

Limitations on autoregressive token generation via sequential sampling

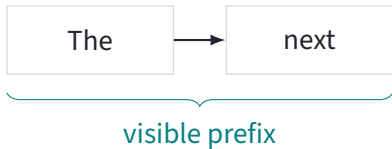
What can a next-token generator sample?

Gal Arnon* Andrej Bogdanov[‡] Paul Lou[†]
Giulio Malavolta[†] Alon Rosen[†] Lorenzo Taschin[‡]

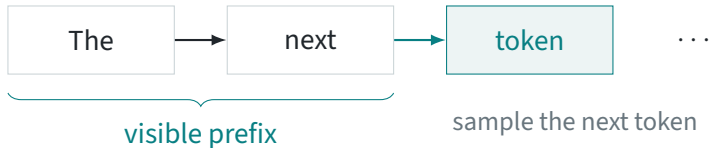
Authors listed in alphabetical order

[†]Bocconi University [‡]University of Ottawa *NTT Research (Work done mostly at Bocconi).

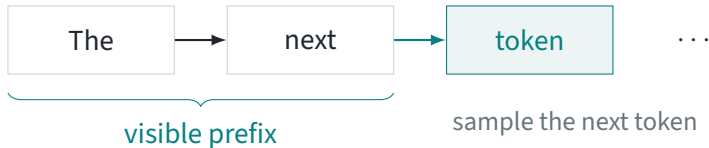
Left-to-right token generation



Left-to-right token generation



Left-to-right token generation



Which joint distributions can this generate efficiently?

Foundational questions: limitations and capabilities

Limitations

What is provably hard to generate?

Foundational questions: limitations and capabilities

Limitations

What is provably hard to generate?

AI Safety

Can limited private state prevent covert communication?

Foundational questions: limitations and capabilities

Limitations

What is provably hard to generate?

AI Safety

Can limited private state prevent covert communication?

Capabilities

Can changing the generation order make models more powerful?

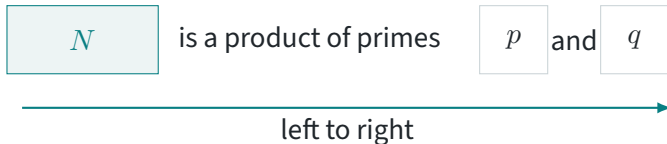
A factoring example

Random primes p, q $N = pq$

N is a product of primes p and q

A factoring example

Random primes p, q $N = pq$



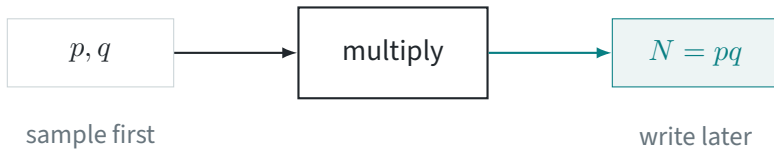
After writing N , how do we find p and q ?

Easy right-to-left generation

p, q

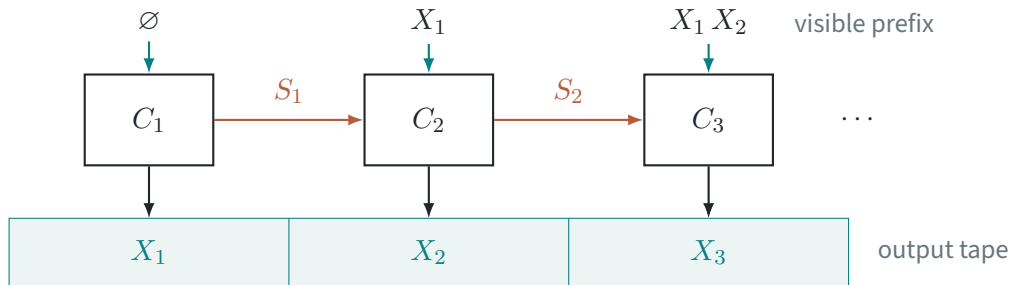
sample first

Easy right-to-left generation

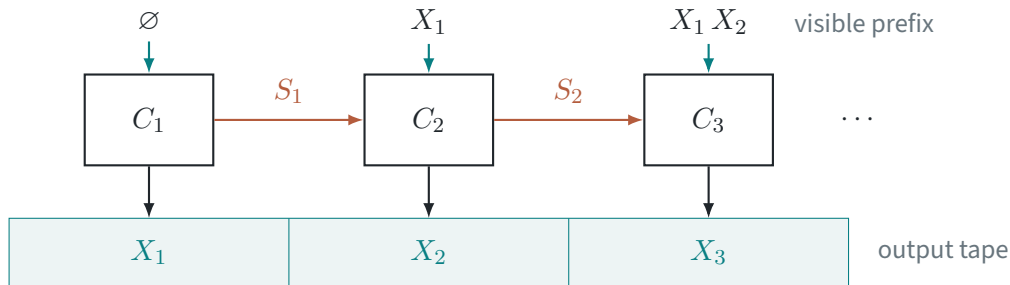


The generation order matters.

A model of sequential computation

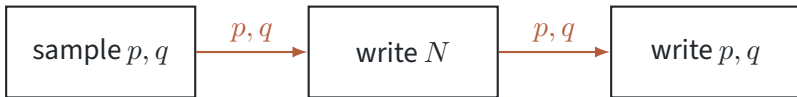


A model of sequential computation

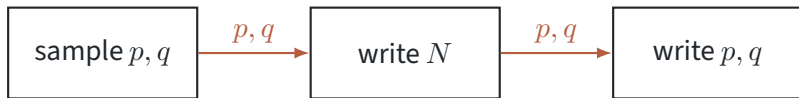


B gates per circuit k private bits between steps ℓ bits per token on the output tape

Private state

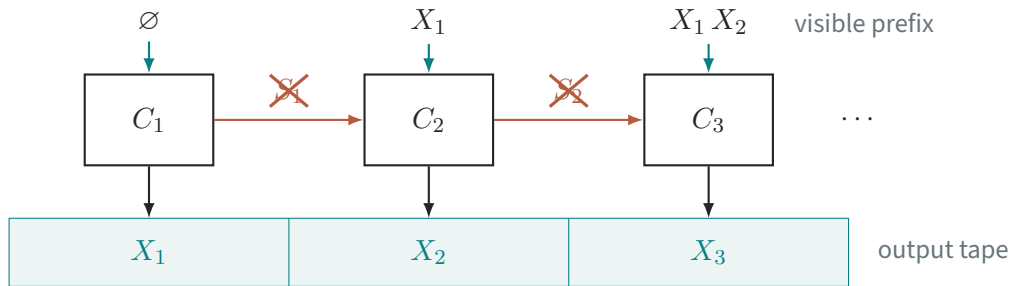


Private state

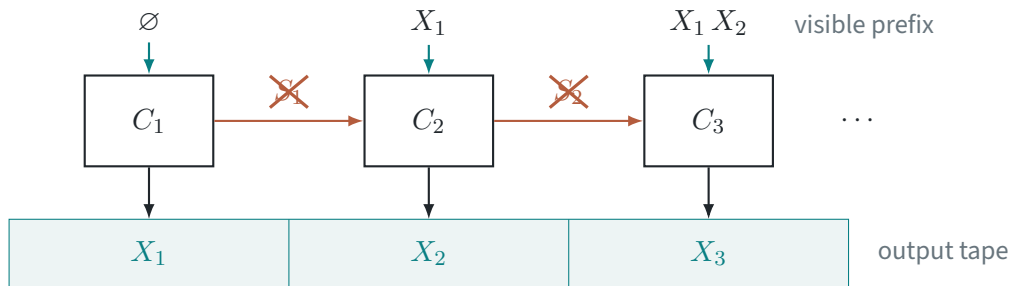


How much power does k bits of state buy?

Stateless setting

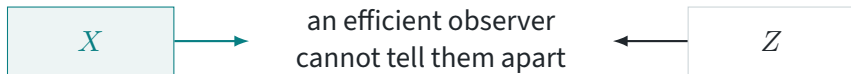


Stateless setting

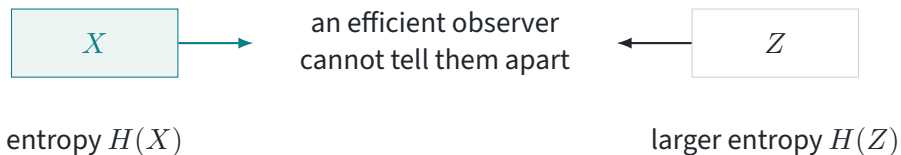


Which output distributions can we efficiently sample in this manner?

Pseudoentropy



Pseudoentropy



$$\text{Pseudoentropy gap} \geq H(Z) - H(X)$$

A pseudorandom generator (PRG)

k random bits



entropy at most k

A pseudorandom generator (PRG)

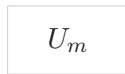
k random bits



entropy at most k

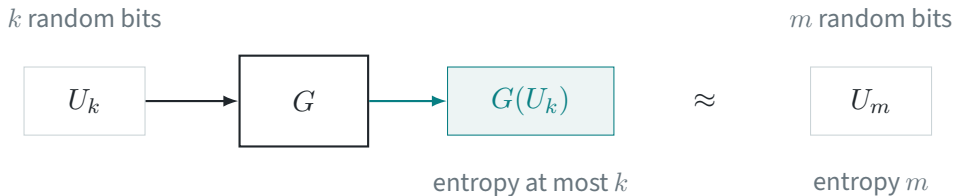
$\approx$

m random bits



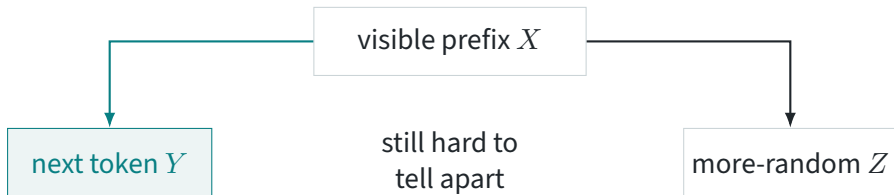
entropy m

A pseudorandom generator (PRG)

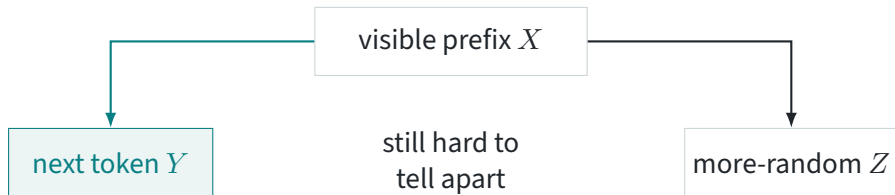


$$\text{Pseudoentropy gap} \geq m - k$$

Conditional pseudoentropy



Conditional pseudoentropy



$$H(Z \mid X) > H(Y \mid X) + \delta$$

Pseudoentropy characterizes sampling hardness

Hidden conditional entropy



Hardness of conditional sampling

Pseudoentropy characterizes sampling hardness

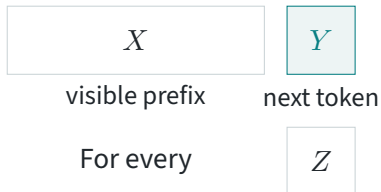
Hidden conditional entropy



Hardness of conditional sampling

Given the prefix, no efficient procedure
can accurately reproduce the next-token distribution.

Proof Sketch



$$H_{\text{comp}}(Y \mid X) < H(Y \mid X) + \delta$$

$$H(Z \mid X) \geq H(Y \mid X) + \delta$$

Proof Sketch

X

visible prefix

Y

next token

$$H_{\text{comp}}(Y \mid X) < H(Y \mid X) + \delta$$

For every

Z

$$H(Z \mid X) \geq H(Y \mid X) + \delta$$

there exists D_Z such that

$$\Pr[D_Z(X, Y) = 1] - \Pr[D_Z(X, Z) = 1] \geq \varepsilon$$

Proof Sketch

X

visible prefix

Y

next token

$$H_{\text{comp}}(Y \mid X) < H(Y \mid X) + \delta$$

For every

Z

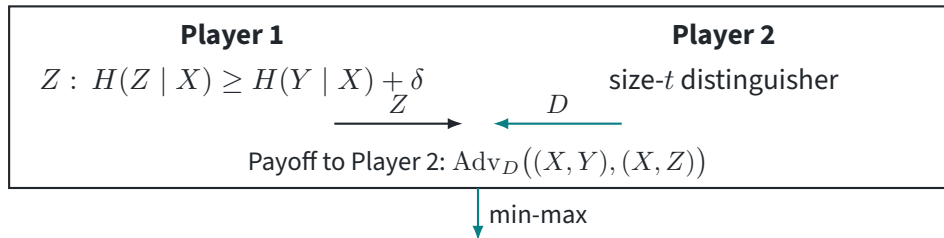
$$H(Z \mid X) \geq H(Y \mid X) + \delta$$

there exists D_Z such that

$$\Pr[D_Z(X, Y) = 1] - \Pr[D_Z(X, Z) = 1] \geq \varepsilon$$

For a sampler, we need a universal distinguisher.

A Universal Distinguisher



There exists a single distinguisher D

$$\forall Z : H(Z | X) \geq H(Y | X) + \delta$$

$$\text{Adv}_D((X, Y), (X, Z)) \geq \epsilon$$

A sampler from the universal distinguisher

Want Samp_D such that

$$D_{\text{KL}}((X, Y) \parallel (X, \text{Samp}_D(X))) \leq \delta.$$

The conditional Boltzmann sampler

$$\Pr[\text{Samp}_D(x) = y] = \frac{2^{kD(x,y)}}{\sum_{z \in \Sigma} 2^{kD(x,z)}}.$$

Choose $k \geq 0$ so that

$$H(\text{Samp}_D(X) \mid X) = H(Y \mid X) + \delta.$$

The entropy gap bounds the sampling error

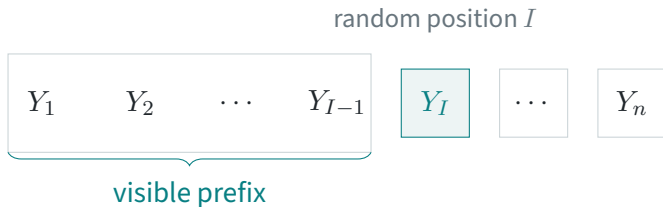
For this choice of sampler, we can prove the following fact:

$$\begin{aligned} D_{\text{KL}}((X, Y) \parallel (X, S)), \quad S &:= \text{Samp}_D(X) \\ &= \underbrace{H(S \mid X) - H(Y \mid X)}_{= \delta} - \underbrace{k \text{Adv}_D((X, Y), (X, S))}_{\geq k\varepsilon \geq 0} \\ &\leq \delta \end{aligned}$$

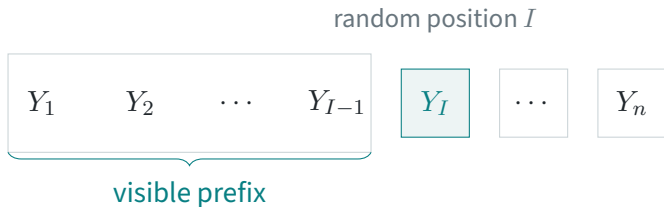
Next-block pseudoentropy



Next-block pseudoentropy



Next-block pseudoentropy



Measure the hidden entropy of the next token.
Average over positions.

The stateless characterization

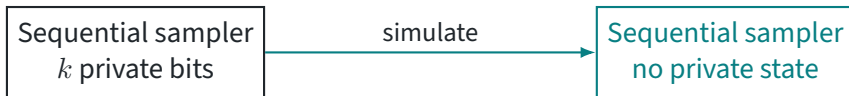
Small next-block pseudoentropy gap



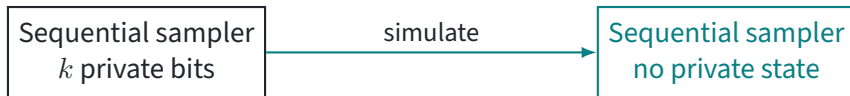
Accurate, efficient
stateless sequential sampling

Vadhan–Zheng, STOC 2012; Agrawal–Chen–Horel–Vadhan, CRYPTO 2019

Eliminating k bits of private state



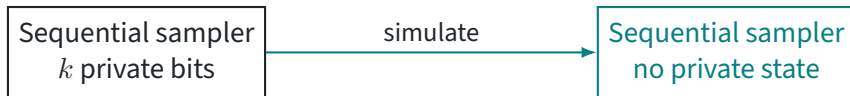
Eliminating k bits of private state



Circuit size bound

$$2^{O(k)} \times \text{polynomial in the other parameters}$$

Eliminating k bits of private state

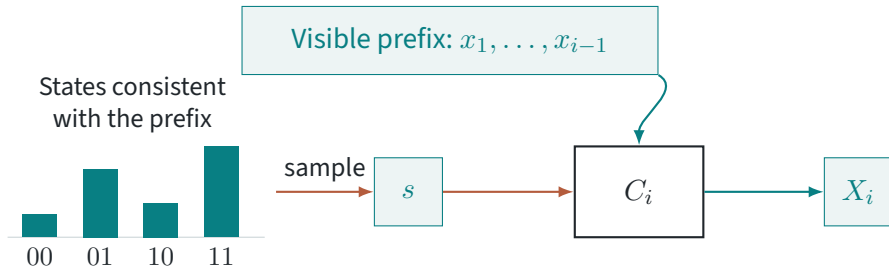


Circuit size bound

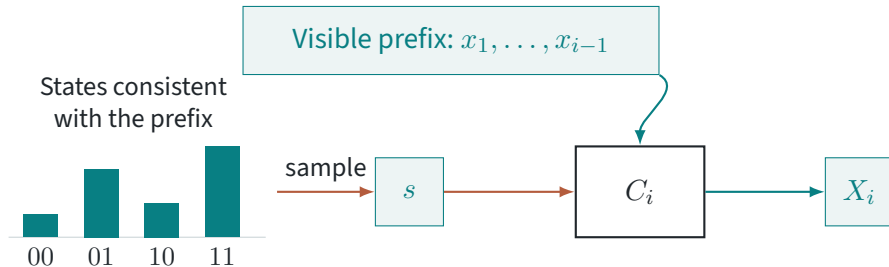
$$2^{O(k)} \times \text{polynomial in the other parameters}$$

To any desired accuracy.

The prefix determines a distribution over states



The prefix determines a distribution over states



Simulate this state distribution to generate the next token.

State simulation

Given prefix: $x_1, \dots, x_{j-1}, \underbrace{a}_{x_j}$

Sample states; run C_j

$(a, 01)$

$(b, 10)$

$(a, 01)$

$(a, 11)$

State simulation

Given prefix: $x_1, \dots, x_{j-1}, \underbrace{a}_{x_j}$

Sample states; run C_j

$(a, 01)$

$(b, 10)$

$(a, 01)$

$(a, 11)$



Keep states with token a

01

01

11

State simulation

Given prefix: $x_1, \dots, x_{j-1}, \underbrace{a}_{x_j}$

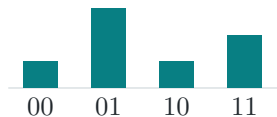
Sample states; run C_j

$(a, 01)$
 $(b, 10)$
 $(a, 01)$
 $(a, 11)$

Keep states with token a

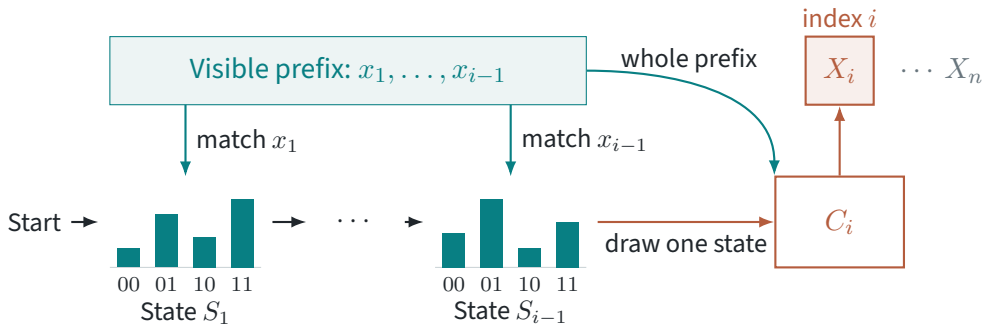
01
01
11

Add one to every count



Reconstructing the state distribution for C_i

At each index i , rebuild the smoothed state histograms through step $i - 1$.



Drawing the required samples for each histogram costs $\tilde{O}(2^{2k})$ gates naively, reduced to $\tilde{O}(2^k)$ via sorting networks.

Estimating an unknown state distribution is costly



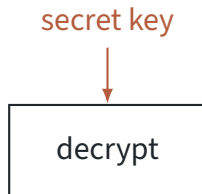
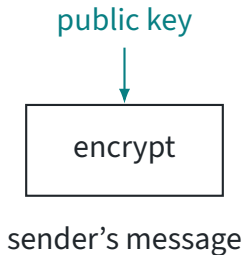
Expected KL error $\leq 1/4$ can require $\Omega(2^k)$ samples.

Logarithmic state can be eliminated efficiently

Logarithmic private state can be simulated with a polynomial circuit-size blow-up.

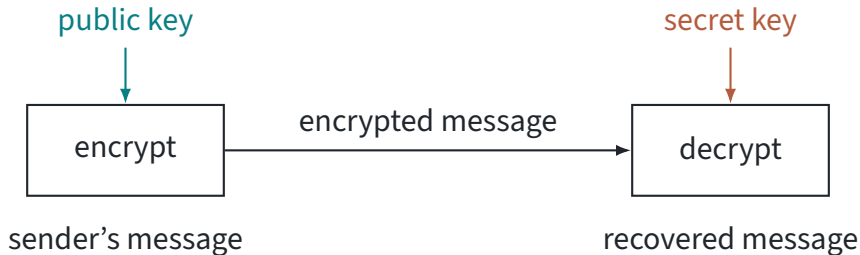
No PKE for small-state sequential samplers

Public-key encryption (PKE)



No PKE for small-state sequential samplers

Public-key encryption (PKE)



Public-key encryption: informal theorem

An efficient sequential sampler
with only **logarithmic private state**

cannot implement secure
public-key encryption.

Public-key encryption: proof sketch

Correctness forces a noticeable next-token difference between m_0 and m_1 at some step.

Public-key encryption: proof sketch

Correctness forces a noticeable next-token difference between m_0 and m_1 at some step.

Rerun the same stateless next-token sampler on the observed prefix equally often with each message and accept if the observed token occurs more often for m_0 ,
contradicting IND-CPA security.

Steganography

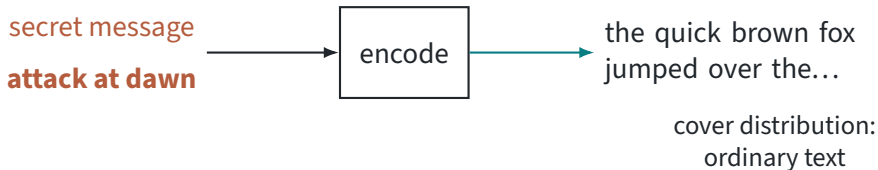
secret message

attack at dawn

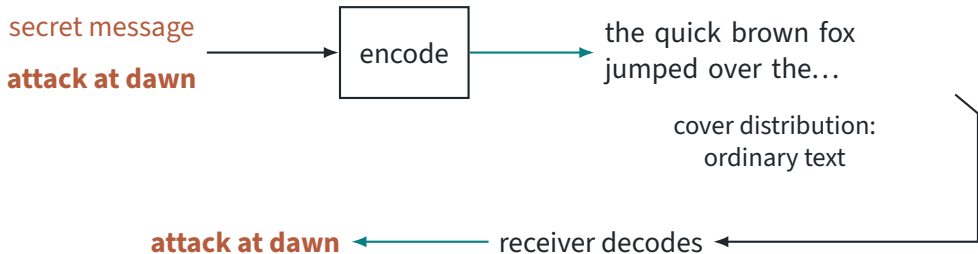


encode

Steganography



Steganography



Public-key steganography



Public-key steganography is public-key encryption
whose ciphertext looks like ordinary text.

Insertional generation

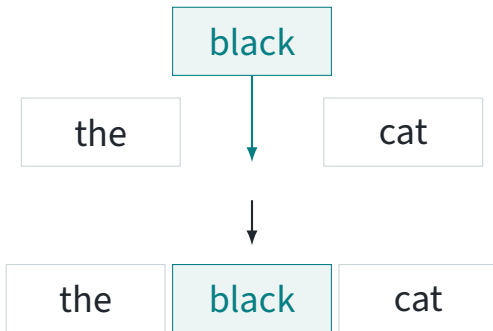
Insertional generation

the

cat

Stern, Chan, Kiros, Uszkoreit, ICML 2019.

Insertional generation



Stern, Chan, Kiros, Uszkoreit, ICML 2019.

Insertional generation is strictly more powerful

One-way function f : easy to compute, hard to invert.

Target: $(f(X), X)$, with X uniform.

Left to right
no private state



Hard

Insertional generation is strictly more powerful

One-way function f : easy to compute, hard to invert.

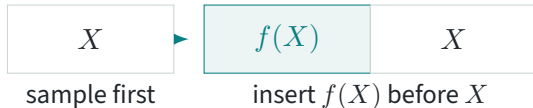
Target: $(f(X), X)$, with X uniform.

Left to right
no private state



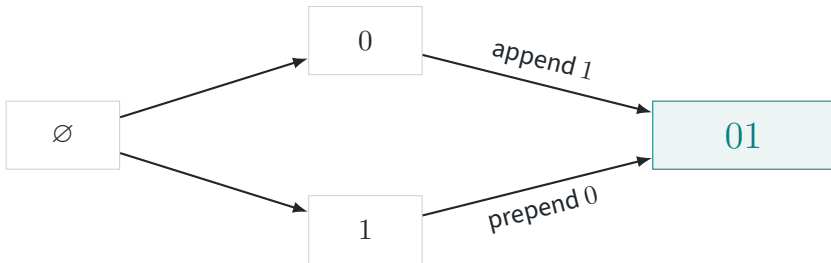
Hard

Insertion



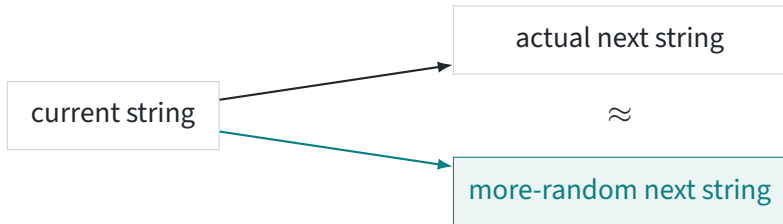
Easy

Different decompositions, same outcome



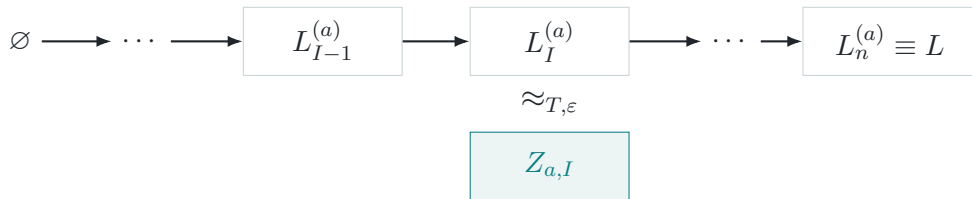
Insertional pseudoentropy

A large insertional pseudoentropy gap means: every generation procedure has a more-random distribution that fools *all* efficient tests.



Definition: insertional pseudoentropy

Insertional pseudoentropy $\geq k$: $\forall$ insertional decomposition a of L , $\exists Z_{a,I}$.



I : independent uniform step. Compare jointly with $(I, L_{I-1}^{(a)})$.

$$H(Z_{a,I} \mid I, L_{I-1}^{(a)}) - H(L_I^{(a)} \mid I, L_{I-1}^{(a)}) \geq \frac{k - H(L)}{n}.$$

Our work: characterizing insertional samplability

Small insertional pseudoentropy gap



Accurate, efficient insertional sampling

No private state; polynomial-size token alphabet.

PRGs can be insertionally samplable

A PRG looks random. Shouldn't every fixed order be hard?

PRGs can be insertionally samplable

A PRG looks random. Shouldn't every fixed order be hard?

Surprisingly, there exists a PRG that is easy to generate insertionally.

PRGs can be insertionally samplable

A PRG looks random. Shouldn't every fixed order be hard?

Surprisingly, there exists a PRG that is easy to generate insertionally.

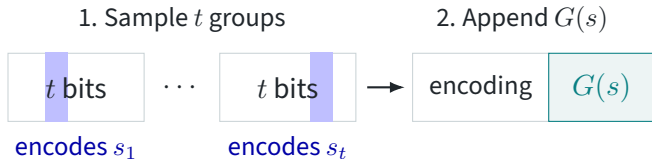
1. Sample t groups



PRGs can be insertionally samplable

A PRG looks random. Shouldn't every fixed order be hard?

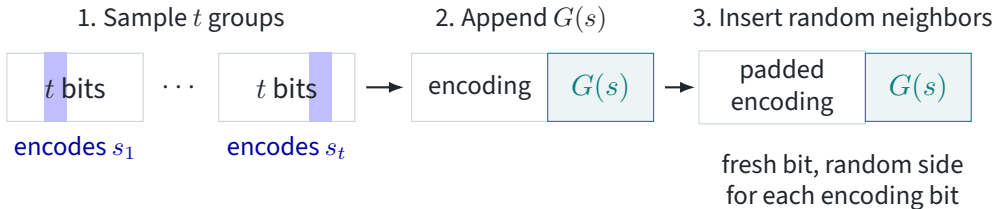
Surprisingly, there exists a PRG that is easy to generate insertionally.



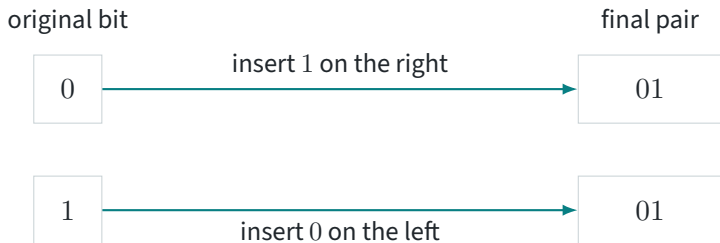
PRGs can be insertionally samplable

A PRG looks random. Shouldn't every fixed order be hard?

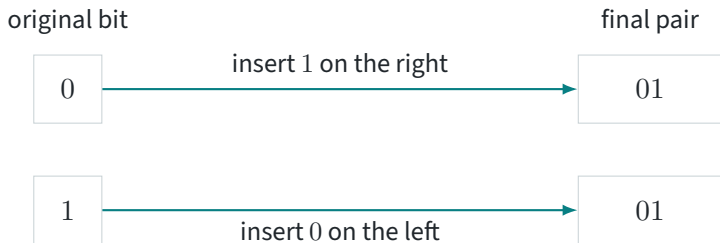
Surprisingly, there exists a PRG that is easy to generate insertionally.



Random insertions conceal the seed

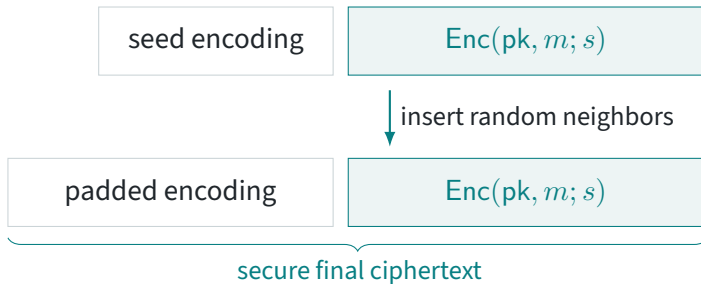


Random insertions conceal the seed

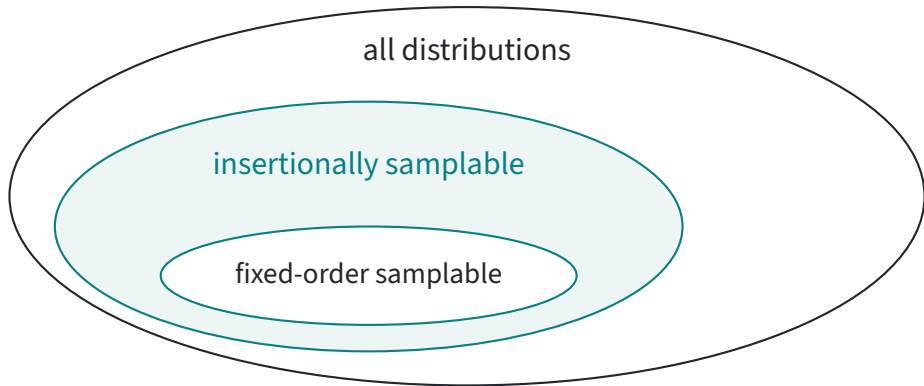


Which bit was there first?

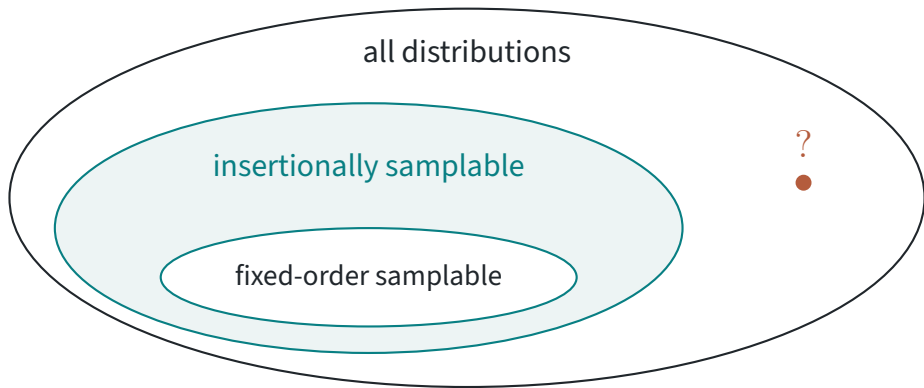
The same construction applies to PKE



The power of efficient, stateless sampling



Can we construct a distribution here efficiently?



Insertional generation also has limits

Theorem (informal)

Assuming sufficiently strong one-way functions, some distributions are

easy for ordinary
randomized circuits

hard for stateless
insertional samplers

Insertional generation also has limits

Theorem (informal)

Assuming sufficiently strong one-way functions, some distributions are

easy for ordinary
randomized circuits

hard for stateless
insertional samplers

Alphabet size $2^{\Theta(n)}$, for n output tokens.

Open questions

Can low-state agents establish a secret interactively?

Open questions

Can low-state agents establish a secret interactively?

What if every intermediate string is public?

Open questions

Can low-state agents establish a secret interactively?

What if every intermediate string is public?

Can insertion be hard even with a small alphabet?

Appendix: pseudoentropy definitions

For distributions P, Q on the same finite set, write $P \approx_{t,\varepsilon} Q$ when every Boolean circuit A of size at most t satisfies

$$|\Pr[A(P) = 1] - \Pr[A(Q) = 1]| \leq \varepsilon.$$

Ordinary pseudoentropy

$$H_{\text{comp}}^{t,\varepsilon}(X) = \sup_{Z: X \approx_{t,\varepsilon} Z} H(Z).$$

Conditional pseudoentropy For a fixed joint distribution (U, V) ,

$$H_{\text{comp}}^{t,\varepsilon}(V \mid U) = \sup_{Z: (U,V) \approx_{t,\varepsilon} (U,Z)} H(Z \mid U).$$

Here Z has the same range as V and is jointly distributed with U . The conditional gap is $\Delta_{t,\varepsilon}(V \mid U) = H_{\text{comp}}^{t,\varepsilon}(V \mid U) - H(V \mid U)$.

Appendix: next-block pseudoentropy and sampling

Let $Y = (Y_1, \dots, Y_n) \in (\{0, 1\}^\ell)^n$, with $n, \ell \geq 1$, and choose $I \leftarrow [n]$ uniformly and independently of Y .

$$\text{NBPE}_{t,\varepsilon}(Y) = nH_{\text{comp}}^{t,\varepsilon}(Y_I \mid Y_{<I}), \quad \text{gap} = \text{NBPE}_{t,\varepsilon}(Y) - H(Y).$$

The witness is one jointly distributed $Z \in \{0, 1\}^\ell$ satisfying $(Y_{<I}, Y_I) \approx_{t,\varepsilon} (Y_{<I}, Z)$. Prefix length determines I , and $nH(Y_I \mid Y_{<I}) = H(Y)$.

For finite distributions P, Q , the forward KL error is

$$D_{\text{KL}}(P \parallel Q) = \sum_x P(x) \log_2 \frac{P(x)}{Q(x)},$$

with value $+\infty$ if $Q(x) = 0 < P(x)$ for some x . V is (B, d) KL-hard to sample from U if every randomized circuit S of size at most B satisfies

$$D_{\text{KL}}((U, V) \parallel (U, S(U))) > d.$$

Appendix: Vadhan–Zheng with parameters

Let $(U, V) \in \{0, 1\}^a \times [q]$, with $t \geq 1$, $a \geq 0$, $q \geq 2$, $d, g \geq 0$, and $0 < \varepsilon, \eta, \sigma < 1$. There exist a universal $c_0 \in (0, 1]$ and a fixed polynomial p_0 such that:

A large gap gives hardness. If $\Delta_{t,\varepsilon}(V \mid U) \geq d$, then every randomized circuit S with

$$|S| \leq \frac{\min\{t^{c_0}, \sigma/\varepsilon\}}{p_0(a, q, 1/\eta, \log(1/\sigma))}$$

satisfies $D_{\text{KL}}((U, V) \parallel (U, S(U))) > d - \sigma - \eta$.

A small gap gives a sampler. If $\Delta_{t,\eta}(V \mid U) \leq g$, there exists a randomized circuit S with

$$|S| \leq p_0(t, a, q, 1/\eta), \quad D_{\text{KL}}((U, V) \parallel (U, S(U))) \leq g + 2\eta.$$

Appendix: the two quantitative characterizations

Let $Y \in (\{0, 1\}^\ell)^n$, where $n, \ell \geq 1$, and let $P_{t,\varepsilon}(Y)$ be either $\text{NBPE}_{t,\varepsilon}(Y)$ or $\text{IPE}_{t,\varepsilon}(Y)$. Use the corresponding stateless sampler model. Assume $t \geq 1$, $d \geq \varepsilon$, and $0 < \varepsilon, \eta, \sigma < 1$.

For universal $c_0 \in (0, 1]$ and a fixed polynomial p :

1. If $P_{t,\varepsilon}(Y) \geq H(Y) + d$, then every sampler with per-step size

$$B \leq \frac{\min\{t^{c_0}, \sigma/(n\varepsilon)\}}{p(n, 2^\ell, 1/\eta, \log(n/\sigma))}$$

satisfies $D_{\text{KL}}(Y \| C) > d - \sigma - n\eta$.

2. If $P_{t,\varepsilon}(Y) \leq H(Y) + d - \varepsilon$, there is a sampler with

$$B \leq p(t, n, 2^\ell, 1/\varepsilon), \quad D_{\text{KL}}(Y \| C) \leq d.$$

The polynomial and constant may differ between the two models.

Appendix: insertional pseudoentropy

Let $L \in (\{0, 1\}^\ell)^n$, with $n, \ell \geq 1$. A decomposition is a joint sequence $(L_0, \dots, L_n)$ with $L_0 = \emptyset$, $L_n \equiv L$, and L_i obtained by one insertion into L_{i-1} .

For $t \geq 1$, $k \geq 0$, and $0 < \varepsilon < 1$, L has (t, ε) insertional pseudoentropy at least k if **for every decomposition**, with independent uniform $I \leftarrow [n]$, **there exists** a jointly distributed Z , of length I tokens, such that

$$(I, L_{I-1}, L_I) \approx_{t, \varepsilon} (I, L_{I-1}, Z)$$

and

$$H(Z \mid I, L_{I-1}) - H(L_I \mid I, L_{I-1}) \geq \frac{k - H(L)}{n}.$$

$\text{IPE}_{t, \varepsilon}(L)$ is the supremum of such k ; its gap is $\text{IPE}_{t, \varepsilon}(L) - H(L)$.

The same witness Z must fool every size- t circuit. It need not be a valid insertion into the current string.

Appendix: the state-elimination bound

Let $C = (C_1, \dots, C_n)$ have k private bits, size at most B per step, and token alphabet $\mathcal{X} = \{0, 1\}^\ell$. Put $\mathcal{S} = \{0, 1\}^k$. Assume $n \geq 1, k, \ell \geq 0$, and $0 < \delta \leq 1$.

There is a stateless sequential sampler $D = (D_1, \dots, D_n)$ satisfying

$$D_{\text{KL}}(C(1^n) \| D(1^n)) \leq \delta$$

with circuit size bound

$$\max_i |D_i| \leq O\left(\frac{n^3 |\mathcal{X}| |\mathcal{S}|}{\delta} \left[B + \log^3\left(\frac{n |\mathcal{X}| |\mathcal{S}|}{\delta}\right) \right]\right).$$

The circuit construction is deterministic; sampling and estimation use fresh internal coins.

For $|\mathcal{X}| = \text{poly}(n)$, $B = \text{poly}(n)$, $k = O(\log n)$, and $1/\delta = \text{poly}(n)$, the resulting circuits have polynomial size.

Appendix: accumulation of the simulation error

For the target sampler C , let $\pi_j^{x_{\leq j}}$ be the distribution of S_j conditioned on $X_{\leq j} = x_{\leq j}$. Let $\tilde{\pi}_j$ be its implemented estimate, including finite-coin rounding. Use $m = \lceil 4n^2 |\mathcal{X}| |\mathcal{S}| / \delta \rceil$ samples per joint estimate and set

$$e_j = \mathbb{E}_{x_{\leq j} \leftarrow X_{\leq j}} \mathbb{E}_{\text{estimation coins}} D_{\text{KL}}(\pi_j^{x_{\leq j}} \| \tilde{\pi}_j).$$

The joint Laplace estimate, conditioning, and data processing give

$$e_1 \leq \frac{2|\mathcal{X}| |\mathcal{S}|}{m}, \quad e_j \leq e_{j-1} + \frac{2|\mathcal{X}| |\mathcal{S}|}{m}.$$

The chain rule and convexity yield

$$D_{\text{KL}}(C \| D) \leq \sum_{j=1}^{n-1} e_j \leq \frac{n(n-1) |\mathcal{X}| |\mathcal{S}|}{m} \leq \delta.$$

All estimation coins are independent of the target prefix.

Appendix: estimating an unknown state distribution

An algorithm receives at most N independent samples from an unknown distribution P on $\mathcal{S} = \{0, 1\}^k$, where $k \geq 1$. It receives no other information about P and outputs a probability vector $\hat{\pi}$, with no subsequent oracle access.

There exists P for which

$$\mathbb{E}[D_{\text{KL}}(P \parallel \hat{\pi})] \geq 1 - \frac{2N}{2^k}.$$

The expectation includes the samples and the algorithm's coins. Hence

$$\forall P : \mathbb{E}[D_{\text{KL}}(P \parallel \hat{\pi})] \leq \frac{1}{4} \quad \implies \quad N \geq \frac{3}{8} 2^k.$$

This is a sample bound for estimating the probability vector, rather than a circuit lower bound for every possible state simulation.

Appendix: public-key encryption and steganography

Let a polynomial-time IND-CPA scheme be secure against nonuniform polynomial-size circuits, with negligible correctness error and security parameter λ . Ciphertexts have a fixed polynomial number of $O(\log \lambda)$ -bit tokens.

For any two distinct equal-length, polynomial-length messages m_0, m_1 , there cannot be polynomial-size sequential samplers C_0, C_1 with $O(\log \lambda)$ private bits satisfying, for both $b \in \{0, 1\}$,

$$D_{\text{KL}}((\text{pk}, \text{Enc}(\text{pk}, m_b)) \parallel (\text{pk}, C_b(\text{pk}, m_b))) = \text{negl}(\lambda).$$

The public key, message, and visible prefix are available at every step.

Public-key steganography. The same conclusion holds for encodings with negligible decoding error, indistinguishable from the same cover distribution, jointly with the public key, with the same token and size bounds and no pre-shared secret. This statement concerns public-key encodings; it does not settle interactive key agreement.

Appendix: insertionally generated encryption

Let $\text{Enc}(\text{pk}, m; s)$ use $t = \text{poly}(\lambda)$ uniform coin bits, with $t \geq \lambda$ after padding, and have fixed ciphertext length L .

1. Append t^2 uniform bits X . The seed s consists of the parities of its t groups, each containing t bits.
2. Append the L encryption bits, recomputing s from X at each step.
3. Insert a fresh uniform bit before or after each X_i with equal probability. Call the padded string X^{pad} . Output $X^{\text{pad}} \parallel \text{Enc}(\text{pk}, m; s)$.

Current length identifies the phase; in phase 3, X_i is at position $2i - 1$. Each transition uses only the current string, pk , m , and fresh coins.

$$\Delta(\text{Enc}'(\text{pk}, m), U_{2t^2} \parallel \text{Enc}(\text{pk}, m; U_t)) \leq t2^{-t}.$$

Decryption ignores the padded prefix. Correctness is unchanged, and IND-CPA advantage increases by at most $2t2^{-t}$. Security covers the final string; the insertion history reveals s .

Appendix: general sampling versus insertion

Assume length-preserving one-way functions f_λ with $T = 2^{\gamma\lambda^\alpha}$ security, where $\gamma > 0$ and $1/2 < \alpha < 1$: size- T circuits invert with probability at most $1/T$.

For every $t \geq 2$, put $\ell = 2t + \lceil \log(2t) \rceil$. There exists $L_t \in (\{0, 1\}^\ell)^{2t}$ such that:

1. A $\text{poly}(t)$ -size randomized circuit samples L_t exactly.
2. For every constant $c > 0$ and all sufficiently large t , every stateless insertional sampler C of size at most 2^{ct} satisfies

$$D_{\text{KL}}(L_t \| C(1^{2t})) \geq \gamma t^{2\alpha} - 2t.$$

The alphabet has 2^ℓ symbols: the token length grows with t . The fixed-small-alphabet case remains open.

References

Gal Arnon, Andrej Bogdanov, Paul Lou, Giulio Malavolta, Alon Rosen, Lorenzo Taschin.

Limitations on Autoregressive Token Generation via Sequential Sampling.

Salil Vadhan and Colin Jia Zheng.

Characterizing Pseudoentropy and Simplifying Pseudorandom Generator Constructions.

STOC 2012.

Rohit Agrawal, Yi-Hsiu Chen, Thibaut Horel, Salil Vadhan.

Unifying Computational Entropies via Kullback–Leibler Divergence.

CRYPTO 2019.

Mitchell Stern, William Chan, Jamie Kiros, Jakob Uszkoreit.

Insertion Transformer: Flexible Sequence Generation via Insertion Operations.

ICML 2019.