoids this obstacle. Extrapolating instead a sequence that appends $\mathsf{Enum}(S, 1), \dots, \mathsf{Enum}(S, M)$ to the examples reproduces the Bayesian generator for $M$ rounds, since the extrapolated continuation depends on the target only through the examples; this suffices for any polynomial horizon, and inverting the sampling process, coins included, yields the description itself and with it permanent correctness.

### F.4 The Bayesian generator as the exact case

This subsection proves Proposition 2.4. Fix $N$ and let $S^*$ be drawn from $\pi_n(\cdot \mid X_{1:N})$ with fresh coins. Conditional on $(S, X_{1:N}, S^*)$, the continuation $X_{N+1}, X_{N+2}, \ldots$ is i.i.d. from $D_S$, because $S^*$ depends on the past only through $X_{1:N}$ and independent coins. On the event $L_{S^*} \subseteq L_S$, every output of the Bayesian generator lies in $L_{S^*} \setminus V_t \subseteq L_S \setminus V_t$.

On the complementary event, let $j^*$ be the least index with $\mathsf{Enum}(S^*, j^*) \notin L_S$; it is determined by $(S, S^*)$. Each of the finitely many words $\mathsf{Enum}(S^*, j)$ with $j < j^*$ lies in $L_S$ and has positive $D_S$-mass, so conditionally almost surely each appears in the continuation, and some almost surely finite time $T$ has all of them in $V_T$. For every $t \geq \max(T, N)$, the first word of $\mathsf{Enum}(S^*, \cdot)$ outside $V_t$ is $\mathsf{Enum}(S^*, j^*)$, since $V_t \subseteq L_S$; the output at such a round is invalid. The two events therefore agree up to a null set, which proves the proposition. The same argument shows that a generator which freezes any candidate $u$ and enumerates $L_u$ fails eventually, almost surely, whenever $L_u \not\subseteq L_S$: for this rule, containment is necessary as well as sufficient for permanent success.

If the inverter is exact and no word is censored, Theorem 3.1 gives that $(S, X_{1:N}, \widehat{S})$ has the law of $(S, X_{1:N}, S^*)$, so the simulated Bayesian generator is the Bayesian generator. Under an arbitrary target-valid continuation only the inequality $\mathbb{P}[\text{permanent success}] \geq \mathbb{P}[L_{S^*} \subseteq L_S]$ survives: a continuation that withholds one word of $L_S$ can keep a non-contained candidate from ever failing.

### F.5 Bayes-risk characterization of posterior identifiability

Aggregate posterior mass over descriptions naming the same language: $q_\ell(X) = \mathbb{P}[L_S = \ell \mid X]$.

**Proposition F.1** (Bayes-risk characterization). *Let $\beta_N = \mathbb{P}[L_{S^*} \neq L_S]$ and let $r_N$ be the smallest exact-identification error of any rule, computable or not, that outputs a language after $N$ samples. Then*

$$\beta_N = \mathbb{E}\big[1 - \textstyle\sum_\ell q_\ell(X)^2\big], \qquad r_N = \mathbb{E}\big[1 - \max_\ell q_\ell(X)\big], \qquad r_N \leq \beta_N \leq 2r_N.$$

*Consequently, a presented family belongs to $\mathcal{K}_{\mathrm{pid}}$ exactly when some polynomially bounded, polynomial-time computable $N'(n, p)$ gives $r_{N'(n,p)} \leq 1/p$ for every $n$ and $p \geq 2$.*

*Proof.* Given $X$, $S$ and $S^*$ are independent posterior draws, which gives the formula for $\beta_N$, and a Bayes rule picks a language of largest posterior mass, which gives $r_N$. With $a = \max_\ell q_\ell$, $a^2 \leq \sum_\ell q_\ell^2 \leq a$ and $1 - a^2 \leq 2(1 - a)$. Replacing $p$ by $2p$ in one direction gives the sample-count equivalence. □

## G Padding and collapse

Throughout, assume that no uniform one-way functions exist, fix a presented family and a posterior-sampling generator $A$ with prefix $N_0$ and $q$ draws, and fix a constant $e$ with $q(n) \leq n^e$ for all large $n$.

Figure 9 illustrates the padding.

**Bounded choice.** For a nonempty ordered set $J$ of size $b$, define $\mathsf{Pick}(J)$ by reading exactly $\lceil \log_2 b \rceil$ fair bits. Return the corresponding element when the drawn integer is below $b$, and return $\perp$ otherwise. Each element has probability $2^{-\lceil \log_2 b \rceil} \geq 1/(2b)$. For $b = 1$, no coins are read and its only element is returned. Selection failure always contributes zero success.

**The padded function.** Given a cutoff $\kappa_0(n)$, let $b_0(n) = b(n + \kappa_0(n))$ and $d(n) = n + N_0(n) b_0(n)$, and let $\Phi_n$ be the sampling map for $N_0(n)$ examples with cutoff $\kappa_0(n)$. These are public polynomially bounded computable functions. Choose fixed integers $D \geq 1, n_0 \geq 2$ such that $d(n) \leq n^D$ for $n \geq n_0$. The function $f$ in (8) is well defined at all $m \geq n_0^D$: if $n = \lfloor m^{1/D} \rfloor$ then $n^D \leq m$, so its active input fits. Integer root computation, reading the active bits, running the bounded sampler, and writing the unary length fields all take polynomial time in $m$. At smaller lengths let $f(x) = \langle 1^{|x|}, x \rangle$;

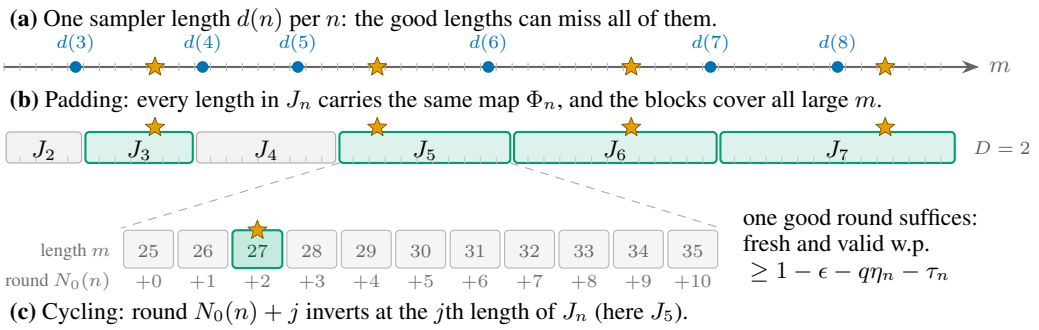



Figure 9: Padding across blocks (Lemma 4.1): the same sampling problem at every input length of a block. Without one-way functions, a fixed inverter has small error $\mathrm{err}_{f,I}(m) < m^{-c}$ at infinitely many input lengths (stars), but it does not reveal which. (a) A sampling map placed at one length $d(n)$ per $n$ can miss all of them; here the $d(n)$ are even and the good lengths odd. (b) Padding places the same sampling problem at every length of $J_n = \{n^D, \ldots, (n+1)^D - 1\}$, and these blocks partition all large lengths, so infinitely many blocks contain a good length. (c) The generator never learns which length is good. Part (i) of Theorem 4.2 picks one at random, losing a factor $1/(2|J_n|)$; part (ii) instead tries them in turn, one per round, so the good round alone succeeds with probability at least $1 - \epsilon - q\eta_n - \tau_n$.

its arity distinguishes this branch. The explicit length field forces any valid preimage of a large output to have the same length $m$, and for a short transcript $z$ and candidate $s$, (16) yields

$$\#\{x \in \{0,1\}^m : x_{1:n} = s,\ f(x) = \langle 1^m, 1^n, z\rangle\} = 2^{m-d(n)} 2^{N_0(n) b_0(n)} \prod_{j=1}^{N_0(n)} D_s(z_j). \tag{17}$$

The factors outside the product are identical for every $s$, so projecting a uniform padded preimage onto its description coordinate gives precisely the original posterior. For a length $m$ with associated $n$, the projected inverter prepends the public length fields to a transcript, runs the inverter for $f$ with $1^m$, rejects an output whose length is not $m$, and otherwise projects to the first $d(n)$ active bits. Data processing turns $\mathrm{err}_{f,I}(m)$ into an upper bound on the error (6) of the projected inverter for $\Phi_n$, and $m \geq n^D \geq n$. One must check the output length before projection, because a short or malformed inverse cannot be silently interpreted as a valid description. Blocks $J_n$ are finite and partition all $m \geq n_0^D$, so if the inverter is good at infinitely many $m$, infinitely many blocks contain a good length. This proves Lemma 4.1.

**Part (i) of the collapse.** Take $\kappa_0(n) = \kappa(n, \lceil 4n^\gamma \rceil N_0(n))$, so that $\tau_n \leq N_0(n) \sup_s \mathbb{P}_{D_s}[|W| > \kappa_0(n)] \leq \frac{1}{4} n^{-\gamma}$, and $c = \gamma + e + 2$. By Section B.2, one uniform PPT inverter $I$ has $\mathrm{err}_{f,I}(m) < m^{-c}$ at infinitely many lengths $m$. At such a length in $J_n$, the projected inverter has error $\eta_n < n^{-c}$, so $q(n)\eta_n \leq n^{-\gamma-2} \leq \frac{1}{4} n^{-\gamma}$ for $n \geq 2$, and Theorem 3.1 gives the branch generator $A^{\mathsf{Inv}}$ at that length success, in the same sense as $A$, with probability at least $n^{-\gamma} - \frac{1}{4}n^{-\gamma} - \frac{1}{4}n^{-\gamma} = \frac{1}{2}n^{-\gamma}$, for all large such $n$. For $n < n_0$ the generator $A'$ outputs $\perp$. For $n \geq n_0$, at round $N_0(n)$ it selects $m = \mathsf{Pick}(J_n)$, independently of the samples, and thereafter runs $A^{\mathsf{Inv}}$ with the projected inverter at length $m$; before round $N_0(n)$ it runs $A$ unchanged. At every such good $n$, the chance of selecting one particular good length is at least $1/(2|J_n|)$, and multiplying by $\frac{1}{2}n^{-\gamma}$ gives (9). There is no assertion that $m$ is computably recognizable as a good length.

**Part (ii) of the collapse.** Take $\kappa_0(n) = \kappa(n, \lceil n^c \rceil N_0(n))$, so that $\tau_n \leq n^{-c}$, and let the inverter be good for exponent $c + e + 1$, so that $q(n)\eta_n \leq n^{-c-1} \leq n^{-c}$ at good lengths. For $0 \leq j < |J_n|$, round $N_0(n) + j$ of $A''$ uses its own segment of the persistent tape to invert the prefix $q(n)$ times at the $j$th length of $J_n$, and outputs the round-$(N_0(n) + j)$ output of $A$ run with these descriptions. If the $j$th length is good and $n$ is large, Theorem 3.1 shows that the simulated run is permanently successful from round $N_0$ with probability at least $1 - \epsilon(n) - 2n^{-c}$, and on that event its output at round $N_0(n) + j$ is fresh and valid. Hence $\mathrm{Succ}_{A'',H''}(n) \geq 1 - \epsilon(n) - 2n^{-c}$ at infinitely many $n$.

**Uniformity and running time.** In both parts $D, n_0, \gamma, c, e$, the family algorithms, the generator $A$, and the fixed inverter are part of one finite machine description; there is no advice encoding the infinite set of good lengths. Every length used is at most $(n+1)^D$, and a single fixed polynomial bounds the inverter's running time there. Each round performs at most $q(n)$ inversions and one invocation of $A$, so each invocation takes time polynomial in the accumulated input length. The lower bound in (i) is inverse polynomial in $n$ because $|J_n| = O(n^{D-1})$, so it rules out the strong gap without asserting almost-everywhere generation with inverse-polynomial success.

**Identification.** The proof of Corollary 6.3 uses part (ii) with $A$ the identifier that outputs its single posterior draw at every round from $N(n,p)$ on. Theorem 3.1 controls the event $\{L_{T_1} = L_S\}$ exactly as it controls permanent success, since it bounds the distance between the joint laws of the target and the draws; cycling through $J_n$ then gives identification success at least $1 - 1/p - 2n^{-c}$ at infinitely many $n$.

## H More on the barrier

**The sampling map need not be one-way.** Remark 4.4 asserts distributional one-wayness only. Mix into a family with a strong gap one extra description, drawn with probability $2^{-n}$, whose language is all of $\{0,1\}^*$ and whose source draws a geometric length and then uniform bits. The posterior probability of the extra description averages to its prior, so the family stays in $\mathcal{K}_{\mathrm{post}}$, and efficient generators gain at most $2^{-n}$, so the gap remains. Yet some preimage of every short transcript is now easy to find: the extra description together with coins that make its source emit the observed words.

**Example H.1** (Beyond the Bayesian generator)**.** At even description lengths, extend $\mathcal{C}_{\mathrm{tag}}$ by a second hidden block $v \in \{0,1\}^n$ and a special word $u_v = \langle 1^n, 2, v\rangle$, listed first by the enumeration and emitted by the source with probability $2^{-n}$. Until $u_v$ appears, the posterior on $v$ is uniform, so a posterior draw names a language containing a wrong special word, and the Bayesian generator outputs that word at every round; it is permanently correct with probability at most $(N+1)2^{-n}$, and the family lies outside $\mathcal{K}_{\mathrm{pid}}$. A posterior-sampling generator that enumerates only the rigidified part of its draw's language is permanently correct after one example with probability at least $1 - 2^{-\lfloor n/5 \rfloor} - 2^{-n}$, so the family lies in $\mathcal{K}_{\mathrm{post}}$. An efficient generator for it yields one for $\mathcal{C}_{\mathrm{tag}}$ by simulating $v$ and $u_v$, since outputting the unseen word $u_v$ requires guessing $v$; under one-way functions the family therefore has a strong gap. Theorem 4.2 covers it, whereas a converse stated through the Bayesian generator alone would not.

**Impagliazzo's worlds.** In the five worlds of Impagliazzo (1995), (a) of Theorem 6.1 holds exactly in Minicrypt and Cryptomania. In Algorithmica, Heuristica and Pessiland, posterior samples can be removed from every generator at infinitely many lengths, even where NP is hard on average (Figure 10).

**Mistakes.** Counting abstentions as mistakes, the strong gap of $\mathcal{C}_{\mathrm{tag}}$ gives every efficient generator $H(1 - \mathrm{negl})$ expected mistakes in the first $H$ rounds, whereas an unbounded generator that decodes the tag and switches to a revealed key makes at most $\mathbb{P}[R \in \Delta_\lambda] \cdot 2^\lambda \le 1$ in expectation, a separation in the spirit of Blum (1994).

**Samplable presentations and quantifiers.** The converse uses the i.i.d. law only through Lemma 2.2 and the length tails that bound $\tau_n$, so Theorem 4.2 should extend to any efficiently samplable presentation of the prefix with pathwise bounds and a tail bound on its total length; we have not written out the details. Prediction of the next example is tied to almost one-way functions (Naor & Rothblum, 2006) and generation to ordinary ones. One expects a dual equivalence in which almost one-way functions correspond to gaps at infinitely many description lengths and the padding disappears, since inverters would then exist at all large lengths; it would rest on infinitely-often versions of Håstad et al. (1999) and Goldreich et al. (1986), which we have not verified.

# I THE HIDDEN-PARITY FAMILY

This appendix verifies the claims of Example 2.6. Fix $n \geq 1$ and a description $a \in \{0,1\}^n$. Words have the form $\langle 1^n, x, b\rangle$ with $x \in \{0,1\}^*$, $|x| \geq n$, and $b = \langle a, x_{1:n}\rangle \in \{0,1\}$, the inner product over $\mathbb{F}_2$ of $a$ with the first $n$ bits of $x$.

**Presentation.** Membership parses the canonical tuple, checks $|x| \geq n$, and compares $b$ with the inner product; this takes time linear in $n + |w|$, every word reveals $n$, and $|w| \geq n$. Let $x_1, x_2, \ldots$ enumerate the words of length at least $n$ in length-lexicographic order and set $\mathsf{Enum}(a, j) = \langle 1^n, x_j, \langle a, (x_j)_{1:n}\rangle\rangle$. The encoding is injective and the $x_j$ are distinct, so the enumeration is duplicate-free; the $j$th word has length $O(n + \log(j+1))$ and is computable in time polynomial in $n + j$. The sampler reads fair coins until the first zero, lets $\ell$ be the number of preceding ones, so that $\mathbb{P}[\ell = r] = 2^{-r-1}$, then reads $n + \ell$ uniform bits as $x$ and outputs the word. A terminating run uses $n + 2\ell + 1$ coins and time polynomial in $n + \ell$, and its output has length $O(n + \ell)$, so a cutoff $\kappa(n, q) = c(n + \lceil \log_2 q \rceil + 1)$ has tail at most $1/q$ for a fixed constant $c$. Every $x$ with $|x| \geq n$ has mass $2^{-(|x|-n)-1}2^{-|x|} > 0$, so the source has full support on $L_a$. Distinct descriptions name distinct languages: if $a_j \neq a'_j$, the word $\langle 1^n, e_j, a_j\rangle$, where $e_j \in \{0,1\}^n$ is the $j$th unit vector, lies in $L_a \setminus L_{a'}$.

**Posterior.** Under description $a$, a transcript $(x^{(1)}, b_1), \ldots, (x^{(N)}, b_N)$ has likelihood

$$\prod_{j=1}^{N} 2^{-2|x^{(j)}|+n-1}\, \mathbf{1}\big[b_j = \langle a, x^{(j)}_{1:n}\rangle\big].$$

The first factor does not depend on $a$, so the posterior (4) is uniform on the solution set

$$V(X) = \{a' \in \{0,1\}^n : \langle a', x^{(j)}_{1:n}\rangle = b_j \text{ for all } j \leq N\},$$

which contains the true $a$ and is a coset of the orthogonal complement of $\mathrm{span}\{x^{(j)}_{1:n}\}$; hence $|V(X)| = 2^{n-\rho}$ where $\rho$ is the rank of the observed prefixes. Given $X$, $S^*$ is uniform on $V(X)$ and independent of $S$, so $\mathbb{P}[S^* \neq S \mid X] = 1 - 2^{-(n-\rho)} \leq \mathbf{1}[\rho < n]$. The prefixes are independent uniform vectors in $\mathbb{F}_2^n$, and for each fixed nonzero $y$ the bits $\langle y, x^{(j)}_{1:n}\rangle$ are independent and fair, so

$$\mathbb{P}[\rho < n] = \mathbb{P}[\exists y \neq 0 : \langle y, x^{(j)}_{1:n}\rangle = 0\ \forall j] \leq (2^n - 1)2^{-N} < 2^{n-N}.$$

Since distinct descriptions name distinct languages, $\mathbb{P}[L_{S^*} \neq L_S] = \mathbb{P}[S^* \neq S] \leq 2^{n-N}$, which is the bound stated in Example 2.6. Conversely, if $N < n$ then $\rho \leq N < n$ on every transcript and the error is at least $1/2$, so identification needs at least $n$ examples. The Bayes-optimal identifier solves the linear system by Gaussian elimination; the family is therefore posterior-identifiable for an elementary, non-cryptographic reason.

**Generation.** Given one example $\langle 1^n, x, b\rangle$, the word $\langle 1^n, x0, b\rangle$ has the same first $n$ bits, hence the same parity, is valid, and differs from every observed word with a different $x$; scanning the extensions $x0, x1, x00, \ldots$ of an observed $x$ finds one outside $V_t$ within $t + 1$ steps. Generation therefore succeeds from the first example on, without identifying $a$. This is consistent with Theorem 6.1: membership in $\mathcal{K}_{\mathrm{pid}}$ is a statistical property, and a strong polynomial-horizon gap within the class requires one-way functions.

# J A PRESENTED COMMON-CORE FAMILY

Let

$$A_n = \{\langle 1^n, 0, i\rangle : i \in \{0,1\}^*\}, \qquad u_{n,s} = \langle 1^n, 1, s\rangle \quad (s \in \{0,1\}^n).$$

For a description $s \in \{0,1\}^n$ set $L_s = A_n \cup \{u_{n,s}\}$. The source samples $u_{n,s}$ with probability $2^{-n}$, implemented by $n$ zero coins, and otherwise samples an index of $A_n$ by the geometric law, independently of $s$. The branch marker makes the core and special words disjoint, and distinct descriptions have distinct special words, so distinct descriptions name incomparable languages.

Membership, duplicate-free enumeration, length revelation, output-sensitive sampling, and uniform length tails all have the same elementary implementations as in Section D, with no cryptographic operations. A generator reads $n$ and scans the first $t+1$ core words for one outside $V_t$. It succeeds on every positive stream from every target, beginning after the first example, so the family lies in $\mathcal{K}_{\text{post}}$ with $q = 0$. Completeness of the stream is not required.

Conditional on seeing no special word in $N$ samples, every core transcript has the same likelihood under every description, so the posterior is uniform on $\{0,1\}^n$ and an independent posterior draw equals $S$ with probability $2^{-n}$. If a special word occurs, it names $S$ and every posterior draw equals $S$. Because the languages are incomparable, containment and equality coincide, and

$$\mathbb{P}[L_{S^*} \not\subseteq L_S] = \mathbb{P}[L_{S^*} \neq L_S] = (1-2^{-n})^N(1-2^{-n}) = (1-2^{-n})^{N+1}.$$

For any polynomially bounded $N$ the containment probability is therefore at most $(N+1)2^{-n}$, so the family lies outside $\mathcal{K}_{\text{pid}}$, and by Proposition 2.4 its Bayesian generator is permanently correct with probability at most $(N+1)2^{-n}$, despite the trivial generation algorithm. A premise stated through the Bayesian generator alone would exclude this family; the premise $\mathcal{K}_{\text{post}}$, stated through posterior samples and efficient post-processing, includes it.

**Missing information.** For Example 2.8, let $c_n = \langle 1^n \rangle$ and $L_s = \{c_n\} \cup \{\langle 1^n, s, i\rangle : i \in \{0,1\}^*\}$. The source reads $n$ fair coins, emits $c_n$ unless all are zero, and otherwise emits $\langle 1^n, s, i\rangle$ with $i$ drawn from the geometric index law; membership, enumeration ($c_n$ first), pathwise bounds and length tails are elementary. Let $G$ be the event that $X_1 = \cdots = X_H = c_n$, of probability at least $1 - H2^{-n}$ under every description. Conditional on $G$, the target $S$ is still uniform and the view of any generator is independent of it. On $G$ the word $c_n$ lies in $V_t$ for every $t \geq 1$, so a fresh valid output at round $t$ must be $\langle 1^n, S, i\rangle$ for some $i$, which the generator produces with probability at most $2^{-n}$. A union bound over $H$ rounds gives $\text{Succ}_{A,H}(n) \leq 2H2^{-n}$ for every generator, efficient or not. The same argument applies to a posterior-sampling generator: on $G$ its posterior draws are uniform on $\{0,1\}^n$ and conditionally independent of $S$, so its whole view, draws included, is independent of $S$, and its success is also at most $2H2^{-n}$. The family therefore lies outside $\mathcal{K}_{\text{post}}$.

## K Canonical encoding and probability spaces

For any binary word $v$, let $\text{code}(v) = 1^{|v|}0v$. This code is injective and prefix-free, with length $2|v|+1$. Encode an $r$-tuple by

$$\langle v_1, \ldots, v_r\rangle = 1^r 0\, \text{code}(v_1) \cdots \text{code}(v_r).$$

The arity and component lengths are determined uniquely, parsing is linear-time, and total length is linear in $r + \sum_j |v_j|$. Each length parameter $1^n$ is itself a component, so every valid word has length at least $n$. A tag $P$ may be a fixed-length concatenation of its $R, C_j$ blocks once $n$ determines $\lambda$, or a nested tuple with only a constant-factor change in length. Branch markers are one-bit strings. Censored outputs use a disjoint leading flag for $\perp$ before transcript encoding. There are no alternative padded encodings of one semantic word.

The tag-hiding experiment, the actual source experiment, and the experiment of the algorithm $B$ have distinct roles. The first samples and knows $k$ in order to generate examples and evaluate a success predicate. The second hides $s$ from the generator but samples it from the specified prior. The third shows $B$ examples of $L_k$ with an independent public tag and does not reveal $k$. The proof passes between these experiments by coupling and computational indistinguishability, not by giving the generator the verifier's private information.

In every statement involving $S^*$, the target $S$ is drawn first from the description prior, samples are drawn conditional on it, and $S^*$ is then drawn independently from the conditional posterior. Conditional independence is only claimed after conditioning on the transcript, not unconditionally. Inversion coins are fresh and independent of the actual sampler coins. A persistent generator tape ensures that one set of candidates and one chosen padding branch are used for all future rounds.

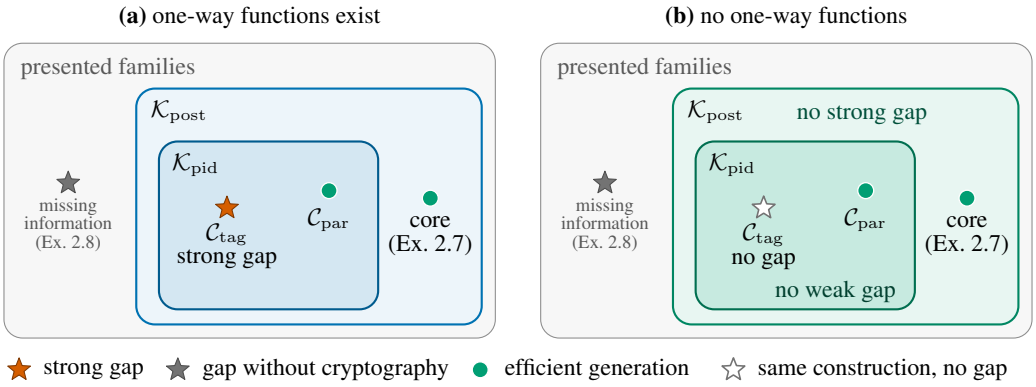



Figure 10: The dichotomy of Theorem 6.1. (a) With one-way functions, the witness $\mathcal{C}_{\text{tag}}$ lies in $\mathcal{K}_{\text{pid}}$ and has a strong gap, although one example identifies its language. Generation can be easy, unconditionally, inside $\mathcal{K}_{\text{pid}}$, as for hidden parities (Example 2.6), and outside it, as for the core family of Example 2.7, which lies in $\mathcal{K}_{\text{post}}$ because efficient generators are posterior-sampling generators with no draws. Outside $\mathcal{K}_{\text{post}}$, a gap can hold for purely statistical reasons (Example 2.8). (b) Without one-way functions, posterior samples can be removed from every generator at infinitely many lengths (Theorem 4.2): no family in $\mathcal{K}_{\text{post}}$ has a strong gap, and no family in $\mathcal{K}_{\text{pid}}$ has even a weak gap. The construction of $\mathcal{C}_{\text{tag}}$, with any efficiently computable candidate primitives, still lies in $\mathcal{K}_{\text{pid}}$, because binding uses only the size of an image (Lemma 5.1).

## L Comparison with computational generation and impersonation

Table 2 separates the tasks, statistical premises, and hardness criteria of the closest results, and Figure 10 places the examples of the paper in the two worlds of Theorem 6.1. This distinction matters for interpreting an equivalence: changing a success criterion or a length quantifier changes what must be proved in the converse.

Table 2: Related computational barriers. The row for Naor & Rothblum (2006) records the *almost* one-way functions used in their formal learnability equivalence.

| | Naor & Rothblum (2006) | Jimenez et al. (2026) | Li et al. (2026) | This paper |
|---|---|---|---|---|
| Task | approximate the next output distribution | mistake-bounded generation on finite domains | mistake-bounded generation of infinite languages | one fresh valid word from an infinite language |
| Presentation | efficiently generated public/secret process | adversarial enumeration | complete distinct enumeration | uniform hidden description; i.i.d. full-support source |
| Statistical premise | no exact-identification requirement | no exact-identification requirement | closure dimension zero; unbounded zero-mistake generation | Bayes risk $\leq 1/p$ after $\text{poly}(n, p)$ samples for the witness; generation with posterior samples for the converse |
| Assumption | almost one-way functions | strong RSA | random oracle | ordinary one-way functions |
| Direction | equivalence | hardness | separation | equivalence |
| Barrier | error in approximating a distribution | no polynomial-time polynomial-mistake guarantee | exponential expected mistakes on some target | negligible probability of any success in every polynomial horizon |

**What inversion supplies.** Naor & Rothblum (2006) use distributional inversion to reconstruct hidden state for approximate impersonation. Their Section 2.1 distinguishes ordinary one-way functions, hard at all sufficiently large lengths, from almost one-way functions, hard infinitely often;

their Theorem 4.2 uses the latter. Our converse starts from the absence of ordinary one-way functions and retains the resulting infinitely-often guarantee. The two quantifier arrangements are dual, and neither equivalence implies the other. Distributional inversion also supports learning reductions in other settings (Binnendyk et al., 2022). Here its output must support unseen words, and a single containment event certifies all future outputs of one frozen enumerator.

**Posterior sampling as a resource.** Two ingredients of our converse are classical, and we claim neither. First, a uniform preimage of a sampler, coins included, is a draw from the conditional law of its hidden inputs: the proof of Theorem 4.2 of Naor & Rothblum (2006) inverts a sampler on its public outputs for exactly this purpose, and the same step appears in Bogdanov & Trevisan (2006, Section 5.2), in Hirahara & Nanashima (2023, Section 8.1), and informally in Impagliazzo (1995). Second, without one-way functions every samplable joint distribution admits efficient approximate conditional sampling at the same, infinitely many, input lengths (Ostrovsky & Wigderson, 1993, Section 2.5); Hirahara & Nanashima (2023) give a full proof through distributional inversion, assuming that no infinitely-often one-way function exists in order to obtain all lengths. What is specific to generation is the rest: the success event compares recovered descriptions with the hidden target inside a running generator, so we need the $q$-fold product of conditional samples composed with the generator's kernel and continuation (Theorem 3.1); fresh words require enumeration rather than resampling; the guarantee must reach infinitely many description lengths of a presented family, not merely infinitely many input lengths of one map (Lemma 4.1); and on the hard side the posterior must be a point mass after one example. Voitovych et al. (2026) study conditional-sampling oracles in online learning and prove statistical and oracle-efficiency separations, not a cryptographic equivalence. The template "statistically far but computationally close ensembles exist if and only if one-way functions do" is due to Goldreich (1990), and Naor & Rothblum (2006, Theorem 4.1) prove a sharper version by inverting a sampler of a hidden bit; our theorem is the analogue for fresh generation.

**What universal extrapolation supplies.** Impagliazzo & Levin (1990) show that, without one-way functions, samplable distributions admit universal extrapolation: given a prefix of samples, one can efficiently sample a continuation close to its true conditional law. Hirahara & Nanashima (2023) use it to make agnostic learning, learning adaptively changing distributions, and several other tasks efficiently solvable in this regime, and Blum et al. (1993) conversely derive cryptographic primitives from average-case hardness of learning. Applied to the example stream, extrapolation is predictive inference and does not help: under an i.i.d. source, a continuation drawn close to the source repeats an observed word with probability tending to one, whereas a fresh output must be valid and unseen. Applied instead to a sequence that appends $\mathsf{Enum}(S, 1), \ldots, \mathsf{Enum}(S, M)$ to the examples, it reproduces the Bayesian generator for $M$ rounds, since the extrapolated continuation depends on the target only through the examples. Our converse inverts the sampler, coins included, which yields a description serving every round at once, and the padding step places the guarantee at the right lengths. On the hard side, the witness must also stay statistically identifiable, which hardness of prediction does not provide by itself.

**Computational versus statistical sample complexity.** Decatur et al. (1999) show that, under cryptographic assumptions, some concept classes are learnable from few examples without computational limits but need many more examples in polynomial time; our gap is the generation analogue, with one example sufficing statistically. Gupta et al. (2024) show that one-way functions make posterior sampling for diffusion models intractable even when unconditional sampling is easy; the posterior-sampling generators of Definition 2.3 make posterior sampling the explicit resource, and our witness gives a generation instance in which one example already determines the posterior. Arenas et al. (2026) show that generation in the limit can require prohibitively many samples for regular and context-free languages, a statistical barrier complementary to our computational one. Finally, Blum (1994) uses one-way functions to separate PAC learning from mistake-bounded learning; counting abstentions as mistakes, the strong gap likewise gives efficient generators $H(1 - \mathrm{negl})$ expected mistakes in $H$ rounds, against at most one for an unbounded generator (Section H).

**Regular families.** Suppose each description names a regular language through an automaton with at most $q(n) = \mathrm{poly}(n)$ states. Any example $w$ with $|w| \geq q(n)$ can be pumped: some split $w = xyz$ with $|xy| \leq q(n)$ and $|y| \geq 1$ has $xy^iz \in L_S$ for every $i \geq 0$. A generator that has seen such a $w$ can try its $O(|w|^2)$ splits in turn, one per round, outputting $xy^iz$ with $i$ large enough that the word

is longer than every example; at the round of a correct split the output is fresh and valid, and each round takes polynomial time. Hence a strong gap for such a family requires that examples longer than $q(n)$ appear only with negligible probability within polynomial horizons. Whether natural regular or context-free families, with sources of this kind, can exhibit the barrier remains open.

**What the generation lower bounds supply.** Jimenez et al. (2026) give polynomial-time algorithms for several finite-domain classes and, under strong RSA, rule out polynomial-time mistake-bounded generation for disjunctions of three literals. Li et al. (2026) obtain their zero-versus-exponential separation relative to a random oracle and discuss sufficient conditions for a standard-model counterpart. Our construction addresses the related statistical-versus-computational question with ordinary primitives, but uses an average-case stochastic experiment and permits a small statistical failure probability. It does not establish their zero-mistake or exponential-mistake claims in the standard model.