LANGUAGE MODELING IS MONOTONE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

A long-standing empirical observation spanning the domains of artificial intelligence and neuroscience asserts that intelligence is equivalent to compression—an entity that is able to compress information efficiently demonstrates intuitive notions of intelligence and learning, and vice versa. Indeed, recent experimental works verify this intuition by showing connections between the capabilities of large language models (LLMs) and their ability as compressors: for instance, Delétang et al. (ICLR'24) demonstrate that LLMs can be used as powerful compressors, and Huang et al. (COLM'24) show that the compression ability of LLMs is highly correlated with performance on benchmarks for knowledge and reasoning. In this work, we initiate a *theoretical* study of this connection. Our main result is that LLMs (formally modeled as next-token predictors) are equivalent to *monotone* (a.k.a. order-preserving) compression algorithms—namely, compression algorithms where the encoding process preserves the ordering of the inputs—in the sense that the one can be constructed from the other while preserving the same error up to an additive gap of 2. We next show that the monotonicity is *required* for this equivalence to hold if and only if cryptographic (infinitely-often) one-way functions exist. As a direct corollary, we get a cryptographic result of independent interest: the notion of next-bit pseudoentropy (a computational analogue of entropy) of a distribution is equivalent to *monotone* incompressibility of the distribution. (Previously, it was only known (Haitner et al., ITCS'23) that incompressibility *implies* next-bit pseudoentropy.)

1 INTRODUCTION

One of the most central theoretical questions in the study and development of artificial intelligence is: how exactly does one define the concept of "intelligence"? Turing's classic "imitation game" (Turing, 1950), which posits that a machine should be considered intelligent if a human is unable to distinguish it from another human given a transcript of a conversation, gives an answer that is reasonable but ultimately subjective. Towards defining intelligence more concretely, the work of Shannon (1948; 1951) laid the groundwork for a line of research proposing the idea that learning or understanding something is equivalent to being able to process and describe it in a simple way (Mahoney, 1999; Hernandez-Orallo, 1998)—that is, intelligence is the ability to succinctly *compress* information.

Recently, large language models (LLMs) have made this connection more explicit. LLMs are, at their core, *autoregressive* models (that is, next-token predictors) that are trained using a given training dataset to, given some prefix of text (modeled as tokens) $t_1, t_2, \ldots t_k$, predict the next token t_{k+1} in the sequence by outputting a probability distribution over possible tokens. A model is "good" if its loss (in terms of the negative log-likelihood, or cross-entropy, of the distribution of the model's autoregressively sampled outputs relative to the distribution of its training data) is close to minimal.

It is well known that good compression schemes can be constructed from good autoregressive models by leveraging *arithmetic encoding* (Shannon, 1948; Gilbert & Moore, 1959): arithmetic encoding is known to produce a near-optimal compression rate for a given distribution, so, intuitively, applying it to the distribution of a model's autoregressively sampled outputs should likewise give a near-optimal compression rate over the respective distribution of training data if the model is good. This was first empirically studied using neural network predictors (Schmidhuber & Heil, 1994) (see also, e.g., Schmidhuber & Heil (1996); Mahoney (2000); Schiopu et al. (2018)), but subsequently has been applied to LLMs, with recent research suggesting not only that compression rivaling state-of-

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.

the-art algorithms can be constructed from LLMs (e.g., Delétang et al. (2024); Zhang et al. (2026); Tsai (2026)), but also that compression ability of LLMs is highly correlated with performance on benchmarks measuring knowledge, reasoning, and coding abilities (Huang et al., 2024).

Given that compression can be constructed from LLMs, a natural question is whether the *converse* may also hold—that is, whether one can construct accurate next-token predictors from good compression algorithms. This idea was originally empirically explored in the work of Frank et al. (2000) constructing models for text categorization from compression algorithms, and is of particular interest because it would imply an intuitive *equivalence* between predictive modeling and compression. More recently, Delétang et al. (2024) suggest and evaluate a heuristic construction of an autoregressive next-token prediction model from compression, leading them to their titular assertion that "language modeling is compression".

However, when one compares the results in each direction of this proposed empirical equivalence, there is a clear difference in terms of the empirical quality of results: compression algorithms constructed from LLMs exhibit competitive compression ratios, whereas works constructing autoregressive models from compression exhibit relatively poor predictive ability compared to state-of-art LLMs. While there are certainly plausible intuitive explanations for this disparity—for instance, the difference in computational power between state-of-the-art LLMs and state-of-the-art compression algorithms—it nonetheless raises a fundamental question:

Is there a formal *connection between autoregressive models (i.e., LLMs) and compression?*

In this work, we address this question, initiating a theoretical study of the intuitive connection between autoregressive models and compression. Our main result shows that, intuitively, "good" autoregressive models are equivalent to "good" *monotone* compression schemes. (Furthermore, this equivalence holds quantitatively up to a constant additive accuracy gap between the cross-entropy loss of the autoregressive model and the compression loss of the monotone compressor.) We additionally show that the monotonicity condition is *necessary* if and only if cryptographic (infinitely-often) one-way function (ioOWFs) exist; this explains the disparity observed above.

1.1 Our Results

We formalize LLMs as *autoregressive models* which, given a string of tokens as a prefix, output a probability distribution over possible next tokens. We consider a setting where the objective of an LLM is, given a training distribution D (e.g., random files selected from the Internet), to learn a model $\mathcal{M}$ that minimizes the expected *negative log-likelihood* $\mathbf{E}_{t\leftarrow D}[-\sum_{i\in[\text{len}(t)]}\log\mathcal{M}(t_i \mid t_{<i})]$, which is in fact equivalent to the *cross-entropy* $H(D, D_{\mathcal{M}}) = H(D) + KL(D \mid\mid D_{\mathcal{M}})$ (where $H(D)$ is Shannon entropy and KL represents Kullback-Leibler divergence) between D and the distribution $D_{\mathcal{M}}$ of outputs autoregressively sampled from $\mathcal{M}$. As the minimum achievable cross-entropy is $H(D)$, we say that a model $\mathcal{M}$ is δ-optimal with respect to D if its cross-entropy is at most $H(D) + \delta$.

A compression scheme is specified by a pair of algorithms $(\mathsf{Enc}, \mathsf{Dec})$ where Enc *encodes* an input from a distribution D into its compressed form (which we assume throughout to be binary), and Dec is the respective inverse. The goal of compression is to minimize the expected compression length $\mathbf{E}_{x\leftarrow D}[\text{len}(\mathsf{Enc}(x))]$; by Shannon's source coding theorem (Shannon, 1948), this is lower-bounded by $H(D)$, and so we say that compression is δ optimal with respect to D if it's expected compression length is at most $H(D)+\delta$ (in other words, the *compression loss* is at most δ). Finally, compression is *monotone* if it preserves a lexicographical ordering $\prec$ over inputs from the support of D: that is, given inputs x, y, $\mathsf{Enc}(x) < \mathsf{Enc}(y)$ if and only if $x \prec y$.

For all of our results, we require autoregressive models and compression schemes to be *efficiently computable*—that is, compressing, decompressing, or determining the next-token probability for any token and prefix of a model should require computation time polynomial in the input length of the algorithm. Furthermore, when discussing cross-entropy and expected compression length, we consider it over *complete samples* from the distribution D.

Autoregressive modeling is monotone compression. We first present our main result, showing the equivalence between autoregressive modeling and monotone compression:

Theorem 1.1 (Theorem 3.1 and Theorem 3.3, informal)**.** *Given a distribution D consisting of strings of tokens from an alphabet Σ, for any δ:*

- *Given an efficiently computable δ-optimal autoregressive model for D, one can construct an efficiently computable monotone compression scheme which is $(\delta + 2)$-optimal with respect to D.*[1]
- *Given an efficiently computable monotone compression scheme which is δ-optimal with respect to D, one can construct a δ-optimal autoregressive model for D.*

In other words, autoregressive models and monotone compressors are equivalent up to an additive gap of 2; we additionally demonstrate this equivalence through an explicit efficient transformation in both directions.

The monotonicity requirement. While the added condition of monotonicity may seem surprising at first glance, it sheds light on the discrepancy we noted earlier between empirical results converting autoregressive models to compression and vice versa. Known constructions of compression from autoregressive models use arithmetic encoding, which is monotone; meanwhile, as monotonicity is not necessarily inherent in compression algorithms, it explains why constructions of autoregressive models from compression tend to have worse empirical performance. Indeed, our next result formally demonstrates that the requirement of monotonicity is not merely coincidental: given a fundamental and widely conjectured cryptographic assumption, the existence of infinitely-often one-way functions (ioOWFs)[2], monotonicity is *necessary* for any strong autoregressive model-compression equivalence:

Theorem 1.2 (Theorem 3.4 and Corollary 3.5, informal)**.** *The following are true:*

- *If (infinitely-often) one-way functions exist, then there exists an efficiently samplable distribution family $\mathcal{D}$ and a* non-monotone *compression scheme $(\mathsf{Enc}, \mathsf{Dec})$ for $\mathcal{D}$ with efficient encoding and decoding such that Enc is perfectly lossless but no efficiently computable autoregressive model can predict $\mathcal{D}$ with expected cross-entropy less than linear in the output length.*
- *If infinitely-often one-way functions do not exist, then, given a δ-optimal compression scheme $(\mathsf{Enc}, \mathsf{Dec})$ with efficient encoding and decoding for an efficiently samplable distribution family $\mathcal{D}$, one can construct an efficiently computable autoregressive model which is statistically close to a δ-optimal autoregressive model for $\mathcal{D}$.*

That is, given ioOWFs, one could not hope to prove Theorem 1.1, or even any weaker analogue with an additive gap constant sublinear in the length of a sample from the training distribution, without requiring monotonic compression. Interestingly, the same is not true if ioOWFs do not exist: in that case, using a classic result from Impagliazzo & Luby (1989), we show that one can drop the monotonicity condition from Theorem 1.1 at a cost of a small amount of statistical error. Thus, in essence, whether modeling is equivalent to *monotone* compression or (approximately) equivalent to *all* compression depends on the truth of this assumption; that said, one-way functions (which in turn imply ioOWFs) are widely conjectured to exist, implying the former equivalence and a corresponding separation between monotone and non-monotone compression.

Pseudoentropy and incompressibility. Lastly, we show a corollary of Theorem 1.1 that provides interesting implications to cryptography. There has been a considerable line of research relating the compressibility of a distribution to its *pseudoentropy* (Håstad et al., 1999), a computational analogue of entropy. In particular, Haitner et al. (2023) demonstrate that, if a family of distributions $\mathcal{D}$ is *incompressible* (Yao, 1982) to some length $k(n)$, then it has *next-bit pseudoentropy* at least $(k(n) - 2)$—that is, the total pseudoentropy of the conditional next-bit distributions of each bit of $\mathcal{D}$

[1] The additive gap of 2 in this result, and later Corollary 1.3, arises from the use of arithmetic encoding in the transformation from autoregressive modeling to compression; some additive loss is inevitable as no compression scheme can achieve perfect losslessness without strict conditions on the distribution D.

[2] *Infinitely-often* OWFs are a relaxation of the standard concept of a OWF: namely, OWFs which are only uninvertible for infinitely many input lengths n rather than all sufficiently large n.

is at least $(k(n)-2)$.[3] However, the converse is false: if one-way functions exist, then pseudoentropy cannot imply incompressibility. As our final result, we obtain a tight version of this characterization by combining Theorem 1.1 with a corollary from Vadhan & Zheng (2012) to show that, similarly to the equivalence between autoregressive models and compression, both directions are true if we specifically require *monotone* incompressibility, or inability to be compressed through specifically monotone schemes:

Corollary 1.3 (Corollary 3.6 and Corollary 3.7, informal)**.** *Given a family of distributions $\mathcal{D}$, the following are equivalent for any function $k(n)$ (to within an additive gap of 2):*

- *$\mathcal{D}$ cannot be efficiently* monotonically *compressed to length $k(n)$.*
- *$\mathcal{D}$ has next-bit pseudoentropy at least $k(n)$.*

1.2 Technical Overview

To construct monotone compression from an autoregressive model $\mathcal{M}$, we can leverage arithmetic encoding relative to the model's output distribution $D_{\mathcal{M}}$: it is fairly straightforward to show that, for a training distribution D, the expected compression length is at most the cross-entropy $H(D, D_{\mathcal{M}})$ plus 2, so δ-optimality of $\mathcal{M}$ implies $(\delta+2)$-optimality of the compression scheme. A minor subtlety is that arithmetic encoding is only efficiently computable when the cumulative distribution function (CDF) of any element of the output distribution is likewise efficiently computable. However, given a polynomial-size alphabet, we can compute the CDF for a given LLM output efficiently by using the assigned next-token probabilities to iteratively sum, for each token, the probability for the prefix to be equal to *and* the current token to be smaller than that of the given output; thus, as long as the next-token probabilities of the model $\mathcal{M}$ are efficiently computable, the arithmetic encoding with respect to $D_{\mathcal{M}}$ is as well.

For the converse, given some encoding Enc over a training distribution D that assigns each element x in the support of D a codeword with some length $\mathrm{len}(x)$, we construct a model $\mathcal{M}$ whose output distribution $D_{\mathcal{M}}$ assigns each element x in the support of D a probability of $2^{-\mathrm{len}(\mathsf{Enc}(x))}$. Assuming for simplicity that this defines a well-formed distribution (which we discuss below), this distribution has the same cross-entropy with respect to D as the expected compression loss over D of Enc compared to the optimal compression scheme for D; thus, this transformation preserves δ-optimality with no additive loss. Efficiency is again a non-trivial concern, as we must show that computing any next-token probability for the model we construct must be efficient: however, we show that we can obtain this as long as Enc is monotone. Specifically, any next-token probability can be formulated as the difference in the CDF of two elements of the output distribution $D_{\mathcal{M}}$. And, by our construction where we assign any element x a probability of $2^{-\mathrm{len}(\mathsf{Enc}(x))}$, we can show that sampling from the distribution of encoded outputs must be equivalent to sampling uniformly random bits until a complete codeword is generated, which means that the CDF of such an encoded output essentially amounts to the binary representation of the respective codeword. Thus, as long as Enc is efficient, we can compute any CDF, and thus any next-token probability, efficiently.

Comparison with Delétang et al. (2024). Delétang et al. (2024) present a heuristic construction of next-token prediction from compression; however, in contrast to our construction, which "looks ahead" and retrieves next-token probabilities relative to the *complete output distribution* reconstructed from the compression, they instead compute next-token probabilities myopically by examining the difference in length between the compression of the prefix and the compression of the prefix with each possible subsequent token appended.[4] The advantage of this construction is that, unlike ours, monotonicity of the compression scheme Enc is not needed. However (as the authors themselves acknowledge in their work), because their approach approximates the next-token distribution using partial compressions every time they sample a next token, it risks introducing a constant amount of error to the sampling process every time a token is generated, whereas, with our

[3] We note that Hirahara et al. (2024) demonstrate an analogue of this result for *uniform* algorithms; we, however, focus on non-uniform algorithms, following Haitner et al. (2023).

[4] That is, given some partial output x, they assign each possible next token σ a weight of $2^{-(\mathrm{len}(\mathsf{Enc}(x||\sigma))-\mathrm{len}(\mathsf{Enc}(x)))}$ and sample according to the distribution over all possible next tokens σ with the respective weights.

construction, using a consistent distribution over complete outputs allows us to sample all tokens relative to that output distribution and only incur the error once.[5]

Completeness. A final subtlety in our construction of an autoregressive model is the inherent assumption that, given some encoding Enc, assigning each element x in the support of D a probability of $2^{-\mathsf{len}(\mathsf{Enc}(x))}$ results in a complete distribution. This is not necessarily the case given an arbitrary Enc (for instance, consider an encoding that transforms some other compression scheme by padding it with n copies of 0). However, since any compression scheme is prefix-free, Kraft's inequality (McMillan, 1956) guarantees that it has $\sum_{x \in \sup(D)} 2^{-\mathsf{len}(\mathsf{Enc}(x))} \leq 1$; if equality holds, the model we construct from Enc will be well-formed as the probabilities we assign sum to 1 by construction. We refer to this property—that is, Kraft's inequality being tight for a given Enc—as *completeness*. We show in Appendix B that it can be assumed essentially without loss of generality: any monotone efficient compression scheme can be "compacted" into one that satisfies completeness without lengthening any codewords or sacrificing monotonicity, efficiency, or prefix-freeness.

2 Preliminaries

We recall some basic cryptographic and coding-theoretic definitions in Appendix A.1, in addition to providing proofs of facts deferred from this section.

2.1 Compression

Definition 2.1 ((Goldberg & Sipser, 1985; Trevisan et al., 2005)). *Given finite families of alphabets $\Sigma = \{\Sigma_n\}_{n \in \mathbb{N}}$ and $\Sigma' = \{\Sigma'_n\}_{n \in \mathbb{N}}$ where $|\Sigma_n|$ and $|\Sigma'_n|$ are polynomial in n, functions $m(\cdot), \ell(\cdot)$ where $m(n)$ is polynomial in n, and an ensemble of distributions $\mathcal{D} = \{D_n\}_{n \in \mathbb{N}}$, where $\sup(D_n) \subseteq \Sigma_n^{m(n)}$, we say that a family of injective functions $\mathsf{Enc} = \{\mathsf{Enc}_n\}_{n \in \mathbb{N}}$ is an **encoding**, or **compression scheme**[6]. Enc **compresses $\mathcal{D}$ to length $\ell(\cdot)$** if, for any $n \in \mathbb{N}$, $\mathsf{Enc}_n : \Sigma_n^{m(n)} \to \Sigma_n'^*$ is such that*

$$\mathbf{E}_{x \leftarrow D_n}[\mathit{len}(\mathsf{Enc}_n(x))] \leq \ell(n)$$

Unless otherwise specified, we shall assume that the output of the compression is binary ($\Sigma'_n = \{0,1\}$ for all $n \in \mathbb{N}$). We also define some useful properties of compression functions:

Definition 2.2 (Properties of compression). *Given finite alphabets Σ and Σ', let $\mathsf{Enc} : \Sigma^* \to \Sigma'^*$ be a function defined on distribution D with $\sup(D) \subseteq \Sigma^*$.*

*We refer to Enc as **prefix-free** if there do not exist elements $x \neq y$ and $i \in \mathbb{N}$ such that $x \in \sup(D)$, $y \in \sup(D)$, and $\mathsf{Enc}(x) = (\mathsf{Enc}(y))_{\leq i}$ (i.e., $\mathsf{Enc}(x)$ is a prefix of $\mathsf{Enc}(y)$).*

*We refer to a prefix-free encoding Enc as **complete** if, for any $n \in \mathbb{N}$, Kraft's inequality (McMillan, 1956) is tight, i.e.:*

$$\sum_{x \in \sup(D)} |\Sigma'|^{-\mathit{len}(\mathsf{Enc}(x))} = 1$$

*We refer to Enc as **monotone** if there exist orderings $\prec_\Sigma$ over Σ and $\prec_{\Sigma'}$ over Σ' such that, for any $x, y \in D$, $x \prec_{\Sigma^*} y$ if and only if $\mathsf{Enc}(x) \prec_{\Sigma'^*} \mathsf{Enc}(y)$, where $\prec_{\Sigma^*}$ is the lexicographic ordering[7] induced by $\prec_\Sigma$ on Σ^* and respectively for $\prec_{\Sigma'^*}$.*

[5] As an illustrative example, consider a binary distribution D with constant next-token probabilities skewed away from powers of 2: e.g., every token is drawn as 0 with probability 0.6 and 1 with probability 0.4. Since every token is drawn with probabilities weighted by powers of 2 in the sampling method of Delétang et al. (2024), this introduces a constant amount of KL-divergence to the respective next-token distribution every time a token is sampled; as these errors are additive by the chain rule, the resulting cross-entropy loss is linear in the output length.

[6] Oftentimes the term "compression" is used to indicate strictly prefix-free encodings; however, we consider only prefix-free encodings anyway in our results.

[7] i.e., $x \prec_{\Sigma^*} y$ when there exists $i \in \mathbb{N}$ for which $x_{<i} = y_{<i}$ and either $\mathrm{len}(y) > \mathrm{len}(x) = i - 1$ or $x_i \prec_\Sigma y_i$.

We consider any of the above properties to apply to a family of functions $\mathsf{Enc} = \{\mathsf{Enc}_n\}_{n\in\mathbb{N}}$ *as defined in Definition 2.1, where* Enc_n *is defined on a distribution* D_n *for which* $\sup(D_n) \subseteq \Sigma_n^{m(n)}$*, if they apply to* Enc_n *and* D_n *for each* $n \in \mathbb{N}$*.*

Furthermore, we say that such a family of functions has ***efficient encoding*** *if there exists a polynomial* $p(\cdot)$ *such that* Enc_n *runs in time at most* $p(n)$ *for any* $n \in \mathbb{N}$*, and* ***efficient decoding*** *if there exists a polynomial* $p'(\cdot)$ *and a function family* $\mathsf{Dec} = \{\mathsf{Dec}_n\}_{n\in\mathbb{N}}$ *such that, for each* $n \in \mathbb{N}$*,* Dec_n *computes the inverse* $\mathsf{Enc}_n^{-1}(c)$ *of* Enc_n *for any codeword in its image (or returns* $\perp$ *if the input is not in the image) and additionally runs in time at most* $p'(n)$*.*

We also note the following useful fact about efficient encoding and decoding:

Fact 2.3 (Proof deferred to Appendix A.1)**.** *Any monotone and efficiently encodable family of encodings* $\mathsf{Enc} = \{\mathsf{Enc}_n\}_{n\in\mathbb{N}}$ *is also efficiently decodable.*

We shall generally restrict our focus to monotone and prefix-free encodings. Shannon's source coding theorem (Shannon, 1948) implies a lower bound for the expected length of any prefix-free encoding Enc over a source distribution D in terms of its Shannon entropy $H(D)$:

Theorem 2.4 ((Shannon, 1948))**.** *Given finite alphabets* Σ *and* Σ' *and distribution* D *over* Σ*, let* $\ell^*(D)$ *be the minimal expected length* $\mathbf{E}_{x\leftarrow D}[len(\mathsf{Enc}(x))]$ *over any uniquely decodable encoding* $\mathsf{Enc} : \Sigma^* \to \Sigma'^*$*. Then:*

$$\frac{H(D)}{\log|\Sigma|} \le \ell^*(D) < \frac{H(D)}{\log|\Sigma|} + 1$$

Since any prefix-free encoding over Σ is uniquely decodable over Σ^*, it follows that:

Corollary 2.5. *Given finite alphabets* Σ *and* Σ' *and distribution* D *over* Σ*, let* $\ell^*(D)$ *be the minimal expected length* $\mathbf{E}_{x\leftarrow D}[len(\mathsf{Enc}(x))]$ *over any prefix-free encoding* $\mathsf{Enc} : \Sigma \to \Sigma'$*. Then:*

$$\frac{H(D)}{\log|\Sigma|} \le \ell^*(D)$$

For binary encodings ($\Sigma' = \{0,1\}$), this bound simply becomes $\ell^*(D) \ge H(D)$.

2.1.1 Arithmetic Encoding

Definition 2.6 (Arithmetic encoding (Shannon, 1948; Gilbert & Moore, 1959))**.** *Let* D *be a discrete distribution and let* $\prec$ *be a total order over* $\sup(D)$*. Let* $F : \sup(D) \to [0,1]$ *by* $F(x) = (\sum_{a\prec x} D(a)) + 1/2 \cdot D(x)$*. Then the* ***arithmetic encoding*** $\mathsf{Enc} : \sup(D) \to \{0,1\}^*$ *is defined such that, for* $x \in \sup(D)$*,* $\mathsf{Enc}(x)$ *is the first* $(\lceil -\log D(x)\rceil + 1)$ *bits of* $F(x)$*.*

Arithmetic encoding is monotone and prefix-free (crucial properties for our results), and additionally, given Shannon's lower bound, provides nearly optimal data compression:

Fact 2.7 (Properties of arithmetic encoding, lemma 2.8 of Trevisan et al. (2005))**.** *For every distribution family* $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ *and ordering* $\prec$ *over the corresponding input alphabet* Σ_n*, the functions* Enc_n *given by applying Definition 2.6 to* D_n *and* $\prec$ *(for any* $n \in \mathbb{N}$*) are one-to-one and monotone, and the scheme* $\mathsf{Enc} = \{\mathsf{Enc}_n\}_{n\in\mathbb{N}}$ *is prefix-free and compresses* $\mathcal{D}$ *to length* $H(D_n) + 2$*.*

Two properties missing from arithmetic encoding as defined above are efficient encoding and completeness. Indeed, these properties do not necessarily hold unconditionally. That said, we claim the following:

Claim 2.8 (Proof deferred to Appendix A.1)**.** *For some distribution family* $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ *and ordering* $\prec$*, if the* cumulative distribution function $\sum_{a\preceq x} D_n(a)$ *can be efficiently computed, then there exists a prefix-free, monotone, efficiently computable, and complete encoding* $\mathsf{Enc} = \{\mathsf{Enc}_n\}_{n\in\mathbb{N}}$ *that additionally compresses* $\mathcal{D}$ *to length* $H(D_n) + 2$*.*

2.1.2 KL-divergence and Compression Loss

One intuitive interpretation of the KL-divergence $KL(D \mid\mid D')$ between two distributions is as the expected number of additional bits that would be needed to encode samples from D using compression optimized for the distribution D' as opposed to using compression optimized for D. We restate

this for schemes, such as arithmetic encoding, which are "close to optimal" in the sense that the encoding length of a sample is additively close to the empirical entropy of that sample:

Claim 2.9 (Proof deferred to Appendix A.1). *Let* $\mathsf{Enc} = \{\mathsf{Enc}_D\}$ *be an encoding parameterized by a distribution* D *such that, for any distribution* D *and any element* $x \in \sup(D)$*:*

$$\textit{len}(\mathsf{Enc}_D(x)) \leq \log(1/D(x)) + \alpha$$

Then:

$$\mathbf{E}_{x \leftarrow D}[\textit{len}(\mathsf{Enc}_{D'}(x))] \leq H(D) + KL(D \mid\mid D') + \alpha$$

By definition, the family $\mathsf{Enc} = \{\mathsf{Enc}_D\}$ where Enc_D is the arithmetic encoding relative to D has this property for $\alpha = 2$ and thus satisfies:

$$\mathbf{E}_{x \leftarrow D}[\mathsf{len}(\mathsf{Enc}_{D'}(x))] \leq H(D) + KL(D \mid\mid D') + 2$$

2.2 Cross Entropy and LLM Training Objectives

We recall the definition of cross-entropy:

Definition 2.10. *The* ***cross-entropy*** *of a distribution* D_2 *relative to a distribution* D_1 *is given by:*

$$H(D_1, D_2) = -\mathbf{E}_{x \leftarrow D_1}[\log D_2(x)] = -\sum_{x \in \sup(D_1)} D_1(x) \log D_2(x)$$

Cross-entropy is closely related to KL-divergence by the following fact:

Fact 2.11. *For any distributions* D_1, D_2*:*

$$H(D_1, D_2) = KL(D_1 || D_2) + H(D_1)$$

2.2.1 Autoregressive Models

We shall generally treat LLMs in a black-box way and consider the process of training an LLM in terms of the objective function that it attempts to optimize. Let us assume that an LLM is being trained on some distribution D over sequences of tokens $t \in \Sigma^*$ from a vocabulary Σ. During pretraining, the majority of the training process, an LLM is given samples from the distribution D and attempts to "learn" the distribution D by building an *autoregressive model* $\mathcal{M}$ that maps any sequence of tokens $t_{<i} \in \Sigma^*$ to a distribution $\mathcal{M}_{t_{<i}}$ over next tokens that assigns each possible next token t_i some probability $\mathcal{M}(t_i \mid t_{<i})$. Formally:

Definition 2.12. *Given an alphabet* Σ *and a maximum output length* m*,*[8] *we define an* ***autoregressive model*** $\mathcal{M}$ *over* Σ^m *to be a collection of distributions* $\{\mathcal{M}_t\}_{t \in \Sigma^{<m}}$ *where every distribution* $\mathcal{M}_t$ *has support* Σ*. We define the induced distribution* $D_{\mathcal{M}}$ *to be the distribution of complete outputs resulting from the process of starting with an empty string* t *and iteratively appending a sample from* $\mathcal{M}_t$ *to* t m *times to construct a length-*m *string* $t \in \Sigma^m$*.*

The objective of this training process, then, is to minimize the expectation over samples $t \leftarrow D$ of the loss function

$$L(\mathcal{M}, t) = -\sum_{i \in [\mathsf{len}(t)]} \log \mathcal{M}(t_i \mid t_{<i}) \tag{1}$$

which is known as *cross-entropy loss* or *negative log-likelihood*. We refer to an autoregressive model as δ*-optimal* with respect to a distribution D if, for any autoregressive model $\mathcal{M}^*$ defined over the same alphabet:

$$\mathbf{E}_{t \leftarrow D}[L(\mathcal{M}, t)] - \mathbf{E}_{t \leftarrow D}[L(\mathcal{M}^*, t)] \leq \delta$$

that is, the expected loss of $\mathcal{M}$ over D is no more than δ worse than the minimum achievable loss. An important observation about this loss function is the following:

Claim 2.13 (Proof deferred to Appendix A.1). *For alphabet* Σ*, maximum length* m*, and a fixed training distribution* D *with support* $\sup(D) \subseteq \Sigma^m$*, if there exist an autoregressive model* $\mathcal{M}$ *over* Σ^m *and constant* δ *such that* $\mathcal{M}$ *is* δ*-optimal with respect to* D*, then, for any autoregressive model* $\mathcal{M}^*$ *defined over* Σ^m*:*

$$KL(D \mid\mid D_{\mathcal{M}}) - KL(D \mid\mid D_{\mathcal{M}^*}) \leq \delta$$

[8] We can assume without loss of generality that outputs of length less than m are padded to length m by a special End token.

This intuitively means that training an LLM by attempting to minimize the expected loss $\mathbf{E}_{t \leftarrow D}[L(\mathcal{M}, t)]$ is equivalent to minimizing the KL-divergence $KL(D \mid\mid D_{\mathcal{M}})$.

One final result that will be useful to us is demonstrating that the cumulative distribution function of a distribution $D_{\mathcal{M}}$ induced by an autoregressive model $\mathcal{M}$ can be efficiently computed:

Claim 2.14 (Proof deferred to Appendix A.1)**.** *Given a polynomial-sized alphabet family $\Sigma = \{\Sigma_n\}_{n \in \mathbb{N}}$ with lexicographical ordering $\prec$, polynomial maximum length $m(\cdot)$, and family of autoregressive models $\mathcal{M} = \{\mathcal{M}_n\}_{n \in \mathbb{N}}$ over $\Sigma_n^{m(n)}$, the following statements are equivalent:*

1. *For any $t \in \Sigma_n^{m(n)}$, $\mathcal{M}_n(t_i \mid t_{<i})$ is efficiently computable.*
2. *For any $n \in \mathbb{N}$ and $a \in \Sigma_n^{\leq m(n)}$, the cumulative distribution function $\Pr_{x \leftarrow D_{\mathcal{M}_n}}[x \preceq a]$ is efficiently computable.*

3 Results

3.1 Autoregressive Modeling and Monotone Compression

We first state our main result equating autoregressive modeling to monotone compression. First, given an autoregressive model $\mathcal{M}$ for a given distribution such that the expectation of $L(\mathcal{M}, t)$ over samples from the distribution is close to its optimum, one can use a near-optimal encoding scheme (e.g., arithmetic encoding) to construct efficient compression that is similarly near-optimal with respect to coding length.

Theorem 3.1. *Given an alphabet Σ and maximum length m, let $\mathsf{Enc}^{(\cdot)}$ be an oracle-aided[9] algorithm such that there exists constant α for which, given any autoregressive model $\mathcal{M}^*$ over Σ^m, $\mathsf{Enc}^{\mathcal{M}^*}(\cdot)$ is a prefix-free encoding and $len(\mathsf{Enc}^{\mathcal{M}^*}(x)) \leq \log(1/D(x)) + \alpha$. Then, for any autoregressive model $\mathcal{M}$ over Σ^m, distribution D with support $\sup(D) \subseteq \Sigma^m$, and constant δ such that $\mathcal{M}$ is δ-optimal with respect to D,*

$$\mathbf{E}_{x \leftarrow D}[len(\mathsf{Enc}^{\mathcal{M}}(x))] \leq H(D) + \delta + \alpha$$

We can instantiate this using Claim 2.8, which gives $\alpha = 2$, for the following corollary, noting that Claim 2.14 gives the requisite property that the cumulative distribution function is efficiently computable:

Corollary 3.2. *Given a polynomial-sized alphabet family $\Sigma = \{\Sigma_n\}_{n \in \mathbb{N}}$ and polynomial maximum length $m(\cdot)$, there exists an oracle-aided algorithm $\mathsf{Enc}^{(\cdot)}$ so that, for any family of distributions $\mathcal{D} = \{D_n\}_{n \in \mathbb{N}}$, family of autoregressive models $\mathcal{M} = \{\mathcal{M}_n\}_{n \in \mathbb{N}}$ over $\Sigma_n^{m(n)}$ where $\mathcal{M}_n(t_i \mid t_{<i})$ is efficiently computable for any $t \in \Sigma_n^{\leq m(n)}$, and function $\delta(\cdot)$ so that, for any $n \in \mathbb{N}$, $\mathcal{M}_n$ is $\delta(n)$-optimal with respect to D_n, the family of encodings $\mathsf{Enc} = \{\mathsf{Enc}^{\mathcal{M}_n}(1^n, \cdot)\}_{n \in \mathbb{N}}$ is prefix-free, monotone, and efficient, and additionally, for any $n \in \mathbb{N}$:*

$$\mathbf{E}_{x \leftarrow D_n}[len(\mathsf{Enc}^{\mathcal{M}_n}(1^n, x))] \leq H(D_n) + \delta(n) + 2$$

Proof (Theorem 3.1). By Claim 2.13, we have $KL(D \mid\mid D_{\mathcal{M}}) \leq \delta$ (as the optimum is attained at $D = D_{\mathcal{M}}$ which results in a divergence of 0). By Claim 2.9:

$$\mathbf{E}_{x \leftarrow D}[\mathsf{len}(\mathsf{Enc}^{\mathcal{M}}(x))] \leq H(D) + KL(D \mid\mid D_{\mathcal{M}}) + \alpha \leq H(D) + \delta + \alpha$$

□

Next, we state the other direction: if one has an encoding that compresses a given distribution near-optimally, it is possible to create an autoregressive model that is close to optimal in terms of the expected next-token cross entropy loss $\mathbf{E}_{t \leftarrow D}[L(\mathcal{M}, t)]$. Formally:

[9] We abuse notation slightly by denoting, by a model $\mathcal{M}$ passed as an oracle to Enc, an oracle that, given input $(t_i, t_{<i})$ for some $t_{\leq i} \in \Sigma^i$, returns the next-token probability $\mathcal{M}(t_i \mid t_{<i})$. In Corollary 3.2, the respective oracle implements the same functionality analogously for the case of a family parameterized by $n \in \mathbb{N}$ given an additional input 1^n denoting the parameter.

Theorem 3.3 (Proof deferred to Appendix A.2.1)**.** *Given a polynomial-sized alphabet family $\Sigma = \{\Sigma_n\}_{n\in\mathbb{N}}$ and polynomial maximum length $m(\cdot)$, there exists an oracle-aided algorithm $\mathcal{M}^{(\cdot)}$ such that, for any family of distributions $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$, function $\delta(\cdot)$, and prefix-free, monotone, and efficiently encodable family of encodings $\mathsf{Enc} = \{\mathsf{Enc}_n\}_{n\in\mathbb{N}}$ where, for any $n \in \mathbb{N}$,*

$$\mathbf{E}_{x\leftarrow D_n}[\mathit{len}(\mathsf{Enc}_n(x))] \leq H(D_n) + \delta(n)$$

$\mathcal{M}^{(\cdot)}$ defines a family of autoregressive models $\{\mathcal{M}_n\}_{n\in\mathbb{N}}$ where, for any $n \in \mathbb{N}$ and $t \in \Sigma_n^{\leq m(n)}$, $\mathcal{M}_n(t_i \mid t_{<i})$ is given by $\mathcal{M}^{\mathsf{Enc}}(1^n, t_i, t_{<i})$, $\mathcal{M}^{(\cdot)}$ is efficiently computable, and the respective model $\mathcal{M}_n$ is $\delta(n)$-optimal with respect to D_n.

3.2 The Monotonicity Requirement and Infinitely-Often One-Way Functions

In contrast to Theorem 3.3, which shows that any *monotone* compression scheme for a distribution can be transformed into a next-bit predictor for that distribution whose negative log-likelihood is identical to the compression loss, we next show that, assuming the existence of infinitely-often one-way functions, there is a family of distributions $\mathcal{D}$ and a *non-monotone* compression scheme that is *optimal* for $\mathcal{D}$ but is such that no efficiently computable autoregressive model exists that can predict $\mathcal{D}$ with even an arbitrary constant negative log-likelihood:

Theorem 3.4 (Proof deferred to Appendix A.3.2)**.** *Assuming the existence of one-way functions (respectively, infinitely-often one-way functions), there exists an efficiently samplable family of distributions $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ and a prefix-free family of encodings $\mathsf{Enc} = \{\mathsf{Enc}_n\}_{n\in\mathbb{N}}$ with efficient encoding and decoding such that $\mathbf{E}_{x\leftarrow D_n}[\mathit{len}(\mathsf{Enc}_n(x))] = H(D_n)$ for any $n \in \mathbb{N}$, but, given any efficiently computable family of autoregressive models $\{\mathcal{M}_n\}_{n\in\mathbb{N}}$, for any constant $\epsilon > 0$ and all sufficiently large (resp., infinitely many) $n \in \mathbb{N}$, the respective output distribution $D_{\mathcal{M}_n}$ is such that $KL(D_n \mid\mid D_{\mathcal{M}_n}) \geq (1-\epsilon)n$.*[10]

However, if infinitely-often one-way functions do *not* exist, we can instead show the following corollary of a theorem from Impagliazzo & Luby (1989), which we present and discuss further in Appendix A.3.3, that amounts to showing an analogue of Theorem 3.3 for (non-monotone) compression, albeit with an additional loss of $\epsilon(n)$ statistical distance for some $\epsilon(n) \in 1/\mathsf{poly}(n)$:

Corollary 3.5. *If infinitely-often one-way functions do not exist, then the following is true:*

For any $\epsilon(n) \in 1/\mathsf{poly}(n)$, there exists an efficient oracle-aided algorithm $\mathcal{M}^{(\cdot)}$ such that, given a family $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ of efficiently samplable distributions and a compression scheme $(\mathsf{Enc}, \mathsf{Dec})$ (with efficient encoding and decoding) that is δ-optimal for $\mathcal{D}$, $\mathcal{M}^{(\cdot)}$ defines a family of autoregressive models $\{\mathcal{M}_n\}_{n\in\mathbb{N}}$ where, for any $n \in \mathbb{N}$ and $t \in \Sigma_n^{\leq m(n)}$, $\mathcal{M}_n(t_i \mid t_{<i})$ is given by $\mathcal{M}^{\mathsf{Enc}}(1^n, t_i, t_{<i})$, $\mathcal{M}^{(\cdot)}$ is efficiently computable, and the respective model $\mathcal{M}_n$ is $\epsilon(n)$-approximately $\delta(n)$-optimal with respect to D_n.

3.3 Pseudoentropy and Incompressibility

Lastly, we demonstrate an interesting application of our results to cryptography—specifically, the relation between the compressibility of a distribution and its *next-bit pseudoentropy* (a computational analogue of entropy). Haitner et al. (2023) showed that incompressibility of a distribution implies high next-bit pseudoentropy, but that the converse is false. We provide a new characterization showing that, as a corollary of Theorems 3.1 and 3.3, *monotone* incompressibility is *equivalent* to high next-bit pseudoentropy.

Corollary 3.6. *Given polynomial maximum length $m(\cdot)$ and family of distributions $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ where D_n is a distribution over $\{0,1\}^{m(n)}$, if, for some function $k(\cdot)$, $\mathcal{D}$ cannot be monotonically compressed to length $k(n)$, then $\mathcal{D}$ has next-bit pseudoentropy at least $k(n) - 2$.*

Corollary 3.7. *Given polynomial maximum length $m(\cdot)$ and family of distributions $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ where D_n is a distribution over $\{0,1\}^{m(n)}$, if, for some function $k(\cdot)$, $\mathcal{D}$ has next-bit pseudoentropy at least $k(n)$, then $\mathcal{D}$ cannot be monotonically compressed to length $k(n)$.*

We discuss these results further in Appendix A.4.

[10] This implies that, for any constant δ, $\mathcal{M}$ cannot be δ-optimal with respect to $\mathcal{D}$.

GENERATIVE AI USE DISCLOSURE

While finalizing this paper for submission, we used generative AI tools to review our draft in order to fix typographical errors and suggest improvements (which were subsequently implemented by the authors) to the structure and readability of our introduction. We emphasize that all results and writing presented in this paper were produced by the authors; in other words, we did not use AI tools for any tasks which would *require* disclosure.

REFERENCES

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jznbgiynus.

Eibe Frank, Chang Chui, and Ian H. Witten. Text categorization using compression models. In *Proceedings DCC 2000. Data Compression Conference*, pp. 555–, 2000. doi: 10.1109/DCC.2000.838202.

E. N. Gilbert and E. F. Moore. Variable-Length Binary Encodings. *The Bell System Technical Journal*, 38(4):933–967, 1959. doi: 10.1002/j.1538-7305.1959.tb01583.x.

Andrew V. Goldberg and Michael Sipser. Compression and Ranking. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pp. 440–448, New York, NY, USA, 1985. Association for Computing Machinery. ISBN 0897911512. doi: 10.1145/22145.22194. URL https://doi.org/10.1145/22145.22194.

Iftach Haitner, Noam Mazor, and Jad Silbak. Incompressiblity and Next-Block Pseudoentropy. In Yael Tauman Kalai (ed.), *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 66:1–66:18, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-263-1. doi: 10.4230/LIPIcs.ITCS.2023.66. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2023.66.

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.

Jose Hernandez-Orallo. A Formal Definition of Intelligence Based on an Intensional Variant of Algorithmic Complexity. In *Proceedings of International Symposium of Engineering of Intelligent Systems (EIS'98)*, pp. 146–163, 07 1998.

Shuichi Hirahara, Zhenjian Lu, and Mikito Nanashima. Optimal Coding for Randomized Kolmogorov Complexity and Its Applications. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 369–378, 2024. doi: 10.1109/FOCS61266.2024.00030.

Yuzhen Huang, Jinghan Zhang, Zifei Shan, and Junxian He. Compression represents intelligence linearly. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=SHMj84U5SH.

Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *30th Annual Symposium on Foundations of Computer Science*, pp. 230–235. IEEE Computer Society, 1989.

Matthew V. Mahoney. Text compression as a Test for Artificial Intelligence. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, pp. 970, USA, 1999. American Association for Artificial Intelligence. ISBN 0262511061.

Matthew V. Mahoney. Fast Text Compression with Neural Networks. In *Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference*, pp. 230–234. AAAI Press, 2000. ISBN 1577351134.

Brockway McMillan. Two inequalities implied by unique decipherability. *IRE Transactions on Information Theory*, 2(4):115–116, 1956. doi: 10.1109/TIT.1956.1056818.

Ionut Schiopu, Yu Liu, and Adrian Munteanu. CNN-based Prediction for Lossless Coding of Photographic Images. In *2018 Picture Coding Symposium (PCS)*, pp. 16–20, 2018. doi: 10.1109/PCS.2018.8456311.

Jürgen Schmidhuber and Stefan Heil. Predictive Coding with Neural Nets: Application to Text Compression. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'94, pp. 1047–1054, Cambridge, MA, USA, 1994. MIT Press.

Jürgen Schmidhuber and Stefan Heil. Sequential Neural Text Compression. *IEEE Transactions on Neural Networks*, 7(1):142–146, 1996. doi: 10.1109/72.478398.

C. E. Shannon. Prediction and entropy of printed English. *The Bell System Technical Journal*, 30 (1):50–64, 1951. doi: 10.1002/j.1538-7305.1951.tb01366.x.

Claude E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

Luca Trevisan, Salil Vadhan, and David Zuckerman. Compression of Samplable Sources. 14(3): 186–227, 2005.

Chen-Han Tsai. Revisiting Data Compression with Language Modeling. *arXiv preprint*, 01 2026. doi: 10.48550/arXiv.2601.02875.

A. M. Turing. Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL `https://doi.org/10.1093/mind/LIX.236.433`.

Salil Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pp. 817–836, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312455. doi: 10.1145/2213977.2214051. URL `https://doi.org/10.1145/2213977.2214051`.

Andrew C. Yao. Theory and Application of Trapdoor Functions. In *23rd Annual Symposium on Foundations of Computer Science (FOCS 1982)*, pp. 80–91, 1982. doi: 10.1109/SFCS.1982.45.

Yeqin Zhang, Yizheng Zhao, Chen Hu, Binxing Jiao, Daxin Jiang, Ruihang Miao, and Cam Tu Nguyen. Learning to Compress: Unlocking the Potential of Large Language Models for Text Representation. In *Proceedings of the Fortieth Annual AAAI Conference on Artificial Intelligence*, 2026. doi: 10.48550/arXiv.2511.17129.

APPENDIX

A DEFERRED PROOFS AND ADDITIONAL DISCUSSION

A.1 PRELIMINARIES

Let $\mathbb{N} = \{1, 2, 3, \ldots\}$ and $[n] = \{m \in \mathbb{N} \mid m \leq n\} = \{1, 2, \ldots, n\}$. For some symbol σ in an alphabet Σ, take σ^n to be the string consisting of n copies of σ. Let $\mathbf{1}_P$ be the *indicator function* of a logical statement P, defined as being 1 when P is true and 0 otherwise. Given a set X, we define $\Delta(X)$ to be the set of probability distributions over elements in X. Given some string $x \in \Sigma^*$ over a vocabulary Σ, we will denote substrings with notation such as $x_{\leq i}$ (representing the prefix consisting of the first i symbols), and the length of a string $x \in \Sigma^*$ by $\text{len}(x)$. We use similar

notation to denote, e.g., $\Sigma^{\leq n} = \bigcup_{i=0}^{n} \Sigma^i$ as the set of strings of tokens from an alphabet Σ with at most length n.

We say a statement is true for "sufficiently large $n \in \mathbb{N}$" if there exists $N \in \mathbb{N}$ such that the statement holds for all $n \geq N$. We shall call a function $\epsilon(\cdot)$ *negligible* if, for any polynomial $p(\cdot)$, $\epsilon(n) < 1/p(n)$ for sufficiently large $n \in N$; conversely, a function is *non-negligible* or *polynomial* if the above is not true. We refer to an algorithm as *efficient* if its running time is polynomial in its input size.

We define an *oracle-aided algorithm* $\Pi^{\mathcal{O}}$ as an algorithm Π that is given access to another algorithm $\mathcal{O}$ as a black box, and may make many calls to its oracle $\mathcal{O}$ on potentially many different inputs before determining its own output. We say that such an algorithm is efficient if its running time (excepting calls made to $\mathcal{O}$) and the number of calls it makes to the oracle $\mathcal{O}$ are both polynomial in its input size—that is, as long as $\mathcal{O}$ is efficient, the total running time of Π, including that taken by calls made to $\mathcal{O}$, is polynomial in its input size.

Unless otherwise indicated, any logarithm denoted by $\log$ will be base 2 (as opposed to the base-e natural logarithm $\ln$). Given a discrete distribution D, we denote by $\sup(D)$ its support and define the (Shannon) *entropy* of D to be the quantity

$$H(D) = \sum_{x \in \sup(D)} -D(x) \log D(x)$$

Given two discrete distributions D, D', we denote the *statistical distance* (or total variation distance) between D and D' by $\Delta(D, D')$ and the Kullback-Leibler (KL) divergence between the two distributions by

$$KL(D||D') = \sum_{x \in \sup(D)} D(x) \log \frac{D(x)}{D'(x)}$$

A.1.1 Proof of Fact 2.3

Proof. Consider a decoding algorithm that, given a codeword, performs binary search over the possible inputs in $\sup(D_n)$. This is possible due to monotonicity—it can be determined whether the correct decoding is lower or higher than a given input by comparing the respective codewords. Additionally, since there are at most $|\Sigma_n|^{m(n)}$ possible inputs in $\sup(D_n)$, binary search requires at most $O(m(n) \log |\Sigma_n|)$ iterations of Enc_n. Since by definition $m(n)$ and $|\Sigma_n|$ are polynomial in n, and since Enc is itself efficient, this decoding process must likewise be efficient. □

A.1.2 Proof of Claim 2.8

Proof. Consider the arithmetic encoding Enc' given in Definition 2.6. If $\sum_{a \preceq x} D_n(a)$ can be efficiently computed, then so can

$$D_n(x) = \sum_{a \preceq x} D_n(a) - \sum_{a \preceq x^-} D_n(a)$$

(where x^- is the predecessor of x under $\prec$) and consequently

$$\sum_{a \prec x} D_n(a) = \sum_{a \preceq x} D_n(a) - D(x)$$

and thus $\mathsf{Enc}'(x)$ must be efficiently computable as well.

Completeness can be obtained by applying Claim B.2, which shows that Enc' may be transformed into a complete encoding Enc while preserving monotonicity, prefix-freeness, and efficient encoding, and additionally not increasing the length of any codeword. □

A.1.3 PROOF OF CLAIM 2.9

Proof. By the definition of KL-divergence:

$$\begin{aligned} KL(D \mid\mid D') &= \mathbf{E}_{x \leftarrow D}[\log(1/D'(x))] - \mathbf{E}_{x \leftarrow D}[\log(1/D(x))] \\ &\geq (\mathbf{E}_{x \leftarrow D}[\text{len}(\mathsf{Enc}_{D'}(x))] - \alpha) - H(D) \end{aligned}$$

which directly implies the conclusion. □

A.1.4 PROOF OF CLAIM 2.13

Proof. Let m be the maximum output length of a sample from D. We can apply the chain rule and Fact 2.11 to show:

$$\begin{aligned} KL(D \mid\mid D_{\mathcal{M}}) &= \sum_{i \in [m]} \mathbf{E}_{t_{<i}:t \leftarrow D}[KL(D_{t_{<i}} \mid\mid \mathcal{M}_{t_{<i}})] \\ &= \sum_{i \in [m]} \mathbf{E}_{t_{<i}:t \leftarrow D}[H(D_{t_{<i}}, \mathcal{M}_{t_{<i}}) - H(D_{t_{<i}})] \end{aligned}$$

where $D_{t_{<i}}$ is defined analogously to $\mathcal{M}_{t_{<i}}$ (i.e., the next-token distribution given prefix $t_{<i}$) for the training distribution D. Furthermore:

$$\begin{aligned} &\sum_{i \in [m]} \mathbf{E}_{t_{<i}:t \leftarrow D}[H(D_{t_{<i}}, \mathcal{M}_{t_{<i}})] \\ &= -\sum_{i \in [m]} \mathbf{E}_{t_{<i}:t \leftarrow D}[\mathbf{E}_{t_i \leftarrow D_{t_{<i}}}[\log \mathcal{M}(t_i \mid t_{<i})]] \\ &= -\sum_{i \in [m]} \mathbf{E}_{t_{\leq i}:t \leftarrow D}[\log \mathcal{M}(t_i \mid t_{<i})] \\ &= \mathbf{E}_{t \leftarrow D}\left[-\sum_{i \in [m]} \log \mathcal{M}(t_i \mid t_{<i})\right] = \mathbf{E}_{t \leftarrow D}\left[-\sum_{i \in [\text{len}(t)]} \log \mathcal{M}(t_i \mid t_{<i})\right] = \mathbf{E}_{t \leftarrow D}[L(\mathcal{M}, t)] \end{aligned}$$

Importantly, since the training distribution D is fixed, the entropy $H(D_{t_{<i}})$ is constant, as is its expectation over $t \leftarrow D$, which provides that $KL(D \mid\mid D_{\mathcal{M}})$ and $\mathbf{E}_{t \leftarrow D}[L(\mathcal{M}, t)]$ are separated by the same additive constant for any model $\mathcal{M}$, implying the conclusion since we have

$$\mathbf{E}_{t \leftarrow D}[L(\mathcal{M}, t)] - \mathbf{E}_{t \leftarrow D}[L(\mathcal{M}^*, t)] \leq \delta$$

by assumption of δ-optimality. □

A.1.5 PROOF OF CLAIM 2.14

Proof. (1) → (2). Consider the following procedure for computing $\Pr_{x \leftarrow D_{\mathcal{M}_n}}[x \preceq a]$:

- Let $p = 0$, $p' = 1$, and $i = 1$.
- Let $M = (\mathcal{M}_n)_{a_{<i}}$, the next-token distribution for the ith token of a.
- Set $p \leftarrow p + p'(\sum_{t \in \text{sup}(M): t \prec a_i} M(t))$ and set $p' = p' \cdot M(a_i)$.
- Repeat the previous two steps incrementing i until $i = m(n)$, then take $p \leftarrow p + p'$, terminate and return p.

Correctness holds because each iteration adds the probability that $x \leftarrow D_{\mathcal{M}_n}$ has $x_{<i} = a_{<i}$ but $x_i \prec a_i$ (and the final step $p \leftarrow p + p'$ adds the probability that $x = a$), and the sum of those probabilities is precisely $\Pr_{x \leftarrow D_{\mathcal{M}_n}}[x \preceq a]$.

Efficiency holds because, since $|\Sigma|$ is polynomial, we compute $M(t) = \mathcal{M}_n(t \mid a_{<i})$ at most a polynomial number of times in each iteration of the loop, which itself iterates a polynomial number $p(n)$ of times.

(2) → (1). This follows since $\mathcal{M}_n(t_i \mid t_{<i}) = \Pr_{x \leftarrow D_{\mathcal{M}_n}}[x \preceq t^{\top}] - \Pr_{x \leftarrow D_{\mathcal{M}_n}}[x \preceq t^{\perp}]$, where $t^{\top}$ is the $\prec$-greatest element of $\Sigma_n^{m(n)}$ with $t_{<i}$ as a prefix and $t^{\perp}$ is the $\prec$-greatest element of $\Sigma_n^{m(n)}$ with a prefix $(t^{\perp})_{<i} \prec t_{<i}$. □

A.2 Autoregressive Modeling and Monotone Compression

A.2.1 Proof of Theorem 3.3

Proof. (For ease of notation, we omit the parameter n and demonstrate the following for any $n \in \mathbb{N}$.) First, assume that Enc is complete; this is without loss of generality because, if it is not, we may by Claim B.2 replace it with an encoding that is complete without increasing the length of any codeword. Now, define a distribution U_{Enc} to have the same support as the distribution $D_{\mathsf{Enc}} = \{x \leftarrow D : \mathsf{Enc}(x)\}$ of encoded samples from D; however, let $U_{\mathsf{Enc}}(x) = 2^{-\mathrm{len}(x)}$ for all x in its support. Since we assumed Enc is complete, this defines a complete probability distribution over its support. Additionally:

$$H(D_{\mathsf{Enc}}, U_{\mathsf{Enc}}) = \mathbf{E}_{x \leftarrow D_{\mathsf{Enc}}}[\log 1/U_{\mathsf{Enc}}(x)] = \mathbf{E}_{x \leftarrow D_{\mathsf{Enc}}}[\mathrm{len}(x)] = \mathbf{E}_{x \leftarrow D}[\mathrm{len}(\mathsf{Enc}(x))]$$
$$\leq H(D) + \delta$$

which, since it is also the case that

$$H(D_{\mathsf{Enc}}, U_{\mathsf{Enc}}) = KL(D_{\mathsf{Enc}} \mid\mid U_{\mathsf{Enc}}) + H(D_{\mathsf{Enc}}) = KL(D_{\mathsf{Enc}} \mid\mid U_{\mathsf{Enc}}) + H(D)$$

implies that $KL(D_{\mathsf{Enc}} \mid\mid U_{\mathsf{Enc}}) \leq \delta$. Define $\mathcal{M}^{\mathsf{Enc}}$ by:

$$\mathcal{M}^{\mathsf{Enc}}(t_i, t_{<i}) = \mathcal{M}(t_i \mid t_{<i}) = \frac{\sum_{x \in \mathrm{sup}(D)} (U_{\mathsf{Enc}}(\mathsf{Enc}(x)) \mathbf{1}_{x_{\leq i} = t_{\leq i}})}{\sum_{x \in \mathrm{sup}(D)} (U_{\mathsf{Enc}}(\mathsf{Enc}(x)) \mathbf{1}_{x_{<i} = t_{<i}})}$$

$$= \frac{\sum_{x \in \mathrm{sup}(D)} (2^{-\mathrm{len}(\mathsf{Enc}(x))} \mathbf{1}_{x_{\leq i} = t_{\leq i}})}{\sum_{x \in \mathrm{sup}(D)} (2^{-\mathrm{len}(\mathsf{Enc}(x))} \mathbf{1}_{x_{<i} = t_{<i}})}$$

that is, the probability that, given prefix $t_{<i}$, an output will be selected with the next token t_i if we, instead of sampling from D, sample a random encoding x' from the distribution U_{Enc} and select $\mathsf{Dec}(x')$ (i.e., $\mathsf{Enc}^{-1}(x')$, which is well-defined and one-to-one since we assume Enc to be prefix-free). That is, the output distribution $D_{\mathcal{M}}$ is given by $D_{\mathcal{M}} = \{x' \leftarrow U_{\mathsf{Enc}} : \mathsf{Dec}(x')\}$. Additionally, $D = \{x' \leftarrow D_{\mathsf{Enc}} : \mathsf{Dec}(x')\}$, which since Dec is one-to-one implies that

$$KL(D \mid\mid D_{\mathcal{M}}) = KL(D_{\mathsf{Enc}} \mid\mid U_{\mathsf{Enc}}) \leq \delta$$

Now recall from the proof of Claim 2.13 that:

$$\begin{aligned} KL(D \mid\mid D_{\mathcal{M}}) &= \sum_{i \in [m]} \mathbf{E}_{t_{<i}: t \leftarrow D}[KL(D_{t_{<i}} \mid\mid \mathcal{M}_{t_{<i}})] \\ &= \sum_{i \in [m]} \mathbf{E}_{t_{<i}: t \leftarrow D}[H(D_{t_{<i}}, \mathcal{M}_{t_{<i}}) - H(D_{t_{<i}})] \\ &= c + \sum_{i \in [m]} \mathbf{E}_{t_{<i}: t \leftarrow D}[H(D_{t_{<i}}, \mathcal{M}_{t_{<i}})] \\ &= c + \mathbf{E}_{t \leftarrow D}[L(\mathcal{M}, t)] \end{aligned}$$

where we may consider c as a constant since it only depends on D. Since $KL(D \mid\mid D_{\mathcal{M}})$ is non-negative, the minimal value ℓ^* of $\mathbf{E}_{t \leftarrow D}[L(\mathcal{M}, t)]$ is at least $-c$. But, since we showed that $KL(D \mid\mid D_{\mathcal{M}}) \leq \delta$:

$$\mathbf{E}_{t \leftarrow D}[L(\mathcal{M}, t)] \leq -c + \delta \leq \ell^* + \delta$$

as desired. It remains to show that $\mathcal{M}$ is efficiently computable, which we claim and prove to complete the proof of the theorem:

Claim A.1. *For any $n \in \mathbb{N}$, $i \leq n$, and $t_{\leq i} \in \Sigma_n^i$, the next-token probability $\mathcal{M}_n(t_i \mid t_{<i})$ (and thus the model $\mathcal{M}_{t_{<i}}$ for any $t_{<i} \in \Sigma_n^{<m(n)}$) can be computed in time polynomial in n.*

Proof. (We again omit the parameter n and demonstrate the following for any $n \in \mathbb{N}$.) By Claim 2.14, it suffices to show that we can efficiently compute $\Pr_{x \leftarrow D_{\mathcal{M}}}[x \preceq y]$, or equivalently (by monotonicity) $\Pr_{x \leftarrow U_{\mathsf{Enc}}}[x \leq \mathsf{Enc}(y)]$, for any $y \in \Sigma^{m(n)}$, which we can demonstrate due to the assumed structure of Enc and the induced distribution U_{Enc}. Specifically, consider the following protocol for efficiently sampling from U_{Enc}:

- Initialize c as the empty string.
- For $i \in [n]$:
  - Sample a uniformly random bit $b \leftarrow \{0, 1\}$.
  - Append b to c.
  - If c decodes successfully (Dec returns a valid string $m \in \Sigma^*$), return m and terminate.

Correctness follows from the fact that it clearly assigns any string m with some encoding x a probability of $2^{-\text{len}(x)}$. We can leverage the observation that sampling from U_{Enc} is effectively identical to sampling from a truncated uniform distribution to create an efficient protocol for computing $\Pr_{x \leftarrow U_{\mathsf{Enc}}}[x \leq \mathsf{Enc}(y)]$ by essentially binary search. Consider the following recursive algorithm:

- Input $c \in \{0, 1\}^*$. Let c be the empty string if at the top level of recursion.
- Let $m = \mathsf{Dec}(c)$. If $m \neq \perp$, return 1 if $m \preceq y$, and 0 if not.
- Find c^*, the smallest valid encoding such that c^* has $c||1$ as a prefix.[11]
- If $\mathsf{Dec}(c^*) \succ y$, recurse with $c = c||0$ and return $r/2$, where r is the result.
- Else recurse with $c = c||1$ and return $(r + 1)/2$, where r is the result.

Correctness follows by showing inductively that for any input c this algorithm returns $\Pr_{x \leftarrow U_{\mathsf{Enc}}}[x \leq \mathsf{Enc}(y)]$ conditioned on x having prefix c. The base case (c is a complete encoding, i.e., $m \neq \perp$) is trivial; recursion follows from the observation we made above that prefixes $c||0$ and $c||1$ are equally likely to occur. Then, if $\mathsf{Dec}(c^*) \succ y$, any encoding c' with prefix $c||1$ cannot have $c' \leq \mathsf{Enc}(y)$ by monotonicity (and thus the desired probability for prefix $c||1$ becomes 0); symmetrically, if $\mathsf{Dec}(c^*) \preceq y$, any encoding c' with prefix $c||0$ *must* have $c' \leq \mathsf{Enc}(y)$ (since $c' \preceq c^*$), making the desired probability 1 for that prefix. Thus we need to recursively run the algorithm on only one of the two prefixes $c||0$ or $c||1$.

Efficiency follows from efficiency of Enc and Dec and the fact that at most one recursive call is made per execution of the algorithm, up to $\max_{t \in D} \text{len}(\mathsf{Enc}(t))$ levels (which is polynomial in n when $\sup(D) \subseteq \Sigma^{m(n)}$, $m(\cdot)$ and $|\Sigma|$ are polynomial so Enc's input is also polynomial-sized, and Enc is efficient). □

□

As a final note, since an efficiently computable model $\mathcal{M}$ implies efficient sampling from the respective distribution $D_{\mathcal{M}}$ (by simply sampling each token according to the respective next-token distribution), we also obtain the following corollary, which intuitively states that, given δ-optimal compression Enc for a distribution, it can be efficiently and δ-approximately sampled:

[11] This can be done efficiently by trying to decode $c||1||0^i$ for every $i \geq 0$ in order and returning the first one that decodes correctly; if c^* has prefix $c||1||0^i||1$ then there must not exist the corresponding prefix $c||1||0^i||0$, contradicting completeness by Fact B.1.

Corollary A.2. *Given a polynomial-sized alphabet family $\Sigma = \{\Sigma_n\}_{n\in\mathbb{N}}$ and polynomial maximum length $m(\cdot)$, there exists an oracle-aided sampling algorithm $\mathsf{Sample}^{(\cdot)}$ such that, for any family of distributions $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$, function $\delta(\cdot)$, and prefix-free, monotone, and efficiently encodable family of encodings $\mathsf{Enc} = \{\mathsf{Enc}_n\}_{n\in\mathbb{N}}$ where, for any $n \in \mathbb{N}$,*

$$\mathbf{E}_{x\leftarrow D_n}[\mathit{len}(\mathsf{Enc}_n(x))] \leq H(D_n) + \delta(n)$$

$\mathsf{Sample}^{(\cdot)}$ defines an efficiently computable family of randomized algorithms $\{\mathsf{Sample}_n\}_{n\in\mathbb{N}}$ such that there exists a family of distributions $\{D_n^*\}_{n\in\mathbb{N}}$ where, for any $n \in \mathbb{N}$, Sample_n samples from D_n^* and $KL(D_n \mid\mid D_n^*) \leq \delta(n)$.

A.3 The Monotonicity Requirement and Infinitely-Often One-Way Functions

A.3.1 Preliminaries

Definition A.3. *A family of functions $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, where $f_n : \{0,1\}^n \to \{0,1\}^n$, is **one-way** if $\mathcal{F}$ is efficiently computable but, for any polynomial-time inverter $\mathcal{A}$, there is a negligible function $\epsilon(\cdot)$ such that for any $n \in \mathbb{N}$:*

$$\mathit{Pr}\left[x \leftarrow \{0,1\}^n : f(\mathcal{A}(1^n, f(x))) = f(x)\right] \leq \epsilon(n)$$

If the above instead holds for only infinitely many *$n \in \mathbb{N}$ for any polynomial-time inverter $\mathcal{A}$, we say that $\mathcal{F}$ is **infinitely-often one-way**.*

That is, one-way functions are efficiently computable, but "hard to invert" in the sense that no computationally efficient algorithm, given the image of a uniformly selected input, can find any inverse of that image with probability significantly better than brute force. The existence of one-way functions is a long-standing open conjecture as it would amount to proving $\mathsf{P} \neq \mathsf{NP}$, but they are nonetheless widely used as a fundamental building block of modern cryptography. The existence of one-way functions is also well-known to imply the existence of *pseudorandom generators*:

Definition A.4. *For some output length $\ell(n) > n$, a family of efficiently computable functions $\mathcal{G} = \{g_n\}_{n\in\mathbb{N}}$ is referred to as a family of* pseudorandom generators *(PRGs) if, for any polynomial-time distinguisher D with output in $\{0,1\}$, there exists a negligible function $\epsilon(\cdot)$ such that for any $n \in \mathbb{N}$:*

$$\left|\mathit{Pr}\left[x \leftarrow U_n : D(1^n, g_n(x)) = 1\right] - \mathit{Pr}\left[y \leftarrow U_{\ell(n)} : D(1^n, y) = 1\right]\right| \leq \epsilon(n)$$

where U_k represents the uniform distribution over $\{0,1\}^k$.

That is, the output of a PRG on a uniform input should be computationally indistinguishable from uniform randomness.

A.3.2 Proof of Theorem 3.4

Proof. As the existence of one-way functions implies the existence of PRGs, let $\mathcal{G} = \{g_n\}_{n\in\mathbb{N}}$ be a PRG with output length $\ell(n) = 2n$. Take our alphabet of tokens to be $\Sigma = \{0,1\}$. Let $\mathcal{D}$ be given by $D_n = \{x \leftarrow U_n : (g_n(x), x)\}$. Let Enc be given by $\mathsf{Enc}_n(g_n(x), x) = x$.[12] Clearly, Enc satisfies prefix-freeness (since every codeword has the same length), efficient encoding and decoding, and has $\mathbf{E}_{x\leftarrow D_n}[\mathsf{len}(\mathsf{Enc}_n(x))] = H(D_n) = n$.

Now fix $\epsilon > 0$ and assume for the sake of contradiction that there exists a family of autoregressive models $\{\mathcal{M}_n\}_{n\in\mathbb{N}}$ such that $KL(D_n \mid\mid D_{\mathcal{M}_n}) < (1-\epsilon)n$ for infinitely many $n \in \mathbb{N}$. We use this to construct a distinguisher $\mathcal{A}$ (renamed from D to avoid confusion with the distributions D_n and $D_{\mathcal{M}_n}$) as follows: Take constant $c > \log(2/\epsilon)$, and, given an input y (which will either be drawn as U_{2n} or as $g_n(U_n)$), determine $D_{\mathcal{M}_n}(y, *)$ as the probability for the first $2n$ bits of an output from $\mathcal{M}_n$ to be equal to y, and let $\mathcal{A}$ output 1 when $D_{\mathcal{M}_n}(y, *) \geq 2^{c-2n}$ and 0 otherwise. We make the following claims to complete the proof by contradiction:

Claim A.5. $\mathit{Pr}\left[y \leftarrow U_{2n} : \mathcal{A}(y) = 1\right] \leq 2^{-c}$.

[12] That is, simply take the last n bits of the input. This can be efficiently decoded as well by simply computing $g_n(x)$.

Proof. $\mathbf{E}_{y \leftarrow U_{2n}}[D_{\mathcal{M}_n}(y, *)] = 2^{-2n}$, so this follows from Markov's inequality. □

Claim A.6. *For any constant $\epsilon' > 0$, $Pr[x \leftarrow U_n : \mathcal{A}(g_n(x)) = 1] > \epsilon/2 - \epsilon'$ for sufficiently large $n \in \mathbb{N}$.*

Proof. This follows from the assumption that $KL(D_n \mid\mid D_{\mathcal{M}_n}) < (1-\epsilon)n$, and thus, by data processing, that $KL((D_n)_{\leq 2n} \mid\mid (D_{\mathcal{M}_n})_{\leq 2n}) < (1-\epsilon)n$ (where $(D_n)_{\leq 2n}$ represents the distribution of the first $2n$ bits of D_n, and respectively for $D_{\mathcal{M}_n}$). Thus:

$$KL((D_n)_{\leq 2n} \mid\mid (D_{\mathcal{M}_n})_{\leq 2n}) = \mathbf{E}_{y \leftarrow D_n}[\log(1/D_{\mathcal{M}_n}(y, *))] - H((D_n)_{\leq 2n}) < (1-\epsilon)n$$

and so

$$\begin{aligned}\mathbf{E}_{x \leftarrow U_n}[\log(1/D_{\mathcal{M}_n}(g_n(x), *))] &= \mathbf{E}_{y \leftarrow D_n}[\log(1/D_{\mathcal{M}_n}(y, *))] \\ &< (1-\epsilon)n + H((D_n)_{\leq 2n}) < (2-\epsilon)n\end{aligned}$$

since $H((D_n)_{\leq 2n}) < H(D_n) = n$. Now, by construction of $\mathcal{A}$, $\mathcal{A}$ outputs 1 whenever $\log(1/D_{\mathcal{M}_n}(y, *)) < 2n - c$; by Markov's inequality and the above bound on the expectation, the probability that this is *not* true (i.e., that $\mathcal{A}$ outputs 0) when y is given by $g_n(x)$ for $x \leftarrow U_n$ can be at most $\frac{(2-\epsilon)n}{2n-c}$, so the probability that $\mathcal{A}$ outputs 1, then, must be at least

$$1 - \frac{(2-\epsilon)n}{2n-c} = \frac{\epsilon n - c}{2n-c}$$

which asymptotically approaches $\epsilon/2$ as $n \to \infty$, implying the conclusion. □

Given these, since we assumed by construction that 2^{-c} is a constant smaller than $\epsilon/2$, there is a non-negligible (constant) separation between the two probabilities for any sufficiently large n, which contradicts the assumption that $\mathcal{G}$ is a PRG and completes the proof. (We note that the proof holds identically for the case of infinitely-often OWFs, with the exception that it applies only to infinitely many $n \in \mathbb{N}$ rather than all sufficiently large n.) □

A.3.3 COROLLARY 3.5

First, we must define precisely what we mean by a model being "approximately as good" in this case—unlike our previous results, this will involve a loss of statistical distance in addition to the inherent loss of negative log-likelihood resulting from the underlying compression scheme's loss:

Definition A.7. *For alphabet Σ and distribution D over Σ^m, we say that an autoregressive model $\mathcal{M}$ is ϵ-**approximately** δ-**optimal** with respect to D if there exists a model $\mathcal{M}'$ that is δ-optimal with respect to D and, for every $i \in [m]$:*

$$\Pr_{t_{\leq i} \leftarrow D_{\leq i}}[\mathcal{M}(t_i \mid t_{<i}) / \mathcal{M}'(t_i \mid t_{<i}) \in (1-\epsilon, 1+\epsilon)] \geq 1 - \epsilon$$

(Notably, this implies that, for every $i \in [m]$, the distributions $\mathcal{M}(\cdot \mid t_{<i})$ and $\mathcal{M}'(\cdot \mid t_{<i})$ defined by sampling a random $t \leftarrow D$, taking the prefix $t_{<i}$, and sampling t_i according to the respective probabilities $\mathcal{M}(t_i \mid t_{<i})$ and $\mathcal{M}'(t_i \mid t_{<i})$ are such that $\Delta(\mathcal{M}(\cdot \mid t_{<i}), \mathcal{M}'(\cdot \mid t_{<i})) \leq \epsilon$.)

We can then leverage the following theorem from Impagliazzo & Luby (1989):

Theorem A.8 (Infinitely-often one-way functions and next-bit predictors (Impagliazzo & Luby, 1989))**.** *Let $\mathcal{D} = \{D_n\}_{n \in \mathbb{N}}$ be a family of efficiently samplable distributions where each D_n has support $\{0,1\}^{m(n)}$ for some polynomial $m(n)$. Then, for every function $\epsilon(n) \in 1/\mathsf{poly}(n)$, there exists a function f such that, if f is not an infinitely-often one-way function, then there exists an $\epsilon(n)$-approximate (0-optimal) efficiently computable autoregressive model for $\mathcal{D}$.*[13]

[13] We note that Impagliazzo & Luby (1989), as well as (in the following section) Vadhan & Zheng (2012) and Haitner et al. (2023), consider autoregressive models that can be *randomized*—that is, the probabilities $\mathcal{M}(t_i \mid t_{<i})$ are not necessarily deterministically generated. However, we note that we can still translate their results to the case of non-uniform deterministic autoregressive models (as we consider here) by considering a randomized autoregressive model with its randomness fixed to the best possible string.

Then, given a non-monotone compression scheme Enc which is δ-optimal for some distribution family $\mathcal{D}$, we can leverage part of the construction from Theorem 3.3 even though Enc is not monotone. Specifically, we showed in the proof of Theorem 3.3 that the distribution U_{Enc}, which can be thought of as sampling a uniformly random string and decoding the (unique) codeword that is a prefix of that string, is not only efficiently samplable even when Enc is non-monotone (it requires only that Enc has an efficient decoding algorithm Dec), but induces a model $\mathcal{M}'$ which is δ-optimal with respect to $\mathcal{D}$. (We note that Enc is not necessarily complete, nor can it necessarily be transformed into a complete compression scheme without loss of generality as it may not be monotone; however, in such a case we can allow U_{Enc} to output $\perp$ if the sampled string has no decodable prefix. This distribution preserves efficient samplability as we may terminate when exceeding the maximum encoding length of Enc, and it preserves δ-optimality with respect to $\mathcal{D}$ as $KL(D \mid\mid D_{\mathcal{M}'})$ remains the same under this transformation.) And, while $\mathcal{M}'$ is not necessarily efficiently computable if Enc is non-monotone, Theorem A.8 shows directly that, if infinitely-often one-way functions do not exist, then for any function $\epsilon(n) \in 1/\mathsf{poly}(n)$ there is a model $\mathcal{M}$ that is $\epsilon(n)$-approximate with respect to U_{Enc} and thus (since the induced distribution $D_{\mathcal{M}'}$ of the model $\mathcal{M}'$ is by construction equivalent to U_{Enc}) an $\epsilon(n)$-approximate δ-optimal model for $\mathcal{D}$.

In fact, Theorem A.8 shows something stronger: the existence of such a model based on an inverter to a *specific* function (as opposed to complete non-existence of infinitely-often one-way functions, which would imply the existence of an inverter for *every* function). Thus, using the function specified in the proof from Impagliazzo & Luby (1989), we can state our result in its strongest form:

Theorem A.9. *For any $\epsilon(n) \in 1/\mathsf{poly}(n)$, there exists a pair of efficient oracle-aided algorithms $(f^{(\cdot)}, \mathcal{M}^{(\cdot)})$ such that the following holds:*

Let $(\mathsf{Enc}, \mathsf{Dec})$ be a δ-optimal compression scheme (with efficient encoding and decoding) for a family $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ of efficiently samplable distributions. If there exists a polynomial-time algorithm $\mathcal{A}$ that, for all sufficiently large $n \in \mathbb{N}$, inverts f^{Dec} with probability $1/2$, then $\mathcal{M}^{\mathcal{A}}$ is an ϵ-approximate δ-optimal efficiently computable autoregressive model for $\mathcal{D}$.

In particular, $f^{\mathsf{Dec}}(x_1, \ldots, x_t) = g^{\mathsf{Dec}}(x_1), \ldots, g^{\mathsf{Dec}}(x_t)$, where $g^{\mathsf{Dec}}(r, h, i, j) = (i, j, h, \mathsf{Dec}(r)_{\le i}, h(r)_{\le j})$, h is a 2-universal hash function, and $t \in O(n/\epsilon)$.

Since the non-existence of infinitely-often one-way functions implies the existence of an inverter as specified above, this immediately implies Corollary 3.5.

A.4 Pseudoentropy and Incompressibility: Corollaries 3.6 and 3.7

Informally, a distribution D has *pseudoentropy* (Håstad et al., 1999) at least h if it is computationally indistinguishable from some distribution D' with Shannon entropy $H(D') \ge h$. Restricting to the case of binary distributions ($\Sigma = \{0, 1\}$), we will formally consider a related notion of *next-bit* pseudoentropy, or pseudoentropy with respect to next-bit prediction:

Definition A.10 ((Vadhan & Zheng, 2012))**.** *Given jointly distributed families of random variables $\mathcal{X} = \{X_n\}_{n\in\mathbb{N}}$ and $\mathcal{B} = \{B_n\}_{n\in\mathbb{N}}$, where $\mathcal{B}$ takes values in $\{0, 1\}$, we say that $\mathcal{B}$ has (non-uniform) **conditional pseudoentropy** at least $h(\cdot)$ given X if, for any polynomial $p(\cdot)$, there exists a $\{0, 1\}$-valued family $\mathcal{C} = \{C_n\}_{n\in\mathbb{N}}$ of random variables jointly distributed with $(\mathcal{X}, \mathcal{B})$ such that:*

1. *For all sufficiently large $n \in \mathbb{N}$, $H(C_n|X_n) \ge h(n) - 1/p(n)$*
2. *The families $(\mathcal{X}, \mathcal{B})$ and $(\mathcal{X}, \mathcal{C})$ are computationally indistinguishable.*[14]

*Given a polynomial $m(\cdot)$ and family of random variables $\mathcal{B} = \{B_n\}_{n\in\mathbb{N}}$ where B_n takes values in $\{0, 1\}^{m(n)}$, we say that $\mathcal{B}$ has (non-uniform) **next-bit pseudoentropy** at least $h(\cdot)$ if, letting family $\mathcal{I} = \{I_n\}_{n\in\mathbb{N}}$ be such that I_n is uniformly distributed over $[m(n)]$, then the conditional pseudoentropy of $\{(B_n)_{I_n}\}_{n\in\mathbb{N}}$ given $\{(B_n)_{<I_n}\}_{n\in\mathbb{N}}$ is at least $h(n)/m(n)$.*

Haitner et al. (2023) show a partial characterization of the relation between next-bit pseudoentropy and compressibility of a family of distributions (a notion originally proposed by Yao (1982)). Specif-

[14] That is, for any efficient distinguisher D, there exists negligible $\epsilon(\cdot)$ such that, for all sufficiently large $n \in \mathbb{N}$, $\Pr[(x, b) \leftarrow (X_n, B_n) : D(1^n, x, b) = 1]$ and $\Pr[(x, c) \leftarrow (X_n, C_n) : D(1^n, x, c) = 1]$ differ by no more than $\epsilon(n)$.

ically, they show that incompressibility implies pseudoentropy (Haitner et al. (2023), Lemma 36): if a family of distributions $\mathcal{B} = \{B_n\}_{n\in\mathbb{N}}$ over $\{0,1\}^{m(n)}$ cannot be compressed with some length $k(n)$, then it must have next-bit pseudoentropy at least $k(n) - 2$. However, they also show that the converse is *not* true; in fact, there exists a separation (Haitner et al. (2023), Prop. 39, Lemma 42) showing that, assuming the existence of one-way functions, next-bit pseudoentropy cannot imply incompressibility unless it is strengthened to a notion of "strong" next-bit pseudoentropy which requires next-bit pseudoentropy of $\{\mathsf{Enc}(B_n)\}_{n\in\mathbb{N}}$ for any efficient encoding $\mathsf{Enc}(\cdot)$.

Our results, in fact, complete the characterization of *standard* next-bit pseudoentropy by showing that, up to a constant, it is equivalent to *monotone* incompressibility. This is illustrated by the following corollary of Corollary 3.9 of Vadhan & Zheng (2012):

Corollary A.11 ((Vadhan & Zheng, 2012; Haitner et al., 2023))**.** *Given polynomial maximum length $m(\cdot)$ and family of distributions $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ where D_n is a distribution over $\{0,1\}^{m(n)}$, for some function $\delta(\cdot)$, $\mathcal{D}$ has next-bit pseudoentropy at least $H(D_n)+\delta(n)$ if and only if there does not exist an efficiently computable autoregressive model family $\mathcal{M} = \{M_n\}_{n\in\mathbb{N}}$ where, for each $n \in \mathbb{N}$, M_n is $\delta(n)$-optimal with respect to D_n.*

The original corollary of Vadhan & Zheng (2012) shows an equivalence between $\mathcal{B}$ given $\mathcal{X}$ (as given in Definition A.10) having conditional pseudoentropy $H(B_n|X_n) + \delta(n)$ and $\mathcal{B}$ being $\delta(n)$-*KL-hard* given $\mathcal{X}$—that is, that there exists no efficient predictor $P(x)$ that attempts to predict B given X and has $KL((X,B) \mid\mid (X,P(X))) \leq \delta(n)$. Haitner et al. (2023) observed that, given a family $\mathcal{D}$ of distributions, one can apply this to a uniformly random bit in a distribution D_n (as in the definition of next-bit entropy) and use the additivity of entropy and KL-divergence chain rule to immediately imply that $\mathcal{D}$ having "non-trivial" next-bit entropy $H(D_n) + \delta(n)$ implies a next-bit predictor $\mathcal{M}$ for *any* bit of the distribution D_n (i.e., an autoregressive model, using our terminology) for which $KL(D_n \mid\mid D_{\mathcal{M}_n}) \leq \delta(n)$, or, in other words, $\mathcal{M}$ is $\delta(n)$-optimal with respect to $\mathcal{D}$.

Returning to the equivalence between next-bit pseudoentropy and monotone compression and applying Corollary A.11, we can now observe that this equivalence is intuitively the same as equivalence between next-bit *prediction* and monotone compression, which, in fact, is precisely what we have shown in Section A.2. Corollary 3.6 is immediate from (the contrapositive of) Corollary 3.2 and Corollary A.11, though we note that it is also already proven implicitly in Lemma 36 of Haitner et al. (2023) because their construction of a compression scheme is monotone. Specifically, if $\mathcal{D}$ has insufficient pseudoentropy, then by Corollary A.11 it must have a $(k(n) - 2 - H(D_n))$-optimal efficiently computable autoregressive model family $\mathcal{M}$, and Corollary 3.2 demonstrates that arithmetic encoding gives efficient and length-$k(n)$ compression. Corollary 3.7 is also immediate from our Theorem 3.3 and Corollary A.11: since Theorem 3.3 shows that $\mathcal{D}$ having length-$k(n)$ compression implies a $(k(n) - H(D_n))$-optimal autoregressive model family $\mathcal{M}$, the other direction of Corollary A.11 implies that $\mathcal{D}$ cannot have next-bit pseudoentropy at least $k(n)$ in this case.

B Properties of Complete Encoding

We discuss the notion of completeness of an encoding—that is, the property that Kraft's inequality for prefix-free codes (McMillan, 1956)

$$\sum_{x\in\mathrm{sup}(D)} |\Sigma'|^{-\mathsf{len}(\mathsf{Enc}(x))} \leq 1$$

is tight for a particular encoding Enc. One desirable property of complete binary codes is as follows:

Fact B.1. *Given a prefix-free binary code* Enc*, the following are equivalent:*

1. Enc *is complete.*
2. *Given the set C of valid codewords of* Enc*, if there exists some $c \in C$ with prefix y, then either (a) $y = c$ or (b) there exist $c_0, c_1 \in C$ such that c_0 has prefix $y||0$ and c_1 has prefix $y||1$.*

Proof. (1) → (2). Assume towards contradiction that there is $c_0 \in C$ with prefix $y||0$ but no $c_1 \in C$ with prefix $y||1$ (the following will hold symmetrically for the other case). Construct a new code

Enc' with any codeword having prefix $y||0$ replaced by the same codeword with prefix y instead. Since by our assumption any codeword with prefix y also has prefix $y||0$, Enc' is also prefix-free, thus it satisfies Kraft's inequality:

$$\sum_{x \in \sup(D)} 2^{-\text{len}(\mathsf{Enc}'(x))} \leq 1$$

However, because in converting Enc to Enc' we strictly decreased the length of at least one codeword (c_0) and did not increase the length of any codewords, this means that

$$\sum_{x \in \sup(D)} 2^{-\text{len}(\mathsf{Enc}(x))} < \sum_{x \in \sup(D)} 2^{-\text{len}(\mathsf{Enc}'(x))} \leq 1$$

contradicting our assumption that Enc is complete.

(2) → (1). We show inductively that given any prefix y of some codeword $c \in C$,

$$\sum_{x \in \sup(D)|_y} 2^{-\text{len}(\mathsf{Enc}'(x))} = 2^{-\text{len}(y)}$$

where $\sup(D)|_y$ is the elements of $\sup(D)$ whose encodings have prefix y. There are two cases given by property (2) to consider:

(a). If $y = c$ is a complete codeword (and thus there are no other codewords with prefix y, then clearly $\sup(D)|_y$ consists of only $\mathsf{Dec}(y)$ and thus the equality holds trivially.

(b). If there exist $c_0, c_1 \in C$ such that c_0 has prefix $y||0$ and c_1 has prefix $y||1$, then this defines disjoint non-empty sets $\sup(D)|_{y||0}$ and $\sup(D)|_{y||1}$ where (by prefix-freeness) $\sup(D)|_y = \sup(D)|_{y||0} \cup \sup(D)|_{y||1}$, and we apply induction to state:

$$\begin{aligned}\sum_{x \in \sup(D)|_y} 2^{-\text{len}(\mathsf{Enc}'(x))} &= \sum_{x \in \sup(D)|_{y||0}} 2^{-\text{len}(\mathsf{Enc}'(x))} + \sum_{x \in \sup(D)|_{y||1}} 2^{-\text{len}(\mathsf{Enc}'(x))} \\ &= 2^{-(\text{len}(y)+1)} + 2^{-(\text{len}(y)+1)} = 2^{-\text{len}(y)}\end{aligned}$$

The conclusion follows when we take y to be the empty string, i.e., $\sup(D)|_y = \sup(D)$ and thus

$$\sum_{x \in \sup(D)} 2^{-\text{len}(\mathsf{Enc}'(x))} = \sum_{x \in \sup(D)|_y} 2^{-\text{len}(\mathsf{Enc}'(x))} = 2^{-\text{len}(y)} = 2^0 = 1$$

□

Intuitively, this means that the canonical binary tree representation of any complete binary code is a complete tree—every node has either 0 or 2 descendants. While completeness may not seem like an intuitively achievable property—for instance, it is not satisfied by arithmetic encoding—we can actually use the intuition from the proof above to show that any incomplete code can be converted into a complete code while preserving other desiderata such as prefix-freeness and monotonicity and also either preserving or reducing the length of every codeword:

Claim B.2. *Given any monotone and prefix-free binary encoding* Enc *defined on a distribution* D*, there exists monotone and prefix-free binary encoding* Enc' *defined on* D *such that* Enc' *is complete and, for any* $x \in D$*,* $\textit{len}(\mathsf{Enc}'(x)) \leq \textit{len}(\mathsf{Enc}(x))$*. Additionally, if* Enc *can be computed efficiently, so can* Enc' *and (by Fact 2.3) the respective inverse* Dec'*.*

Proof. Define $\mathsf{Enc}'(x)$ as follows:

1. Let $y = \mathsf{Enc}(x)$ and $\ell = \text{len}(y)$. Initialize S as the empty set.
2. For $i \in [\ell]$:
   (a) Determine whether there exists $x' \in D$ such that $y'_{\leq i} = y_{<i}||(1 - y_i)$, where $y' = \mathsf{Enc}(x')$. (Since we assume Enc to be monotone, this can be done in $O(\log(|D|))$ computations of Enc by binary search.)

(b) If not, add i to S.

3. Return $y_{[\ell]\setminus S}$, y with the bits indexed by S removed.

To show completeness, consider any $x \in D$ and any bit y_i of $y = \mathsf{Enc}(x)$ that remains in $\mathsf{Enc}'(x)$ (i.e., $i \notin S$). Then there is $x' \in D$ and the corresponding encoding $y' = \mathsf{Enc}(x')$ for which $y'_{\leq i} = y_{<i}||(1-y_i)$. y_i will become bit z_j of $z = \mathsf{Enc}'(x)$ for some $j \leq i$, since it is possible that bits before y_i will be removed, but importantly, y'_i must also then become bit z'_j of $z' = \mathsf{Enc}'(x')$, since $y'_{<i} = y_{<i}$ and thus any index before the ith will be added to S in the computation of $\mathsf{Enc}'(x')$ if and only if the same index was added to S in the computation of $\mathsf{Enc}'(x)$. Therefore $z'_{<j} = z_{<j}||(1-z_i)$. Thus, we have demonstrated the property required to demonstrate completeness by Fact B.1.

For monotonicity, let $x_1, x_2 \in D$ be such that $x_1 \prec x_2$. By monotonicity of Enc, this implies $y_1 < y_2$, where $y_1 = \mathsf{Enc}(x_1)$ and $y_2 = \mathsf{Enc}(x_2)$. Consider the first bit where y_1 and y_2 differ; this must be some index i where $(y_1)_i = 0$ and $(y_2)_i = 1$ (as y_1 ending before the ith bit would violate prefix-freeness of Enc). As above, if $z_1 = \mathsf{Enc}'(x_1)$ and $z_2 = \mathsf{Enc}'(x_2)$, then index i of y_1 corresponds to some index $j \leq i$ in z_1, but, since by assumption $(y_1)_{<i} = (y_2)_{<i}$, it must be true that $(z_1)_{<j} = (z_2)_{<j}$ and $(y_2)_i$ likewise corresponds to $(z_2)_j$. This means that $z_1 = (z_1)_{<j}||0$ and $z_2 = (z_1)_{<j}||1$, so $z_1 < z_2$ as desired.

Lastly, for prefix-freeness, assume for contradiction that there exist $x, x' \in D$ such that (letting $z = \mathsf{Enc}'(x)$ and $z' = \mathsf{Enc}'(x')$) $z' = z_{\leq j}$ for some $j < \text{len}(z')$—that is, z' is a strict prefix of z. Let $y = \mathsf{Enc}(x)$ and $y' = \mathsf{Enc}(x')$ be the corresponding codewords under Enc; we claim that there is an index $i \geq j$ for which $y' = y_{\leq i}$. If not, then there is a first index k where $y_k \neq y'_k$, but this implies $k \notin S$ during the computation of Enc' for both x and x', so both respective bits would be included at the same index in z and z' (since we assumed $y_{<k} = y'_{<k}$), contradicting z' being a strict prefix of z. Thus y' is a prefix of y; it must also be a strict prefix because otherwise $y' = y$ and thus $z' = z$. But this contradicts prefix-freeness of Enc as desired. □