





Machine Learning Predictors for Min-Entropy Estimation

Javier Blanco-Romero ^{1,*}, Vicente Lorenzo ^{1,2}, Florina Almenares Mendoza ¹ and Daniel Díaz-Sánchez ¹

¹ Department of Telematic Engineering, Universidad Carlos III de Madrid, Leganés, 28911 Madrid, Spain; vilorenz@pa.uc3m.es (V.L.); florina@it.uc3m.es (F.A.M.); dds@it.uc3m.es (D.D.-S.)

² Department of Applied Mathematics for ICT, Universidad Politécnica de Madrid, 28031 Madrid, Spain

* Correspondence: frblanco@pa.uc3m.es

Abstract: This study investigates the application of machine learning predictors for the estimation of min-entropy in random number generators (RNGs), a key component in cryptographic applications where accurate entropy assessment is essential for cybersecurity. Our research indicates that these predictors, and indeed any predictor that leverages sequence correlations, primarily estimate average min-entropy, a metric not extensively studied in this context. We explore the relationship between average min-entropy and the traditional min-entropy, focusing on their dependence on the number of target bits being predicted. Using data from generalized binary autoregressive models, a subset of Markov processes, we demonstrate that machine learning models (including a hybrid of convolutional and recurrent long short-term memory layers and the transformer-based GPT-2 model) outperform traditional NIST SP 800-90B predictors in certain scenarios. Our findings underscore the importance of considering the number of target bits in min-entropy assessment for RNGs and highlight the potential of machine learning approaches in enhancing entropy estimation techniques for improved cryptographic security.

Keywords: min-entropy estimation; machine learning predictors; random number generators; autoregressive processes; generalized binary autoregressive models



Academic Editor: Andrea Murari

Received: 3 January 2025

Revised: 23 January 2025

Accepted: 24 January 2025

Published: 2 February 2025

Citation: Blanco-Romero, J.; Lorenzo, V.; Almenares Mendoza, F.; Díaz-Sánchez, D. Machine Learning Predictors for Min-Entropy Estimation. *Entropy* **2025**, *27*, 156. <https://doi.org/10.3390/e27020156>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The security of cryptographic systems often relies on the generation of random values. Although there is a broad spectrum of algorithms and devices used to generate these random values, they are all generically denoted by random number generators (RNGs). Given the important role that RNGs play in the context of cybersecurity, it becomes evident that rigorous criteria are necessary to evaluate the reliability and performance of an RNG.

Multiple approaches are commonly employed to assess the quality of the output of an RNG (cf. [1–6], etc.). In this paper, the emphasis will be put on the following:

- Entropy tests, as those found in NIST Special Publication 800-90B [7], which estimate the entropy of a noise source based on appropriate samples (cf. [8–10], etc.).
- Machine Learning models trained with the output of an RNG aiming to guess the bit or set of bits that follow a given sequence, which can give an insight into how predictable the output of the RNG is (cf. [11–16], etc.).

The relationship between entropy measures and the predictability of sequences is a key concept in information theory. Nevertheless, the link between these two concepts is far from being completely understood, especially if one takes into account the heterogeneity of entropy definitions that can be found in the literature and how much the predictability of the output of an entropy source relies on the predictor being considered. Based on the evidence provided by [17] that the entropy estimators considered by NIST Special

Publication 800-90B [7] tend to underestimate min-entropy, in this paper, an attempt is made to reinforce the argument [18] that predictors are better suited to estimate average min-entropy [19] than min-entropy. In the first stage, the theoretical framework required to support our thesis is developed. In the second stage, experimental validation of the theoretical analysis is conducted.

Whereas [17] concentrates on ensemble, categorical data, and numerical predictors, the focus of this paper will be on machine learning predictors. In particular, a hybrid model that integrates convolutional and recurrent long short-term memory (LSTM) layers, and the transformer-based GPT-2 model will be considered. As in [18], we generate sets of data for which a theoretical entropy can be calculated so that the machine learning entropy estimation can be compared to the theoretical value. Nevertheless, while the data generated in [18] come from an oscillator-based model and Markov processes of order at most 2, our data come from generalized binary autoregressive models [20], a subclass of Markov chains that allows us to easily parameterize correlations and anticorrelations at the bit level and compute min-entropies.

Our research also investigates the influence of the number of target bits on the estimation of min-entropy. We demonstrate that the relationship between average min-entropy and min-entropy is significantly affected by the number of target bits predicted. This finding highlights the importance of considering the target bit count when assessing the min-entropy of RNGs using machine learning predictors.

The remainder of this paper is structured as follows. Section 2 presents a literature review, discussing the current state-of-the-art in the application of predictors for min-entropy estimation. In Section 3, we establish the theoretical framework, where we study the concept of average min-entropy and its relationship with min-entropy, deriving a series of results for order- p Markov chains and gbAR(p). Section 4 outlines our experimental methodology that aims to validate the theoretical findings. Section 5 presents the results of our experiments, followed by Section 6, which offers a discussion of these results and their implications. Finally, Section 7 concludes the paper with a summary of our findings and suggestions for future research.

Notation and Conventions

The following notation and conventions will be considered throughout this paper:

- Random variables: Uppercase letters X_1, X_2, A, \dots represent random variables, while their corresponding realizations are represented by lowercase letters x_1, x_2, a, \dots , and by abusing the notation, $P(A = a)$ will be denoted by $P(a)$. Furthermore, by $X \sim B(n, p)$, we mean that X is a random variable that follows a binomial distribution with number of trials n and a success probability p , and by $(X_1, \dots, X_k) \sim \text{Mult}(n; p_1, \dots, p_k)$, we mean that (X_1, \dots, X_k) is a multivariate random variable that follows a multinomial distribution with number of trials n and probability vector (p_1, \dots, p_k) .
- Expected values: The notation $\langle \cdot \rangle$ is used to indicate expected values. Given discrete random variables X_{t-p}, \dots, X_{t+n} where $t, p, n \in \mathbb{Z}$, and p, n are nonnegative, we will be particularly interested in the following type of expression:

$$\left\langle \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right\rangle_{x_{t-1}, \dots, x_{t-p}} = \sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}). \tag{1}$$

- Logarithms: All logarithmic functions are considered to be base 2 and are denoted by \log .

2. State of the Art

2.1. Entropies

The relationship between entropy and predictability of a sequence was first investigated by Shannon [21], who noticed that the problem of prediction is fundamentally connected to the concept of entropy. Min-entropy, denoted as $H_\infty(X)$, represents the negative logarithm of the probability of a correct guess on the random variable X under an optimal strategy [22]. Mathematically, the probability of guessing the most likely output of an entropy source can be expressed as follows:

$$2^{-H_\infty(X)} = \max_{x \in X} P_X(x). \quad (2)$$

where $P_X(x)$ is the probability distribution of the random variable X .

In cryptography, min-entropy is an important measure, as it provides a conservative estimate of the difficulty of guessing or predicting the most likely output of the entropy source, as emphasized in the NIST recommendation [7].

Moreover, entropy estimation is complex when the output distribution is unknown, and typical assumptions like output being independent and identically distributed (i.i.d.) do not apply. Good entropy estimation requires understanding of the underlying nondeterministic process of the entropy source, and statistical tests, as referenced, can only act as a sanity check on such an estimate [17].

In this context, the concept of predictors has been introduced. As described by Kelsey et al., a predictor contains a dynamic model that operates through a four-step process: (1) Assume a probability model for the source; (2) Estimate the model's parameters from the input sequence on-the-fly; (3) Use these parameters to attempt to predict the still-unseen values in the input sequence; and (4) Estimate the min-entropy of the source from the performance of these predictions [17]. Unlike traditional machine learning methods, this approach is parametric and relies on a model of the underlying probability distribution. Another difference from traditional supervised learning methods, which separate training and testing sets, is that predictors remain in the training phase indefinitely, allowing for continuous adaptation and improvement in prediction accuracy. The predictors are characterized by two primary performance metrics. The first, global predictability, gauges the long-term accuracy of the predictions. Specifically, a predictor's global accuracy p_{acc} represents the probability that it will correctly predict a given sample from a noise source over an extended sequence, effectively measuring the percentage of correct predictions. The second, local predictability, emphasizes the length of the longest streak of correct predictions, becoming important when the source produces highly predictable outputs in short spurts. The final entropy estimate for a predictor is determined by the lesser value between the global and local entropy estimates, represented by $\hat{H} = \min(\hat{H}_{global}, \hat{H}_{local})$.

Hence, predictors play a significant role in setting bounds on an attacker's performance, linking predictability to min-entropy. For a description of the evolution of the introduction of predictors on the NIST SP 800-90B, see [13].

Zhu et al. examined the issue of underestimation in non-IID data pertaining to the NIST collision and compression test, proposing an enhanced method to address the underestimation of min-entropy [18]. They introduced a novel formula specifically aimed at the high-order Markov process, founded on the principles of conditional probability. Furthermore, they highlighted that the correct prediction probability within a predictor can also be understood as a form of conditional probability.

Zhu's min-entropy formula for the Markov process can be related to the concept of average min-entropy as defined by Dodis [19]. Average min-entropy considers the

predictability of a random variable given another possibly correlated random variable and can be expressed as

$$\tilde{H}_\infty(A | B) = -\log\left(\langle \max_a P(a | b) \rangle_b\right) = -\log\left(\langle 2^{-H_\infty(A|B=b)} \rangle_b\right), \quad (3)$$

where

$$H_\infty(A | B = b) = P(b) \cdot \max_a P(a | b).$$

Several works have considered the problem of designing machine learning predictors for evaluating RNGs. Here, we examine some of the most relevant contributions in relation to min-entropy estimation.

Truong et al. [11] introduced the use of a recurrent convolutional neural network (RCNN) to analyze quantum random number generators (QRNGs). This RCNN model was used to evaluate different stages of an optical continuous variable QRNG. The study focused on detecting inherent correlations, particularly under the influence of deterministic classical noise. Their methodology included a comprehensive analysis, from examining the robustness of QRNGs against machine learning attacks to benchmarking with a congruential pseudorandom number generator (CRNG). Their model's prediction accuracy was compared with the guessing probability of the data distribution, effectively entailing a comparison with the min-entropy.

Yang et al. and Lv et al. explored neural-network-based min-entropy estimation for random number generators [12,13]. Their approach involved training predictive models on simulated data, where the correct entropy was ascertainable due to the known output distributions. Additionally, their study included a performance analysis and comparison of their results with the NIST SP 800-90B's predictors, providing a detailed examination of the efficacy and accuracy of their neural-network-based approach in entropy estimation.

Li et al. [14] proposed a predictive analysis based on deep learning to assess the security of a nondeterministic random number generator (NRNG) using white chaos. They employed a temporal pattern attention (TPA)-based deep learning model to analyze data from both the chaotic external-cavity semiconductor laser (ECL) stage and the final output of the NRNG. The model effectively detected correlations in the ECL stage but not in the post-processed output, suggesting the NRNG's resistance to predictive modeling. Prior to this, the model's predictive power was validated on a linear congruential algorithm-based RNG. The study also compared the model's prediction accuracy with the baseline probability, aligning with Truong et al.'s approach of using the guessing probability as a comparative metric for min-entropy estimation.

Finally, Haohao Li et al. [16] proposed a method for min-entropy evaluation using a pruned and quantized deep neural network. They developed a temporal pattern attention-based long short-term memory (TPA-LSTM) model, which they then optimized through pruning and quantization. This optimized model was retrained and tested on various simulated datasets with known min-entropy values. Their results demonstrated greater accuracy in min-entropy estimation compared to the NIST SP 800-90B predictors. This study also investigated why NIST predictors often underestimate min-entropy, attributing it to the sensitivity of local predictability probability to parameter variations. This work parallels Yang et al. and Lv et al.'s in comparing neural-network-based min-entropy estimations with NIST SP 800-90B's predictors.

2.2. Autoregressive Inference and Multi-Token Prediction Strategies

In autoregressive inference, various sampling strategies can be employed to generate sequences, such as greedy decoding [23], beam search [24,25], top-k sampling [23,26], or top-p/nucleus sampling [27], with or without temperature-based sampling techniques [28].

However, these techniques may not always yield the globally optimal sequence as they rely on local decisions at each step.

When performing autoregressive inference, our objective is to find sequences of n bits that, given the previous p bits, maximize the conditional probability:

$$\max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}). \quad (4)$$

To illustrate the potential limitations of autoregressive inference strategies, consider greedy decoding as an example. Greedy decoding selects the most probable bit at each step, conditioned on the previously generated bits. This can be expressed as follows:

$$\prod_{k=t}^{t+n} \max_{x_k} P(x_k \mid x_{k-1}, \dots, x_{k-p}) \Big| x_{k-i} = \arg \max_{x_{k-i}} P(x_{k-i} \mid x_{k-i-1}, \dots, x_{k-i-p}), \quad \forall i \in [1, k \leq p] \quad (5)$$

However, the product of the maximum conditional probabilities at each step does not necessarily equal the global maximum probability over the entire sequence. In other words, the greedy decoding approach may lead to suboptimal sequences, as it does not consider the joint probability of the complete sequence.

While other search methods, such as beam search, top-k sampling, or top-p/nucleus sampling, can perform better than greedy decoding, they still face the same fundamental challenge. Ultimately, the effectiveness of these methods in approximating the global maximum depends on the data and the search space. As the sequence length n increases, the search space grows exponentially, making it increasingly difficult to find the optimal global sequence efficiently.

Recently, incorporating future information into language generation tasks has gained attention. Li et al. (2017) [29] proposed an actor-critic model that integrates a value function to estimate future success, combining MLE-based learning with an RL-based value function during decoding. Oord et al. (2018) [30] aimed to preserve mutual information between context and future tokens by modeling a density ratio rather than directly predicting future tokens. Serdyuk et al. (2018) [31] addressed the challenge of long-term dependency learning in RNNs by running forward and backward RNNs in parallel to better capture future information. Lawrence et al. (2019) [32] trained an encoder by concatenating source and target sequences and using placeholder tokens in the target sequence, which are replaced during inference to generate the final output. These advancements illustrate the growing interest in and potential for optimizing future token predictions in natural language processing tasks.

Qi et al. (2020) [33] introduce ProphetNet, a sequence-to-sequence pre-training model that employs a novel self-supervised objective called future n-gram prediction and an n-stream self-attention mechanism. Unlike traditional sequence-to-sequence models that optimize one-step-ahead prediction, ProphetNet predicts the next n tokens simultaneously based on previous context tokens at each time step. This approach explicitly encourages the model to plan for future tokens and prevents overfitting on strong local correlations. The authors pre-train ProphetNet using base and large-scale datasets and demonstrate state-of-the-art results on abstractive summarization and question generation tasks.

Recent works have explored the use of multi-token prediction to improve the efficiency and performance of large language models. Gloeckle et al. propose a memory-efficient implementation and demonstrate the effectiveness of this approach on various tasks, demonstrating strong performance on summarization, speeding up inference by a factor of 3×, and promoting the learning of longer-term patterns [34]. This method has been shown to improve sample efficiency, downstream capabilities, and inference speed, especially

for larger model sizes and generative benchmarks like coding. Similarly, Stern et al. [35] and Cai et al. [36] introduce methods that augment LLM inference by adding additional decoding heads to predict multiple subsequent tokens in parallel. Cai et al. refine the concept introduced by Stern et al. and propose MEDUSA, which uses a tree-based attention mechanism to construct and verify multiple candidate continuations simultaneously. While all three approaches leverage multi-token prediction, Gloeckle et al. focus on the effects of such a loss during pre-training, whereas Stern et al. and Cai et al. propose model finetunings for faster inference without studying the pre-training effects [34].

3. Theoretical Framework

In this section, we establish the theoretical framework for our study. We focus on investigating the concept of average min-entropy and its relationship with min-entropy, particularly within the context of gbAR(p) models. The proofs and auxiliary results supporting our findings can be found in the Appendix A.

3.1. Entropies

The different entropies that will be considered throughout this paper are gathered in the following:

Definition 1. Let $\{X_t\}_{t \in \mathbb{Z}}$ be a stochastic process with discrete state space and let $(X_{t-p}, \dots, X_{t-1}, X_t, \dots, X_{t+n})$ be a subset of $\{X_t\}_{t \in \mathbb{Z}}$ where $t, p, n \in \mathbb{Z}$, and p, n are nonnegative. Then,

(a) The min-entropy of (X_t, \dots, X_{t+n}) is

$$H_\infty(X_t, \dots, X_{t+n}) = -\log \left[\max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n}) \right]. \tag{6}$$

(b) The min-entropy per bit of (X_t, \dots, X_{t+n}) is [19]

$$h_\infty(X_t, \dots, X_{t+n}) = \frac{1}{n+1} H_\infty(X_t, \dots, X_{t+n}). \tag{7}$$

(c) The min-entropy per bit of $\{X_t\}_{t \in \mathbb{Z}}$ is

$$h_\infty(\{X_t\}_{t \in \mathbb{Z}}) = -\lim_{k \rightarrow \infty} \frac{1}{k+1} \log \left[\max_{x_t, |t| \leq \lfloor k/2 \rfloor} P(\{x_t\}_{|t| \leq \lfloor k/2 \rfloor}) \right], \tag{8}$$

$k \in \mathbb{Z}.$

(d) The worst-case min-entropy of (X_t, \dots, X_{t+n}) is

$$H_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = -\log \left[\max_{x_{t-p}, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right]. \tag{9}$$

(e) The worst-case min-entropy per bit of (X_t, \dots, X_{t+n}) is

$$h_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = \frac{1}{n+1} H_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}). \tag{10}$$

(f) The average min-entropy of (X_t, \dots, X_{t+n}) is [19]

$$\begin{aligned} & \tilde{H}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = \\ & -\log \left(\left\langle \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right\rangle_{x_{t-1}, \dots, x_{t-p}} \right) = \\ & -\log \left[\sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right]. \end{aligned} \tag{11}$$

(g) The average min-entropy per bit of (X_t, \dots, X_{t+n}) is

$$\begin{aligned} & \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = \\ & \frac{1}{n+1} \tilde{H}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}). \end{aligned} \tag{12}$$

Remark 1. When determining the min-entropy of an entire binary stochastic process $\{X_t\}_{t \in \mathbb{Z}}$, the direct evaluation

$$H_\infty(\{X_t\}_{t \in \mathbb{Z}}) = -\log \left[\max_{x_t, t \in \mathbb{Z}} P(\{x_t\}_{t \in \mathbb{Z}}) \right] \tag{13}$$

can lead to undefined behavior. Indeed, if we write this as the limit,

$$H_\infty(\{X_t\}_{t \in \mathbb{Z}}) = \lim_{k \rightarrow \infty} H_\infty(\{X_t\}_{|t| \leq \lfloor k/2 \rfloor}), \quad k \text{ even}$$

the maximum probability decay with k is bounded below by $\frac{1}{2^{k+1}}$ corresponding to the uniform noise, so in that case, the limit

$$H_\infty(\{X_t\}_{t \in \mathbb{Z}}) = -\lim_{k \rightarrow \infty} \log \left[\frac{1}{2^{k+1}} \right] = 1 + \lim_{k \rightarrow \infty} k$$

diverges, growing as $\sim k$ with the number of elements k . Then, the limit

$$h_\infty(\{X_t\}_{t \in \mathbb{Z}}) = -\lim_{k \rightarrow \infty} \frac{1}{k+1} \log \left[\max_{x_t, |t| \leq \lfloor k/2 \rfloor} P(\{x_t\}_{|t| \leq \lfloor k/2 \rfloor}) \right], \tag{14}$$

$k \in \mathbb{Z}$

is bounded by 1 for all $\{X_t\}_{t \in \mathbb{Z}}$. For this reason, we are going to refer to this as the min-entropy per bit of the stochastic process $\{X_t\}_{t \in \mathbb{Z}}$.

3.2. Order- p Markov Chains

Let us begin by defining order- p Markov chains:

Definition 2 (cf. [37,38]). An order- p Markov Chain is a stochastic process $\{X_t\}_{t \in \mathbb{Z}}$ with discrete state-space S such that

$$P(x_{t_0} \mid \{x_t\}_{t < t_0}) = P(x_{t_0} \mid x_{t_0-1}, \dots, x_{t_0-p}) \tag{15}$$

for every $t_0 \in \mathbb{Z}$ and every $x_t \in S$ such that $t \leq t_0$.

The aim of this subsection is to define special types of Markov chains and to prove some formulas that are applicable to them regarding the entropies of Definition 1. Although the experiments performed in this paper mostly involve generalized binary autoregressive models (see Definition 6 below), other types of Markov chains that have connections with generalized binary autoregressive models are also explored (see Figure 1) because they

share certain properties with them, and entropy formulas that are interesting on their own can be derived with relatively little additional effort for these processes.

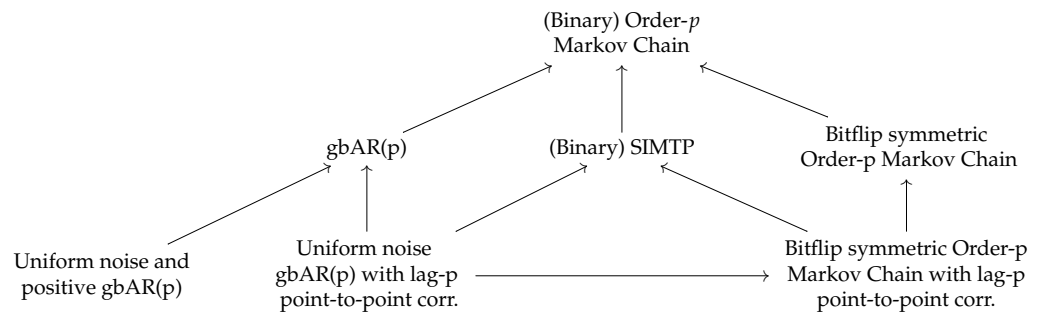


Figure 1. Hierarchy of the main models considered in this paper. An arrow from model M_1 to model M_2 means that every model of type M_1 is of type M_2 .

Definition 3. Let $\{X_t\}_{t \in \mathbb{Z}}$ be an order- p Markov chain with state-space S . Then,

- (i) $\{X_t\}_{t \in \mathbb{Z}}$ is said to be binary if $S = \{0, 1\}$.
- (ii) $\{X_t\}_{t \in \mathbb{Z}}$ is said to be stationary if

$$P(X_{t_1} = x_1, \dots, X_{t_n} = x_n) = P(X_{t_1+\tau} = x_1, \dots, X_{t_n+\tau} = x_n)$$

for every $\tau, t_1, \dots, t_n \in \mathbb{Z}$, every $x_1, \dots, x_n \in S$, and every positive integer n .

- (iii) $\{X_t\}_{t \in \mathbb{Z}}$ is said to have lag- p point-to-point correlations if

$$P(x_t \mid x_{t-1}, \dots, x_{t-p}) = P(x_t \mid x_{t-p}) \text{ for every } t \in \mathbb{Z}.$$

- (iv) $\{X_t\}_{t \in \mathbb{Z}}$ is said to be irreducible if it is stationary, S is finite, and for every $x, x_1, \dots, x_p \in S$, there exists a nonnegative integer k such that

$$P(X_{t+k} = x \mid X_{t-1} = x_1, \dots, X_{t-p} = x_p) > 0.$$

- (v) $\{X_t\}_{t \in \mathbb{Z}}$ is said to be aperiodic if it is stationary, S is finite, and for every $x \in S$,

$$\gcd\{n \geq 1 : P(X_{t+n} = x \mid X_t = x) > 0\} = 1.$$

Remark 2. Note that if $\{X_t\}_{t \in \mathbb{Z}}$ is a stationary order- p Markov chain, then

$$h_\infty(\{X_t\}_{t \in \mathbb{Z}}) = \lim_{n \rightarrow \infty} \frac{1}{n+1} H(X_t, \dots, X_{t+n}). \tag{16}$$

3.2.1. Some Min-Entropy Inequalities for Order- p Markov Chains

Here, we establish several inequalities involving min-entropy, average min-entropy, and worst-case min-entropy.

We start by noting that for a fixed n , the following inequality between min-entropy, average min-entropy, and worst-case min-entropy holds.

Lemma 1. Let $\{X_t\}_{t \in \mathbb{Z}}$ be an order- p Markov chain. Then,

$$h_\infty(X_t, \dots, X_{t+n}) \geq \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) \geq h_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}). \tag{17}$$

We conclude this part of Section 3.2.1 with a result regarding order- p Markov chains that establishes a form of monotonicity for their average min-entropy.

Lemma 2. Let $\{X_t\}_{t \in \mathbb{Z}}$ be an order- p Markov chain. Then

$$\begin{aligned} \tilde{H}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) &\leq \\ \tilde{H}_\infty(X_t, \dots, X_{t+n+m} \mid X_{t-1}, \dots, X_{t-p}). \end{aligned} \tag{18}$$

This lemma establishes that the average min-entropy of an order- p Markov chain is nondecreasing as the length of the future sequence, n , increases. This property reflects the intuitive notion that the uncertainty about future states cannot decrease when considering longer future sequences. This result will be particularly useful later when we discuss an interesting property of generalized binary autoregressive models (see Remark 5).

3.2.2. Convergence Theorem for the Min-Entropy and Average Min-Entropy of Order- p Markov Chains

The purpose of the following results, which are materialized in Theorem 2 below, is to establish conditions under which an asymptotical equivalence between the average min-entropy and the min-entropy of an order- p Markov chain can be guaranteed.

Theorem 1 (Convergence Theorem [39]). Let $\{X_t\}_{t \in \mathbb{Z}}$ be an irreducible and aperiodic stationary order- p Markov chain with finite state-space S . Then, for every $x, x_{t-1}, \dots, x_{t-p} \in S$,

$$\lim_{n \rightarrow \infty} P(X_{t+n} = x \mid X_{t-1} = x_{t-1}, \dots, X_{t-p} = x_{t-p}) = P(X_t = x). \tag{19}$$

Building upon this theorem, we can now establish the asymptotic equivalence between the min-entropy and the average min-entropy for order- p Markov chains satisfying certain conditions.

Theorem 2. If $\{X_t\}_{t \in \mathbb{Z}}$ satisfies the hypothesis of the convergence theorem, i.e., $\{X_t\}_{t \in \mathbb{Z}}$ is an irreducible and aperiodic stationary order- p Markov chain with finite state-space, then

$$\begin{aligned} h_\infty(\{X_t\}_{t \in \mathbb{Z}}) &= \lim_{n \rightarrow \infty} \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = \\ &= \lim_{n \rightarrow \infty} h_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}). \end{aligned} \tag{20}$$

This theorem shows that having conditional information about the process provides no advantage asymptotically under the stated conditions, as the min-entropy and the average min-entropy converge to the same value.

3.2.3. State-Independent Maximum Transition Probability and Bitflip Symmetric Order- p Markov Chains

This section introduces two related classes of Markov chains: state-independent maximum transition probability (SIMTP) and Bitflip symmetric order- p Markov chains. We investigate the properties of these chains with a particular focus on their average min-entropy behavior. Bitflip symmetric chains are of interest as they could represent a physical symmetry of the random number generator, such as the symmetry between the two polarization states of a quantum random number generator (QRNG). Additionally, the SIMTP property enables us to perform exact min-entropy calculations for the process.

Definition 4 (State-Independent Maximum Transition Probability Order- p Markov Chain). *A stationary order- p Markov chain with state-space S is said to be a state-independent maximum transition probability (SIMTP) Markov chain if it satisfies the following property:*

$$\max_{x_t \in S} P(x_t \mid y_{t-1}, \dots, y_{t-p}) = \max_{x_t \in S} P(x_t \mid z_{t-1}, \dots, z_{t-p}) \tag{21}$$

for every $y_{t-1}, \dots, y_{t-p}, z_{t-1}, \dots, z_{t-p} \in S$.

SIMTP models are those stationary Markov chains for which the maximum transition probability is independent of the initial state sequence of length p in the chain.

Proposition 1. *Let $\{X_t\}_{t \in \mathbb{Z}}$ be an order- p SIMTP model with state space S . Then, for every nonnegative integer n and every $x_{t-1}, \dots, x_{t-p} \in S$:*

$$h_\infty(\{X_t\}_{t \in \mathbb{Z}}) = \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = -\log \left[\max_{x_t} P(x_t \mid x_{t-1}, \dots, x_{t-p}) \right]. \tag{22}$$

Hence, the min-entropy of the SIMTP process can be computed straightforwardly from its transition probability.

Definition 5 (Bitflip Symmetry in Binary Order- p Markov Chains). *A binary order- p Markov chain exhibits Bitflip Symmetry if for all states $x_{t-p}, \dots, x_{t-1}, x_t, \dots, x_{t+n} \in \{0, 1\}$ and for all nonnegative integer n , the following property holds:*

$$P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) = P(1 \oplus x_t, \dots, 1 \oplus x_{t+n} \mid 1 \oplus x_{t-1}, \dots, 1 \oplus x_{t-p}) \tag{23}$$

where \oplus represents the XOR operation.

Bitflip symmetric order- p Markov chains are those binary order- p Markov chains for which flipping the bits of all the variables in the conditional probability statement does not change the transition probability. These chains do not distinguish between 0 and 1 but still exhibit some correlation. Our interest in Bitflip symmetric order- p Markov chains is due to the following:

Lemma 3. *Let $\{X_t\}_{t \in \mathbb{Z}}$ be an order- p Bitflip-symmetric Markov chain with lag- p point-to-point correlations. Then, $\{X_t\}_{t \in \mathbb{Z}}$ is a SIMTP order- p Markov chain.*

3.2.4. Generalized Binary Autoregressive Models

The gbAR(p) model [20] is an autoregressive (AR) model for binary time series data. It allows the autoregressive parameters to take values in the range $(-1, 1)$, enabling the model to capture negative autocorrelations and alternating patterns. Despite this flexibility, the gbAR(p) model maintains a parsimonious parameterization, making it a compact but powerful model for binary data. The gbAR(p) model is a parsimonious subclass of p -th order Markov chains for binary data. While sacrificing some flexibility compared to a full p -th order Markov chain, the gbAR(p) model offers a much more compact representation.

Definition 6 (Generalized Binary Autoregressive Models [20]). Given $t \in \mathbb{Z}$, let $(a_t^{(1)}, \dots, a_t^{(p)}, b_t) \sim M(1; |\alpha_1|, \dots, |\alpha_p|, \beta)$ for some $\alpha_1, \dots, \alpha_p \in (-1, 1), \beta \in (0, 1]$ such that

$$\sum_{i=1}^p |\alpha_i| + \beta = 1 \tag{24}$$

and let $e_t \sim B(1, \epsilon_t)$ for some $\epsilon_t \in (0, 1)$. A stationary binary order- p Markov chain $\{X_t\}_{t \in \mathbb{Z}}$ that can be written in operator form as

$$X_t = \sum_{i=1}^p \left[a_t^{(i)} \mathbb{1}_{\{\alpha_i \geq 0\}} (0 \oplus \cdot) + a_t^{(i)} \mathbb{1}_{\{\alpha_i < 0\}} (1 \oplus \cdot) \right] X_{t-i} + b_t e_t, \tag{25}$$

where $\mathbb{1}$ is the indicator function and \oplus is the XOR gate, is said to be a generalized binary autoregressive or gbAR(p) model.

We will denote the array of coefficients $\alpha_1, \dots, \alpha_p$ as α , and its L1 norm (i.e., the sum of the absolute values of its components) as $|\alpha|$.

Our experiments are performed on data generated from gbAR(p) models. The rest of this section is devoted to defining the type of gbAR(p) models we will be most interested in, to proving they satisfy the hypothesis of the convergence Theorem 1, and to obtaining a formula for their min-entropy that will allow us to evaluate machine learning predictors entropy estimations (see Proposition A2 and Proposition 2 below).

Definition 7. Let $\{X_t\}_{t \in \mathbb{Z}}$ be a gbAR(p) model. Then,

- (i) $\{X_t\}_{t \in \mathbb{Z}}$ is said to be positive if $\alpha_i \geq 0$ for every $i \in \{1, \dots, p\}$.
- (ii) $\{X_t\}_{t \in \mathbb{Z}}$ is said to be a uniform noise gbAR(p) model if $e_t \sim B\left(1, \frac{1}{2}\right)$ for every $t \in \mathbb{Z}$.

Special attention will be paid to uniform noise and positive gbAR(p) models.

Proposition 2. Let $\{X_t\}_{t \in \mathbb{Z}}$ be a uniform noise and positive gbAR(p) model. Then,

$$\begin{aligned} h_\infty(\{X_t\}_{t \in \mathbb{Z}}) &= \lim_{n \rightarrow \infty} \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = \\ &= \lim_{n \rightarrow \infty} h_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = -\log\left(1 - \frac{\beta}{2}\right). \end{aligned} \tag{26}$$

Remark 3. Let $\{X_t\}_{t \in \mathbb{Z}}$ be a uniform noise gbAR(p) model with point-to-point lag- p correlations. Apart from having point-to-point lag- p correlations, $\{X_t\}_{t \in \mathbb{Z}}$ is Bitflip-symmetric by Lemma A2. Hence, $\{X_t\}_{t \in \mathbb{Z}}$ is SIMTP by Lemma 3, and therefore, for every nonnegative integer n , Proposition 1 yields

$$h_\infty(\{X_t\}_{t \in \mathbb{Z}}) = \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}). \tag{27}$$

The argumentation above is illustrated in the first two plots of Figure 2, where the equivalence of the average min-entropy per bit and the min-entropy of uniform noise gbAR(p) models with point-to-point lag- p correlations is observed regardless of the values n, p, α_p .

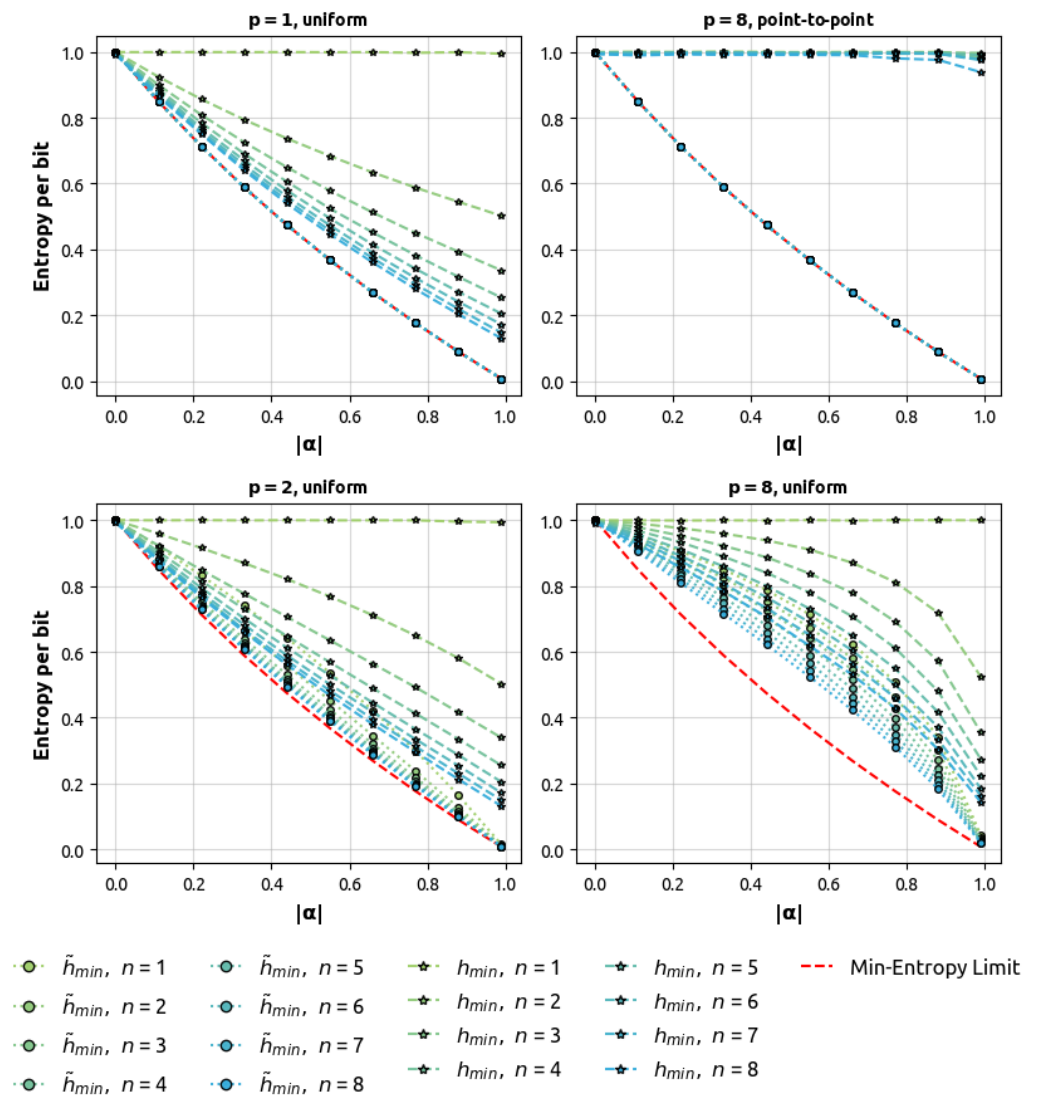


Figure 2. $|\alpha|$ dependence of average min-entropy compared with min-entropy and min-entropy limit for several sequence lengths n , correlation scales p , and autocorrelation functions (uniform and point-to-point). The data points have been evaluated numerically (see Section 4.2).

Remark 4. Let $\{X_t\}_{t \in \mathbb{Z}}$ be a uniform noise and positive gbAR(p) model. Since $\{X_t\}_{t \in \mathbb{Z}}$ is stationary by Definition 6 and it satisfies the hypothesis of the convergence Theorem 1 by Proposition A2, it follows that

$$\lim_{n \rightarrow \infty} h(X_t, \dots, X_{t+n}) = h_\infty(\{X_t\}_{t \in \mathbb{Z}}) \tag{28}$$

by Remark 2 and

$$\lim_{n \rightarrow \infty} \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = h_\infty(\{X_t\}_{t \in \mathbb{Z}}) \tag{29}$$

by Theorem 2. Moreover,

$$h_\infty(X_t, \dots, X_{t+n}) \geq \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) \tag{30}$$

by Lemma 1. The three (in)equations above are illustrated in Figure 3, where we can observe that both the average min-entropy per bit of (X_t, \dots, X_{t+n}) and the min-entropy per bit of (X_t, \dots, X_{t+n}) tend to the min-entropy of $\{X_t\}_{t \in \mathbb{Z}}$ when n goes to infinity, the former partial entropy being lower than the latter.

Remark 5. Let $\{X_t\}_{t \in \mathbb{Z}}$ be an order- p Markov chain. It should be noted that although the average min-entropy of (X_t, \dots, X_{t+n}) cannot decrease with n by Lemma 2, the average min-entropy per bit of $(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p})$ actually can (see Figure 3).

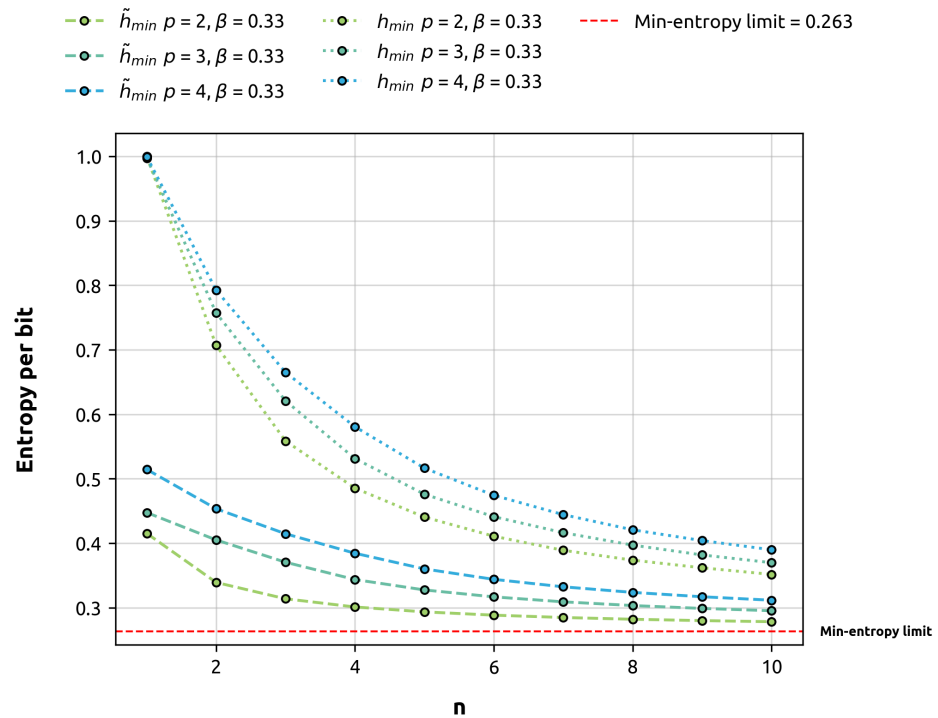


Figure 3. Asymptotic behavior of average min-entropy and min-entropy per bit in terms of the target space size n for gbAR(p) models with several correlation lengths p and fixed β . The α arrays are uniform (i.e., all their components are equal). The data points have been evaluated numerically (see Section 4.2).

4. Experimental Methodology

In this section, we outline the experimental methodology that has been carried out, which is primarily based on code implementations. Our main goal is to validate our theoretical findings, for which we generate correlated data using gbAR(p) models (see Definition 7).

Building upon Kelsey’s predictor concept [17], we use machine learning as a tool for the estimation of the min-entropy. Our methodology adopts the traditional machine learning approach, marked by separate training and evaluation phases. This strategy deviates from Kelsey’s model of continuous updates, which we consider a non-essential aspect of predictor concepts for min-entropy evaluation. Thus, our methodology, termed machine learning predictors, streamlines the process by clearly separating these stages, focusing on essential predictive capabilities without the need for constant updates.

Contrasting with the approach in [11], which examines processes failing randomness due to large periods, we focus on processes with shorter range, bit-level correlations since such correlations could be more similar to the realistic failure modes of physical and hardware-based RNGs, in line with the use of order- k or order-2 Markov chains in [17] and [18], respectively. Physical implementations inherently exhibit such behavior: sampling and holding operations introduce memory effects through exponentially decaying autocorrelation functions [40], while detector characteristics like dead time and afterpulsing create similar temporal correlations between successive bits [41]. Given this requirement for modeling realistic RNG failures with shorter-range dependencies, gbAR(p) models provide

a parsimonious parameterization that allows us to control correlations and anti-correlations, making them a suitable choice for our analysis.

These data serve as the training set for two different types of neural networks, which are tasked with predicting the next `target_bits` bits. As highlighted in the theory section (Remark 3 and Figure 2), order-1 Markov chains may present trivial cases where min-entropy and average min-entropy match. Therefore, it is important to analyze the behavior of our predictors in scenarios where this equivalence does not hold, which is the basis of our experimental approach.

All experiments were performed on an AMD EPYC 9354P 32-Core Processor (Advanced Micro Devices Inc., Santa Clara, CA, USA) (3.24 GHz) running under KVM virtualization on Debian GNU/Linux 11 (bullseye) with Linux kernel 5.10.0-32-amd64. The system includes an NVIDIA GeForce RTX 3090 GPU (NVIDIA Corporation, Santa Clara, CA, USA) with 24.576 GiB of memory running CUDA version 12.3.

The experimental framework can be structured around four primary components: the data generation process using the gbAR(p) model, the Monte Carlo simulation for the evaluation of minimum entropies, the implementation of machine learning model training and evaluation, specifically GPT-2 and a variation of RCNN (a model taken from [11]), and the integration of all data processing steps. This pipeline encompasses the generation of gbAR(p) data, its evaluation using the NIST SP 800-90B test suite, the execution of machine learning predictions and the compilation of relevant results.

For detailed documentation on code usage, parameter explanations, and additional technical details, refer to `README.md` file in the associated code repository [42].

4.1. Data Generation

Data generation is an implementation of the gbAR(p) model (Definition 6) in the `gbAR()` function. The call to this function is wrapped in the function `generate_gbAR_random_bytes()`, which leverages different autocorrelation functions, namely, point-to-point, uniform (all the components being equal), exponential, and Gaussian, to define the autocorrelation pattern through the α parameter (as defined in Definition 6).

It computes binary sequences by considering the autocorrelation defined by α and the (here always uniform) random noise term (weighted by β), sourced from OpenSSL's `RAND_bytes()` function, which generates bytes using a cryptographically secure pseudo random generator (CSPRNG). Running on Linux, this CSPRNG draws its entropy from the kernel's entropy pool (`/dev/urandom`) [43]. The same CSPRNG is employed in the `ossl_rand_mn_rvs()` function to generate samples from a multinomial distribution as required by the gbAR(p) model. These samples are then used to construct the final binary sequence according to the autocorrelation characteristics defined by the model parameters.

The `gbAR()` function includes a mechanism that discards an initial segment (here with size 10^4 bytes) of the generated binary sequence. The rationale behind this is to allow the sequence to reach a state of statistical stationarity, thereby minimizing the initial transient effects introduced in the generation.

4.2. Min-Entropy Calculation

In our approach to numerically evaluate the average min-entropy and min-entropy of the gbAR(p) processes, we employ a Monte Carlo simulation. This involves creating a program that generates 100 samples, each comprising 10^5 bytes. These samples are used to empirically estimate the joint frequencies. The computed frequencies form the basis for calculating both the min-entropy and the average min-entropy (using the known transition probabilities specific to the gbAR(p) processes).

4.3. Machine Learning Predictors

In this work, we use two distinct machine learning models to tackle the task of predicting binary sequences. The first model is an adaptation of the RCNN, while the second model is based on the GPT-2 architecture.

The selection of the RCNN and GPT-2 architectures is driven by our goal to explore prediction capabilities on binary sequences generated from autoregressive models with short-range correlations. The RCNN, as used by Truong et al., has proven effective in detecting correlations in quantum random number generators under deterministic classical noise influence. Its convolutional and recurrent layers are well suited to capture local patterns and short-term dependencies. In contrast, the transformer-based GPT-2 model, with its self-attention mechanism, offers a different approach. Although originally designed for natural language processing, we adapt it to our binary sequence prediction task to examine how it captures order- p Markov chain characteristics. Using these two models enables us to validate the theoretical finding that machine learning predictors tend to estimate average min-entropy independently of architecture, provided they can learn from the data's correlations.

4.3.1. Target Space Representation and Inference Strategies

As discussed in Section 2.2, there are various approaches for multi-token prediction in language models. Although recent work such as Gloeckle et al. [34], Stern et al. [35], and Cai et al. [36] has shown promising results in natural language processing tasks, our research focuses on a different domain. We aim to explore the relationship between model predictions and the min-entropy of the data, specifically for data with different correlations, rather than natural language.

To address the limitations of autoregressive inference strategies that rely on local decisions at each step, we propose directly predicting the entire sequence of n bits simultaneously. This approach allows us to obtain the global maximum probability for the complete sequence from the model, rather than relying on step-by-step decisions. By doing so, we aim to capture long-range dependencies and avoid the potential pitfalls of greedy, beam search, or other methods in finding globally optimal sequences.

Our method involves using different tokenization strategies for input and target spaces:

- Input space: We use binary tokenization where each token represents a single bit (0 or 1).
- Output space: We employ a tokenization where each token represents n bits, resulting in 2^n unique classes.

This tokenization approach allows the model to predict groups of bits as single tokens, considering the joint probability of the entire sequence. We believe that this method will lead to more accurate and globally optimal predictions, as it forces the model to consider the interdependencies between bits in the sequence.

4.3.2. Model Training and Evaluation Methodology

Our training dataset consists of sequences generated from the gbAR(p) model. From these sequences, we extract subsequences of length `seqLen - target_bits` bits. Both models are adapted to classify over 2^n classes, corresponding to all possible sequences of n bits. In this context, each possible combination of `target_bits` is treated as a distinct class in the classification task.

We evaluate the model prediction accuracy P_{ML} as

$$P_{ML} = \frac{n_{correct}}{n_{total}}. \quad (31)$$

Here, $n_{correct}$ represents the number of correct predictions made by the model and n_{total} denotes the total number of evaluations conducted. This measure of accuracy serves as a key indicator of the model's performance and its ability to accurately predict future bits based on the training received. The estimated min-entropy will be

$$h_{ML} = -\log(P_{ML}). \quad (32)$$

We obtain a basic approximation of the error using the Wald approximation for the binomial proportion confidence interval, as outlined in [44]. The propagation of this error yields

$$\Delta h_{ML} = \frac{z}{\text{target_bits}} \ln(2) \sqrt{\frac{\frac{1}{P_{ML}} - 1}{n_{evals}}}, \quad (33)$$

where n_{evals} is the number of evaluation sequences and z is the $1 - \alpha/2$ quantile of a standard normal distribution (1.96 for 95% confidence level).

In addition to accuracy, to assess the performance of the training procedure during the development phase, we have included the evaluation of the binary entropy of the predictions, P_e , the proportion of zeros in the prediction, P_c , and the loss.

4.3.3. RCNN Model

We use a model based on the RCNN model from the framework presented in [11]. The original implementation combines convolutional and LSTM layers followed by fully connected layers. Initially, input integers are converted into one-hot vectors. These vectors are then processed through convolutional layers with max-pooling to extract features, which are subsequently handled by the LSTM layer to capture temporal dependencies.

Our adaptation transitions from byte-based input processing to bit sequence handling, accommodating classification over a fixed number of 2^n classes, where n is the number of `target_bits`. This aligns with the original design intended for classification over fixed 2^n classes. The architecture employs an output layer with 2^n neurons and softmax activation, allowing multiclass classification. The categorical cross-entropy loss function is used for training. We use the RMSprop optimizer with a learning rate of 0.005.

Regarding the model architecture, we have slightly modified Truong's model to increase its size, including

- Convolution1D layers with 32, 64, and 128 filters and kernel sizes of 12, 6, and 3 respectively, all using 'relu' activation and 'same' padding.
- LSTM layers with 256 and 128 units, featuring return sequences and dropout layers with a rate of 0.2 for regularization.
- A final dense layer with an output size equal to `target_bits`, using sigmoid activation.

This model architecture results in approximately $7.6 \cdot 10^5$ trainable parameters.

4.3.4. GPT-2 Model

The GPT-2 model, referenced in [23], is adapted from its typical use in natural language processing, as provided by the hugging face transformers library [45]. This adaptation restructures the traditional classification model over the possible 2^n classes for the next n `target_bits`.

For processing binary sequences, we implement a custom `BinaryDataset` class. Each data entry in this dataset consists of a binary bit sequence with a length defined by the

seqLen parameter. This setup facilitates the mapping of each bit in the input sequence to the next target_bits, aligning with the classification framework.

In terms of the model architecture for the adapted GPT-2, the vocabulary size is set to $2^{\text{target_bits}}$, aligned with the number of classes in our classification framework. The specific configuration of the model includes parameters such as n_positions=512, n_ctx=512, n_embd=768, n_layer=3, and n_head=3. This configuration leads to the GPT-2 model having $\sim 21 \cdot 10^6$ trainable parameters.

For the training phase, we use the RMSprop optimizer with a learning rate of 0.0005. The CrossEntropyLoss loss function is chosen as the loss function.

We incorporate gradient scaling and accumulation in our training approach to enhance memory optimization and computational efficiency, especially important under constrained GPU availability.

4.4. Pipeline

We encapsulate in a pipeline the entire data processing for this work, from generating random numbers to saving results.

For each selection of input parameters, the method generates random bytes using the previously described gbAR(p) model. These bytes are saved to a file that is later used in the data generators within the models to train and evaluate the models. We generate new gbAR(p) sequences for each run to ensure data variability and robustness.

The pipeline runs the NIST SP 800-90B entropy assessment for non-i.i.d. data in parallel with the model execution over a sample of the generated data (here 10^7 bytes). We use the -a flag to analyze the complete dataset.

After the analysis, we meticulously compile various results, including entropy assessments, model parameters, P_{ML} values, execution time, and more, into a CSV file.

5. Results

Our primary objective is to investigate the relationship between the estimated min-entropy and the number of target_bits. To facilitate this analysis, we focus on low-entropy data for several reasons. Firstly, it ensures that models can effectively learn and capture the underlying patterns. Secondly, it improves the distinction between model predictions and inherent noise, allowing for more robust statistical analysis. In high entropy scenarios (characterized by small $|\alpha|$ values), the entropy per bit approaches 1, resembling uniform noise (see Figure 2). This proximity to maximum entropy poses challenges in assessing model performance due to overlapping confidence intervals. These intervals often encompass both the maximum entropy value of 1 and the expected theoretical value, which is also close to 1. Consequently, a large number of evaluations would be required to reduce measurement uncertainty and achieve statistical distinguishability. To address these challenges, we employ a generalized binary autoregressive model of order $p = 10$ with a uniform α vector and a uniform noise term $\beta = 0.5$. This configuration provides a balance between learnable patterns and stochastic noise, enabling effective extraction of the target_bits dependence while maintaining statistical significance in our results.

The training data for both the GPT-2 and RCNN models varied based on the number of target bits to be predicted. For tasks involving 1 to 12 target bits, 20 million bytes of raw data were used, equivalent to 125,000 128 bit training sequences each. This increased to 30 million bytes (187,499 sequences) for 13 to 15 target bits, and further to 42 million bytes (262,499 sequences) for 16 target bits. In each case, 80% of the available data was allocated for training, with the remaining 20% reserved for evaluation.

Our primary objective is to compare the min-entropy estimated by these models against the theoretical calculations and the estimations provided by the NIST SP 800-90B predictors and its overall entropy assessment. The results of these experiments are presented in Figure 4.

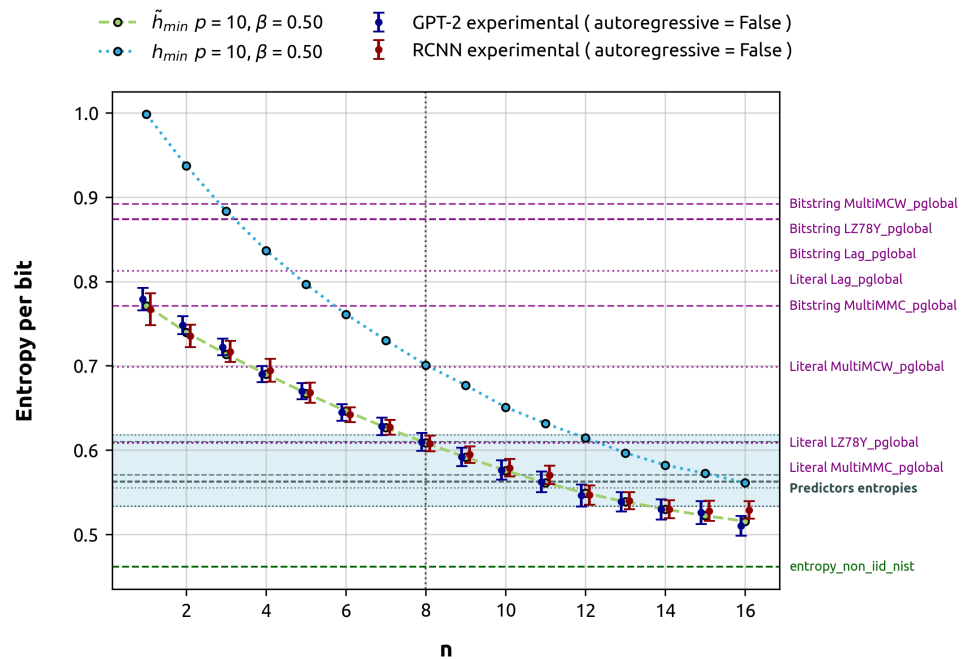


Figure 4. Theoretical minimum entropies versus machine-learning-based estimations of minimum entropy for gbAR(10) with a uniform α vector and a uniform noise term $\beta = 0.5$ (representing a low entropy scenario). For clarity in visualization, the markers representing different models are slightly offset along the x-axis. Error bars are derived using a binomial proportion confidence interval with a 95% confidence level. The results from the NIST SP 800-90B global predictor tests are emphasized. The highlighted predictor entropies are the minimum of local and global estimates, in this case, predominantly influenced by the local estimate. *Bitstring* predictors try to predict the next bit, while *Literal* predictors try to predict the next byte. The `entropy_non_iid_nist` denotes the final outcome of the NIST SP 800-90B analysis, which is the minimum of all conducted tests in the suite. The `h_min_limit` is the theoretical limit of the min-entropy, interpreted as the min-entropy per bit of the entire process. For this specific gbAR(10) configuration with positive α , both min-entropies converge to this limit.

Finally, for illustration purposes, we demonstrate how greedy decoding fails to accurately estimate the min-entropy in data with certain types of correlations. This experiment used 20 million bytes of data, equivalent to 125,000 sequences of 128 bits each. Following our standard protocol, 80% (100,000 sequences) were used for training, and 20% (25,000 sequences) for evaluation. In this case, we focus on predicting 1 to 8 target bits. For comparison, we used two gbAR(2) models with alpha vectors $\frac{1}{4}[+1, +1]$ and $\frac{1}{4}[+1, -1]$. In the case with alternating correlation signs, the global maximum probability cannot be split as the product of the local maximums, so greedy decoding leads to suboptimal predictions compared to inference over 2^n classes. In the first case ($\frac{1}{4}[+1, +1]$), the global maximum can be reached as the product of local maximums at each bit (see Remark A2), so both approaches match. This comparison illustrates how the greedy approach may lead to suboptimal predictions, as it does not consider the joint probability of the complete sequence in all cases. The results of this analysis are illustrated in Figure 5, highlighting how the greedy decoding strategy can fall short in accurately estimating min-entropy under certain correlation conditions, while performing adequately in others.

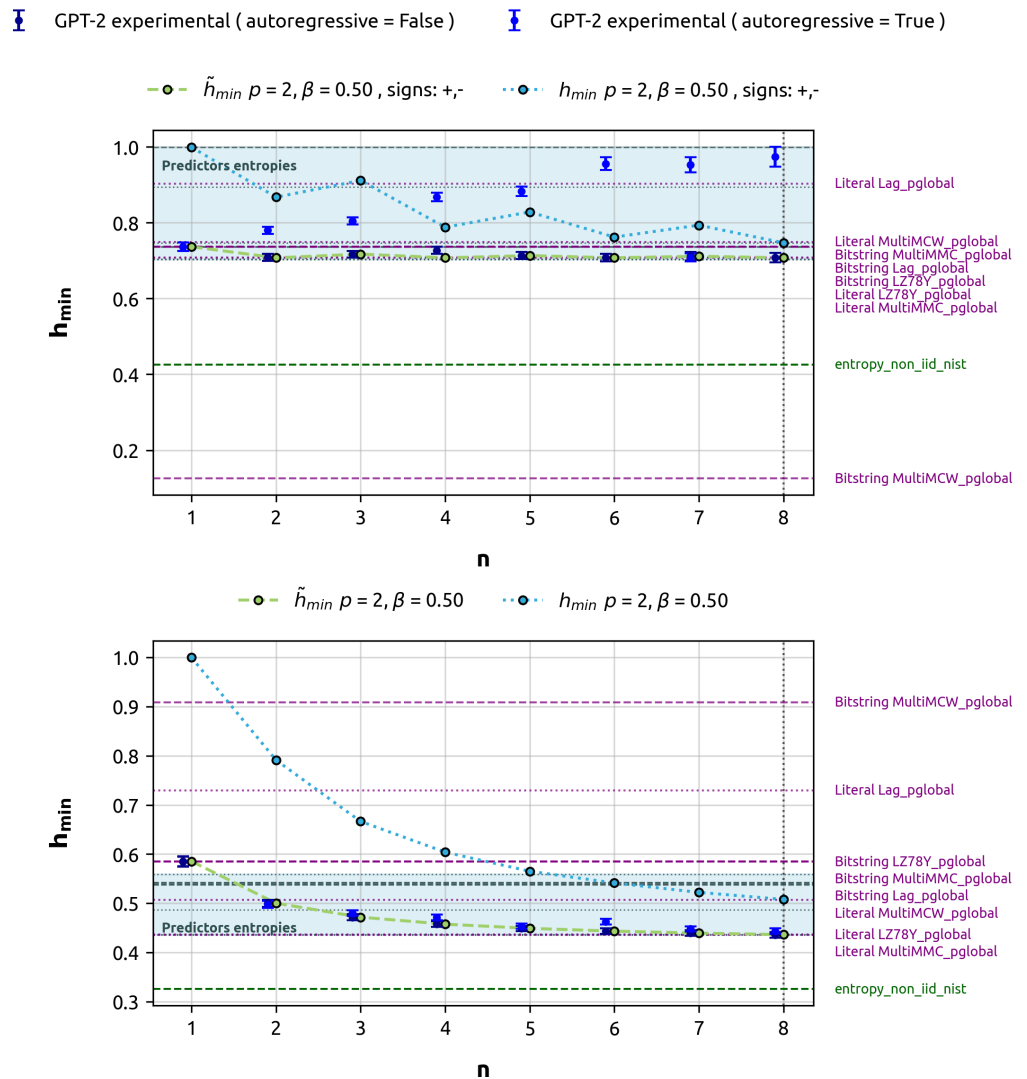


Figure 5. Comparison of min-entropy estimates: Greedy decoding vs. direct prediction over n target_bits for gbAR(2) models. The discrepancy between the experimental estimate of min-entropy and the theoretical is evident for $|\sqrt{\alpha}|[+1, -1]$, while results align for $|\sqrt{\alpha}|[+1, +1]$.

6. Discussion

Our work builds on Kelsey’s definition of predictors [17], showing that these predictors effectively estimate the average min-entropy as long as they can harness correlations and effectively model conditional probabilities, rather than joint. This distinction becomes significant when dealing with stochastic processes with complex correlation structures.

While our theoretical framework was developed for order- p Markov and gbAR(p) models, the ability of machine learning predictors to estimate average min-entropy is more general. This is because these models inherently approximate conditional probabilities and, when we evaluate prediction accuracy by selecting the most likely output, we are taking the maximum of these approximated conditional probabilities. When this is carried out over several evaluation runs on random samples, their prediction accuracy naturally computes an average of these maximum conditional probabilities—exactly the quantity appearing in the average min-entropy definition. Therefore, while NIST predictors are designed for broad coverage of potential randomness deviations, our machine learning approach can effectively capture any type of correlation structure that manifests itself in predictable patterns, not just those arising from Markov processes.

We show that while min-entropy varies with the number of bits considered, defining min-entropy per bit for the entire process is still possible. Lemma 1 establishes that the average min-entropy per bit is always lower than or equal to the min-entropy for order- p Markov chains. Although generally distinct, in specific cases (Theorem 2), both joint min-entropy and average min-entropy converge towards a process-wide min-entropy per bit. Interestingly, different states may exhibit varied decay laws despite this common limit (Figure 3), which is operationally significant when attackers have access to correlated data.

Figure 2 illustrates the interplay between correlation ‘width’ and ‘length’ in how average min-entropy approaches the min-entropy limit. This aligns with Remark 3, where average min-entropy per bit equals the min-entropy of uniform noise gbAR(p) models with point-to-point lag- p correlations, regardless of `target_bits`, p , and $|\alpha|$ values.

As we approach high-entropy limits, all entropy forms converge to the process limit, consistent across various α levels for the considered gbAR(p) models. This reaffirms that min-entropy consistently exceeds average min-entropy, as per Lemma 1.

Figure 4 shows that our min-entropy estimations from both models align with the average min-entropy and stay within the error interval. Interestingly, the NIST Bitstring global predictors, designed to estimate the entropy of 1 target bit, generally overestimate the average min-entropy for 1 target bit, with the notable exception of the MultiMMC predictor. For 8 `target_bits`, the NIST global predictors, specifically MultiMCW and Lag, tend to overestimate the average min-entropy. However, LZ78Y and MultiMMC show results that are close to the theoretical calculation. It is important to note that NIST predictors are not designed to estimate entropy in the large n limit. In the particular case studied, where the min-entropy continues to decay beyond $n = 8$, it is not surprising that the GPT-2 predictor provides a lower estimate in the $n = 16$ run. This estimate is compatible with the theoretically expected value and, moreover, does not overlap with the gray area representing the minimum between the local and global estimates. Consequently, the GPT-2 predictor’s estimate is closer to the min-entropy of the stochastic process, making it a better and more conservative estimation.

Local predictions consistently dominate the min-entropy estimate across all cases. As a result, the overall outcome of the predictor is determined by the local estimate, since the final entropy estimate is the lesser of the local and global predictions. Local predictions fall within the theoretical min-entropy for the 9–14 `target_bits` range and are significantly higher than the min-entropy limit of the process. The overall result of the NIST’s entropy non-IID test, which is the minimum of all tests in the suite, including both predictors and non-predictors, is closer to the min-entropy limit. In this specific scenario, predictors do not significantly contribute to the overall test.

Our analysis of greedy decoding versus direct prediction (Figure 5) reveals limitations in autoregressive inference approaches. For certain correlation structures (e.g., gbAR(2) with $|\sqrt{\alpha}|[+1, -1]$), greedy decoding fails to capture global maximum probability. This underscores the importance of multi-token prediction for accurately estimating min-entropy in complex correlation structures. Single-token or greedy approaches may lead to suboptimal predictions and min-entropy overestimation, emphasizing the need for methods capturing joint probabilities over multiple tokens.

In conclusion, our machine learning predictors demonstrate more consistent performance compared to NIST SP 800-90B in estimating the average min-entropy for both 1 and 8 `target_bits` in low entropy scenarios, while also providing robust estimates for larger values of n . This superior performance can be attributed, in part, to the nonparametric nature of ML min-entropy estimation. Unlike traditional methods that often assume specific underlying distributions, ML approaches allow for flexible modeling, making them particularly powerful in capturing complex, nonlinear dependencies in the data. This flexibility is

especially valuable when dealing with stochastic processes exhibiting intricate correlation structures, as it enables the model to adapt to the data's inherent patterns without being constrained by predetermined statistical assumptions.

High-entropy scenarios present additional challenges, which could require larger training runs and models. Augmenting target bits offers improved capture of long-range correlations but at a significant computational cost. As Equation (33) indicates, maintaining constant error rates with increasing target bits requires exponential growth in the evaluations, such as $P_{ML} \sim 2^{-\text{target_bits}}$ in high entropy limits. This accuracy–computation trade-off necessitates careful balancing of long-range correlation capture and practical computational requirements. Future research should focus on developing efficient algorithms or approximation methods to handle larger target bits without prohibitive computational costs, addressing the challenges of multi-token prediction, autoregressive inference limitations, and target bit scaling in entropy estimation.

7. Conclusions

Our research has revealed several key insights into the estimation of the min-entropy. We have shown that machine learning predictors are good at estimating average min-entropy, as long as they effectively harness correlations by estimating conditional probabilities. This becomes particularly significant in stochastic processes with complex correlation structures, where the difference between average min-entropy and min-entropy is relevant.

Our results show that both entropies depend on the number of `target_bits` considered. Given this important role of `target_bits`, especially in scenarios with complex correlation structures, it may be operationally significant to include assessments of both average min-entropy and min-entropy for specific `target_bits` values relevant to cryptographic (or other) scenarios. Importantly, in the examples studied, we observed that as average min-entropy decays with increasing `target_bits`, targeting only a few bits could lead to an overestimate of the min-entropy. This finding underscores the potential risks of relying on limited-scope entropy estimates in cryptographic applications. While defining min-entropy per bit for the entire process is feasible, and the entropies studied here converge towards this limit (suggesting a lower bound or worst-case scenario), this bound has not been explicitly demonstrated in this work. Therefore, the development of effective methods to estimate this limit remains essential.

We have also found that machine learning predictors can beat NIST SP 800-90B's predictors estimates in some cases, making them suitable tools to be included in entropy assessment suites.

Our research leaves several avenues open for exploration that may be of particular interest in further studies:

- Development of computationally efficient methods for estimating min-entropy in large `target_bits` and high-entropy scenarios where the number of evaluations required to maintain statistical significance grows exponentially with the number of `target_bits`.
- Validation on diverse real-world RNG data sources, with particular emphasis on high-entropy scenarios that present both computational and measurement challenges.
- Exploration of the relationship between the min-entropy of the training data, the size of the model, and the size of the training data necessary for accurate estimation of the min-entropy. This investigation goes beyond aligning the prediction accuracy of the model with the theoretical curves; it aims to provide a deeper understanding of the model learning capacity at various entropy levels. This knowledge could inform the appropriate scaling of computational resources and potentially offer improved estimates by considering theoretical bounds on min-entropy estimation.

In conclusion, while our work has advanced the understanding of min-entropy estimation through machine learning, it also highlights the practical complexity of this method and the need for more research to address its challenges.

Author Contributions: Conceptualization, J.B.-R., V.L., F.A.M. and D.D.-S.; methodology, J.B.-R. and V.L.; software, J.B.-R.; validation, J.B.-R. and V.L.; formal analysis, J.B.-R. and V.L.; investigation, J.B.-R. and V.L.; data curation, J.B.-R. and V.L.; writing—original draft preparation, J.B.-R. and V.L.; writing—review and editing, J.B.-R., V.L. and F.A.M.; visualization, J.B.-R.; supervision, F.A.M. and D.D.-S.; project administration, F.A.M.; funding acquisition, F.A.M. and D.D.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is part of the R&D project TED2021-130369B-C32, funded by MCIN/AEI/10.13039/501100011033 and by the “European Union NextGenerationEU/PRTR” and is part of the project COMPROMISE PID2020-113795RB-C32/AEI/10.13039/501100011033. In addition, it was partially supported by project i-SHAPER PRTR-INCIBE-2023/00623/001, which is being carried out within the framework of the Recovery, Transformation, and Resilience Plan funds, funded by the European Union (Next Generation).

Data Availability Statement: The data used in this study were simulated using Markov chains with parameters and procedures fully detailed in the manuscript. The complete code for data generation and analysis is available in our public GitHub repository [42]. The data generation process uses OpenSSL as a source of randomness through the underlying Linux OS. Due to the stochastic nature of the data generation, exact reproduction will depend on the entropy source used, but with sufficient entropy quality, all results should be reproducible using the provided code and parameters. For detailed documentation on code usage, parameter explanations, and additional technical details, please refer to the README.md file in the GitHub repository.

Acknowledgments: The authors want to thank Miguel Angel Hombrados Herrera and Gonzalo Martínez Ruiz de Arcaute for their help and fruitful comments.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CRNG	Congruential random number generator
ECL	External-cavity semiconductor laser
gbAR	Generalized binary autoregressive
GPT-2	Generative pre-trained transformer 2
i.i.d.	Independent and identically distributed
LSTM	Long short-term memory
ML	Machine learning
NIST	National Institute of Standards and Technology
NRNG	Nondeterministic random number generator
QRNG	Quantum random number generator
RCNN	Recurrent convolutional neural network
RNG	Random number generator
SIMTP	State-independent maximum transition probability
SP	Special Publication
TPA	Temporal pattern attention
XOR	Exclusive OR

Appendix A. Proofs and Auxiliary Results

The purpose of this appendix is to provide the proofs for the results stated in Section 3 and to include some additional auxiliary results needed to prove them.

Appendix A.1. Order- p Markov Chains

We start by reviewing the results presented in Section 3.2.

Proof of Lemma 1. On the one hand, note that

$$\begin{aligned} & \left\langle \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right\rangle_{x_{t-1}, \dots, x_{t-p}} = \\ & \sum_{x_{t-p}, \dots, x_{t-1}} P(x_{t-p}, \dots, x_{t-1}) \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \leq \\ & \sum_{x_{t-p}, \dots, x_{t-1}} P(x_{t-p}, \dots, x_{t-1}) \max_{x_{t-p}, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) = \\ & \max_{x_{t-p}, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}). \end{aligned} \tag{A1}$$

The inequality

$$\tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) \geq h_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}). \tag{A2}$$

follows from (A1) taking logarithms, dividing by $n + 1$, and changing the sign.

On the other hand, the inequality

$$h_\infty(X_t, \dots, X_{t+n}) \geq \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p})$$

easily follows writing

$$\begin{aligned} & H_\infty(X_t, \dots, X_{t+n}) = \\ & -\log \left[\max_{x_t, \dots, x_{t+n}} \sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right] \end{aligned} \tag{A3}$$

and taking into account that

$$\begin{aligned} & \max_{x_t, \dots, x_{t+n}} \sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \leq \\ & \sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}). \end{aligned} \tag{A4}$$

□

Proof of Lemma 2. We have that

$$\begin{aligned} & P(x_t, \dots, x_{t+n+m} \mid x_{t-1}, \dots, x_{t-p}) = \\ & P(x_{t+n+1}, \dots, x_{t+n+m} \mid x_{t+n}, \dots, x_{t-p}) P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \leq \\ & P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}). \end{aligned} \tag{A5}$$

Since this inequality is independent of the realizations $x_{t+n+1}, \dots, x_{t+n+m}$, it follows that

$$\begin{aligned} & \max_{x_t, \dots, x_{t+n+m}} P(x_t, \dots, x_{t+n+m} \mid x_{t-1}, \dots, x_{t-p}) \leq \\ & \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}). \end{aligned} \tag{A6}$$

Since logarithms are monotonically increasing functions, we conclude that

$$\begin{aligned}
 & \tilde{H}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = \\
 & -\log \left[\sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right] \leq \\
 & -\log \left[\sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t, \dots, x_{t+n+m}} P(x_t, \dots, x_{t+n+m} \mid x_{t-1}, \dots, x_{t-p}) \right] = \\
 & \tilde{H}_\infty(X_t, \dots, X_{t+n+m} \mid X_{t-1}, \dots, X_{t-p}).
 \end{aligned} \tag{A7}$$

□

Proof of Theorem 1. Since this result is a restatement of [39] (Theorem 8), we believe that some explanation is necessary. First of all, what [39] (Theorem 8) states is that for every $x, x_{t-1}, \dots, x_{t-p} \in S$,

$$\lim_{n \rightarrow \infty} P(X_{t+n} = x \mid X_{t-1} = x_{t-1}, \dots, X_{t-p} = x_{t-p}) = \pi(x) \tag{A8}$$

where π is a stationary distribution whose existence is required as a hypothesis. In our case, the existence of a stationary distribution π follows from [39] (Theorem 7) because we are assuming that $\{X_t\}_{t \in \mathbb{Z}}$ is irreducible. Having said that, taking the stationarity into account,

$$\begin{aligned}
 & P(X_t = x) = \lim_{n \rightarrow \infty} P(X_t = x) = \lim_{n \rightarrow \infty} P(X_{t+n} = x) = \\
 & \lim_{n \rightarrow \infty} \sum_{x_{t-1}, \dots, x_{t-n}} P(X_{t-1} = x_{t-1}, \dots, X_{t-p} = x_{t-p}) P(X_{t+n} = x \mid X_{t-1} = x_{t-1}, \dots, X_{t-p} = x_{t-p}) = \\
 & \sum_{x_{t-1}, \dots, x_{t-n}} P(X_{t-1} = x_{t-1}, \dots, X_{t-p} = x_{t-p}) \lim_{n \rightarrow \infty} P(X_{t+n} = x \mid X_{t-1} = x_{t-1}, \dots, X_{t-p} = x_{t-p}) = \\
 & \sum_{x_{t-1}, \dots, x_{t-n}} P(X_{t-1} = x_{t-1}, \dots, X_{t-p} = x_{t-p}) \pi(x) = \pi(x)
 \end{aligned} \tag{A9}$$

and our restatement of [39] (Theorem 8) follows. □

Proof of Theorem 2. On the one hand, denoting $\tau = \lfloor \sqrt{n} \rfloor$, we have that

$$\begin{aligned}
 & P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) = \frac{P(x_{t-p}, \dots, x_{t+n})}{P(x_{t-1}, \dots, x_{t-p})} \leq \\
 & \frac{P(x_{t-p}, \dots, x_{t-1}, x_{t+\tau}, \dots, x_{t+n})}{P(x_{t-1}, \dots, x_{t-p})} = P(x_{t+\tau}, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}).
 \end{aligned} \tag{A10}$$

Therefore,

$$\begin{aligned}
 & \sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \leq \\
 & \sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_{t+\tau}, \dots, x_{t+n}} P(x_{t+\tau}, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}).
 \end{aligned} \tag{A11}$$

In particular,

$$\begin{aligned}
 & -\frac{1}{n+1} \log \left(\sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right) \geq \\
 & -\frac{1}{n+1} \log \left(\sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_{t+\tau}, \dots, x_{t+n}} P(x_{t+\tau}, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right).
 \end{aligned} \tag{A12}$$

Now, the left hand side of the inequality (A12) is

$$\tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p})$$

and the limit of the right hand side of the inequality (A12) as n goes to infinity is $h_\infty(\{X_t\}_{t \in \mathbb{Z}})$ because of the convergence theorem and the stationarity. Using Lemma 1 we conclude that

$$h_\infty(\{X_t\}_{t \in \mathbb{Z}}) \geq \lim_{n \rightarrow \infty} \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) \geq h_\infty(\{X_t\}_{t \in \mathbb{Z}}) \tag{A13}$$

and the equality

$$h_\infty(\{X_t\}_{t \in \mathbb{Z}}) = \lim_{n \rightarrow \infty} \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) \tag{A14}$$

follows. On the other hand

$$\begin{aligned}
 & h_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = \\
 & -\frac{1}{n+1} \log \left[\max_{x_{t-p}, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right] \geq \\
 & -\frac{1}{n+1} \log \left[\max_{x_{t-p}, \dots, x_{t+n}} P(x_{t+\tau}, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right] \xrightarrow{n \rightarrow \infty} h_\infty(\{X_t\}_{t \in \mathbb{Z}}).
 \end{aligned} \tag{A15}$$

Note that the inequality of (A15) holds because of (A10), and the limit of (A15) holds because of the convergence theorem and the stationarity. Using Lemma 1, we conclude that

$$h_\infty(\{X_t\}_{t \in \mathbb{Z}}) \geq \lim_{n \rightarrow \infty} h_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) \geq h_\infty(\{X_t\}_{t \in \mathbb{Z}}) \tag{A16}$$

and the result follows. \square

Appendix A.2. State-Independent Maximum Transition Probability and Bitflip Symmetric Order- p Markov Chains

The following results are intended to prove that the average min-entropy of SIMTP models coincides with their min-entropy, as claimed in Proposition 1.

Lemma A1. *Let $\{X_t\}_{t \in \mathbb{Z}}$ be a SIMTP order- p Markov chain. Then,*

$$\begin{aligned}
 & \left\langle \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right\rangle_{x_{t-1}, \dots, x_{t-p}} = \\
 & \max_{x_{t-p}, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}).
 \end{aligned} \tag{A17}$$

Proof. Let us write

$$\begin{aligned}
 & P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) = \\
 & \prod_{i=0}^n P(x_{t+i} \mid x_{t+i-1}, \dots, x_{t+i-p}).
 \end{aligned} \tag{A18}$$

Since $P(x_{t+i} \mid x_{t+i-1}, \dots, x_{t+i-p})$ is independent of $x_{t+i-1}, \dots, x_{t+i-p}$ for every $i \in \{0, \dots, n\}$, it follows from (A18) that

$$\begin{aligned} \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) &= \\ \prod_{i=0}^n \max_{x_{t+i}} P(x_{t+i} \mid x_{t+i-1}, \dots, x_{t+i-p}). & \end{aligned} \tag{A19}$$

Using (A19), the independence of $P(x_{t+i} \mid x_{t+i-1}, \dots, x_{t+i-p})$ with respect to $x_{t+i-1}, \dots, x_{t+i-p}$ for every $i \in \{0, \dots, n\}$ and the fact that the sum of probabilities of all the outcomes within a sample space is 1, we obtain

$$\begin{aligned} \left\langle \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right\rangle_{x_{t-1}, \dots, x_{t-p}} &= \\ \sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) &= \\ \sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \prod_{i=0}^n \max_{x_{t+i}} P(x_{t+i} \mid x_{t+i-1}, \dots, x_{t+i-p}) &= \\ \prod_{i=0}^n \max_{x_{t+i}} P(x_{t+i} \mid x_{t+i-1}, \dots, x_{t+i-p}) &= \\ \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}). & \end{aligned} \tag{A20}$$

□

Proposition A1. Any order- p SIMTP satisfies the following decomposition:

$$\begin{aligned} H_\infty(X_t, \dots, X_{t+n}) &= \\ H_\infty(X_t, \dots, X_{t+p-1}) + H_\infty(X_{t+p}, \dots, X_{t+n} \mid X_{t+p-1}, \dots, X_t) & \end{aligned} \tag{A21}$$

Proof. Let us write

$$\begin{aligned} H_\infty(X_t, \dots, X_{t+n}) &= -\log \left[\max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_n) \right] = \\ -\log \left[\max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+p-1}) P(x_{t+p}, \dots, x_{t+n} \mid x_{t+p-1}, \dots, x_t) \right]. & \end{aligned} \tag{A22}$$

If the process is SIMTP, then we can reach the maximum over $P(x_{t+p}, \dots, x_{t+n})$ independently of the values required to maximize over $P(x_t, \dots, x_{t+p-1})$. In other words, we can maximize both probabilities independently:

$$\begin{aligned} H_\infty(X_t, \dots, X_{t+n}) &= \\ -\log \left[\max_{x_t, \dots, x_{t+p-1}} P(x_t, \dots, x_{t+p-1}) \right] - \log \left[\max_{x_t, \dots, x_{t+n}} P(x_{t+p}, \dots, x_{t+n} \mid x_{t+p-1}, \dots, x_t) \right] &= \\ H_\infty(X_t, \dots, X_{t+p-1}) + H_\infty(X_{t+p}, \dots, X_{t+n} \mid X_{t+p-1}, \dots, X_t). & \end{aligned} \tag{A23}$$

□

Proof of Proposition 1. On the one hand, by Proposition A1, we have

$$\begin{aligned}
 & H_\infty(X_t, \dots, X_{t+n}) = \\
 & H_\infty(X_t, \dots, X_{t+p-1}) + H_\infty(X_{t+p}, \dots, X_{t+n} \mid X_{t+p-1}, \dots, X_t) = \\
 & H_\infty(X_t, \dots, X_{t+p-1}) - \log \left[\max_{x_t} P(x_t \mid x_{t-1}, \dots, x_{t-p})^{n-p} \right].
 \end{aligned} \tag{A24}$$

Then, for finite p , as the first term is bounded,

$$\begin{aligned}
 h_\infty(\{X_t\}_{t \in \mathbb{Z}}) &= \lim_{n \rightarrow \infty} \frac{1}{n+1} H_\infty(X_t, \dots, X_{t+n}) = \\
 & - \log \left[\max_{x_t} P(x_t \mid x_{t-1}, \dots, x_{t-p}) \right].
 \end{aligned} \tag{A25}$$

On the other hand,

$$\begin{aligned}
 & \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = \\
 & - \log \left[\left(\sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right)^{\frac{1}{n+1}} \right] = \\
 & - \log \left[\left(\sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \max_{x_t} P(x_t \mid x_{t-1}, \dots, x_{t-p})^n \right)^{\frac{1}{n+1}} \right] = \\
 & - \log \left[\left(\sum_{x_{t-1}, \dots, x_{t-p}} P(x_{t-1}, \dots, x_{t-p}) \right)^{\frac{1}{n+1}} \max_{x_t} P(x_t \mid x_{t-1}, \dots, x_{t-p}) \right] = \\
 & - \log \left[\max_{x_t} P(x_t \mid x_{t-1}, \dots, x_{t-p}) \right].
 \end{aligned} \tag{A26}$$

The result follows putting (A25) and (A26) together. \square

We end this subsection of the Appendix A by proving that Bitflip-symmetric Markov Chains with lag- p point-to-point correlations are SIMTP, as stated in Lemma 3.

Proof of Lemma 3. Let $x_{t-p} \in \{0, 1\}$. Then, by the definition of Bitflip symmetry,

$$\begin{aligned}
 \max_{x_t} P(x_t \mid x_{t-p}) &= \max\{P(X_t = 0 \mid x_{t-p}), P(X_t = 1 \mid x_{t-p})\} = \\
 & \max\{P(X_t = 1 \mid 1 \oplus x_{t-p}), P(X_t = 0 \mid 1 \oplus x_{t-p})\} = \\
 & \max_{x_t} P(x_t \mid 1 \oplus x_{t-p})
 \end{aligned} \tag{A27}$$

and the result follows. \square

Appendix A.3. Generalized Binary Autoregressive Models

The last subsection of the Appendix A is devoted to gather some interesting properties of gbAR(p) models. In particular, they will allow us to prove the formula for calculating the min-entropy of uniform noise and positive gbAR(p) models stated in Proposition 2.

Remark A1. Let $\{X_t\}_{t \in \mathbb{Z}}$ be a uniform noise gbAR(p) model. Then $P(X_t = 0) = P(X_t = 1) = \frac{1}{2}$ for every $t \in \mathbb{Z}$ by [20] (Lemma 1).

Remark A2. By [20] (Lemma 1), the transition probabilities of a gbAR(p) model $\{X_t\}_{t \in \mathbb{Z}}$ can be written as

$$P(x_t \mid x_{t-1}, \dots, x_{t-p}) = \sum_{i=1}^p [\mathbb{1}_{\{\alpha_i \geq 0\}} |\alpha_i| \cdot (1 \oplus x_t \oplus x_{t-i}) + \mathbb{1}_{\{\alpha_i < 0\}} |\alpha_i| \cdot (x_t \oplus x_{t-i})] + \beta \cdot P(e_t = x_t). \tag{A28}$$

If $\{X_t\}_{t \in \mathbb{Z}}$ is a uniform noise and positive gbAR(p) model, this is written as

$$P(x_t \mid x_{t-1}, \dots, x_{t-p}) = \sum_{i=1}^p \alpha_i \cdot (1 \oplus x_t \oplus x_{t-i}) + \frac{\beta}{2} \tag{A29}$$

and this conditional probability reaches the maximum value

$$\max_{x_{t-1}, \dots, x_{t-p}} P(x_t \mid x_{t-1}, \dots, x_{t-p}) = \sum_{i=1}^p \alpha_i + \frac{\beta}{2} = 1 - \frac{\beta}{2} \tag{A30}$$

when $x_t = x_{t-1} = \dots = x_{t-p}$. In particular, a realization x_{t-p}, \dots, x_{t+n} of X_{t-p}, \dots, X_{t+n} such that

$$x_{t-p} = \dots = x_{t+n}$$

maximizes the conditional probability $P(x_k \mid x_{k-1}, \dots, x_{k-p})$ for every $k \in \{t, \dots, t+n\}$, and it follows that

$$\begin{aligned} \max_{x_{t-p}, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) &= \\ \max_{x_{t-p}, \dots, x_{t+n}} \prod_{k=t}^{t+n} P(x_k \mid x_{k-1}, \dots, x_{k-p}) &= \\ \prod_{k=t}^{t+n} \max_{x_k, \dots, x_{k-p}} P(x_k \mid x_{k-1}, \dots, x_{k-p}) &= \left(1 - \frac{\beta}{2}\right)^{n+1}. \end{aligned} \tag{A31}$$

Remark A3. Note that a gbAR(p) model has point-to-point lag-p correlations if $a_t^{(i)} = \delta_{i,p}$ for every $i \in \{1, \dots, p\}$ and every $t \in \mathbb{Z}$. This is achieved when $\alpha_i = 0$ for every $i \in \{1, \dots, p-1\}$.

Lemma A2. A uniform noise gbAR(p) model with point-to-point lag-p correlations is Bitflip-symmetric.

Proof. By the definition of conditional probability and the definition of the gbAR(p) model with point-to-point lag-p correlations,

$$P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) = \prod_{i=0}^n P(x_{t+i} \mid x_{t+i-p}). \tag{A32}$$

Then, by Remarks A1 and A2,

$$\begin{aligned} P(x_{t+i} \mid x_{t+i-p}) &= \\ &[\mathbb{1}_{\{\alpha_p \geq 0\}} |\alpha_p| \cdot (1 \oplus x_{t+i} \oplus x_{t+i-p}) + \\ &\mathbb{1}_{\{\alpha_p < 0\}} |\alpha_p| \cdot (x_{t+i} \oplus x_{t+i-p})] + \frac{\beta}{2} = \\ &[\mathbb{1}_{\{\alpha_p \geq 0\}} |\alpha_p| \cdot (1 \oplus 1 \oplus x_{t+i} \oplus 1 \oplus x_{t+i-p}) + \\ &\mathbb{1}_{\{\alpha_p < 0\}} |\alpha_p| \cdot (1 \oplus x_{t+i} \oplus 1 \oplus x_{t+i-p})] + \frac{\beta}{2} = \\ &P(1 \oplus x_{t+i} \mid 1 \oplus x_{t+i-p}). \end{aligned} \tag{A33}$$

Putting (A32) and (A33) together,

$$\begin{aligned}
 P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) &= \\
 \prod_{i=0}^n P(x_{t+i} \mid x_{t+i-p}) &= \prod_{i=0}^n P(1 \oplus x_{t+i} \mid 1 \oplus x_{t+i-p}) = \\
 P(1 \oplus x_t, \dots, 1 \oplus x_{t+n} \mid 1 \oplus x_{t-1}, \dots, 1 \oplus x_{t-p}). &
 \end{aligned}
 \tag{A34}$$

□

Proposition A2. Let $\{X_t\}_{t \in \mathbb{Z}}$ be a uniform noise and positive gbAR(p) model. Then $\{X_t\}_{t \in \mathbb{Z}}$ satisfies the hypothesis of the convergence theorem, i.e., $\{X_t\}_{t \in \mathbb{Z}}$ is an irreducible and aperiodic stationary order- p Markov chain with finite state space.

Proof. $\{X_t\}_{t \in \mathbb{Z}}$ is a stationary order- p Markov chain with finite state-space $S = \{0, 1\}$ by Definition 6. Now, $\{X_t\}_{t \in \mathbb{Z}}$ is irreducible because for every $x_{t-p}, \dots, x_t \in S$, we have

$$P(x_t \mid x_{t-1}, \dots, x_{t-p}) = \sum_{i=1}^p \alpha_i \cdot (1 \oplus x_t \oplus x_{t-i}) + \frac{\beta}{2} > 0.
 \tag{A35}$$

Finally, since

$$\sum_{x_{t-2}, \dots, x_{t-p}} P(x_{t-2}, \dots, x_{t-p}) = 1,$$

there exist $y_{t-2}, \dots, y_{t-p} \in S$ such that $P(y_{t-2}, \dots, y_{t-p}) > 0$. Then, by Equation (A35),

$$\begin{aligned}
 P(x_t \mid x_{t-1}) &= \sum_{x_{t-2}, \dots, x_{t-p}} P(x_{t-2}, \dots, x_{t-p}) P(x_t \mid x_{t-1}, x_{t-2}, \dots, x_{t-p}) \geq \\
 &P(y_{t-2}, \dots, y_{t-p}) P(x_t \mid x_{t-1}, y_{t-2}, \dots, y_{t-p}) > 0
 \end{aligned}
 \tag{A36}$$

and we conclude that $\{X_t\}_{t \in \mathbb{Z}}$ is aperiodic. □

Proof of Proposition 2. Since $\{X_t\}_{t \in \mathbb{Z}}$ satisfies the hypothesis of the convergence Theorem 1 by Proposition A2, the equalities

$$\begin{aligned}
 h_\infty(\{X_t\}_{t \in \mathbb{Z}}) &= \lim_{n \rightarrow \infty} \tilde{h}_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) = \\
 &\lim_{n \rightarrow \infty} h_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p})
 \end{aligned}
 \tag{A37}$$

follow from Theorem 2. Now,

$$\begin{aligned}
 h_\infty(X_t, \dots, X_{t+n} \mid X_{t-1}, \dots, X_{t-p}) &= \\
 -\frac{1}{n+1} \log \left[\max_{x_{t-p}, \dots, x_{t+n}} P(x_t, \dots, x_{t+n} \mid x_{t-1}, \dots, x_{t-p}) \right] &= \\
 -\frac{1}{n+1} \log \left[\max_{x_{t-p}, \dots, x_{t+n}} \prod_{i=0}^n P(x_{t+i} \mid x_{t+i-1}, \dots, x_{t+i-p}) \right] &= \\
 -\frac{1}{n+1} \log \left[\prod_{i=0}^n \left(\sum_{j=1}^p \alpha_j + \frac{\beta}{2} \right) \right] &= -\frac{1}{n+1} \log \left[\left(\sum_{i=1}^p \alpha_i + \frac{\beta}{2} \right)^{n+1} \right] = \\
 -\frac{1}{n+1} \log \left[\left(1 - \frac{\beta}{2} \right)^{n+1} \right] &= -\log \left(1 - \frac{\beta}{2} \right).
 \end{aligned}
 \tag{A38}$$

□

References

1. Bassham, L.E., III; Rukhin, A.L.; Soto, J.; Nechvatal, J.R.; Smid, M.E.; Barker, E.B.; Leigh, S.D.; Levenson, M.; Vangel, M.; Banks, D.L.; et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. In *NIST Special Publication*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2010; Volume 800.
2. Marsaglia, G. The Marsaglia Random Number CDROM Including the Diehard Battery of Tests of Randomness. 2008. Available online: <http://www.stat.fsu.edu/pub/diehard/> (accessed on 28 June 2024).
3. Abbott, A.A.; Calude, C.S.; Dinneen, M.J.; Huang, N. Experimentally probing the algorithmic randomness and incomputability of quantum randomness. *Phys. Scr.* **2019**, *94*, 045103. [[CrossRef](#)]
4. Calude, C.S.; Dinneen, M.J.; Dumitrescu, M.; Svozil, K. Experimental evidence of quantum randomness incomputability. *Phys. Rev. A* **2010**, *82*, 022102. [[CrossRef](#)]
5. Kavulich, J.T.; Van Deren, B.P.; Schlosshauer, M. Searching for evidence of algorithmic randomness and incomputability in the output of quantum random number generators. *Phys. Lett. A* **2021**, *388*, 127032. [[CrossRef](#)]
6. Bird, J.J.; Ekárt, A.; Faria, D.R. On the effects of pseudorandom and quantum-random number generators in soft computing. *Soft Comput.* **2020**, *24*, 9243–9256. [[CrossRef](#)]
7. Turan, M.S.; Barker, E.; Kelsey, J.; McKay, K.A.; Baish, M.L.; Boyle, M. Recommendation for the entropy sources used for random bit generation. In *NIST Special Publication*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2018; Volume 800.
8. Abraham, N.; Watanabe, K.; Taniguchi, T.; Majumdar, K. A High-Quality Entropy Source Using van der Waals Heterojunction for True Random Number Generation. *ACS Nano* **2022**, *16*, 5898–5908. [[CrossRef](#)]
9. Islam, M.S. Using ECG signal as an entropy source for efficient generation of long random bit sequences. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 5144–5155. [[CrossRef](#)]
10. Li, X.; Stanwicks, P.; Provelengios, G.; Tessier, R.; Holcomb, D. Jitter-based adaptive true random number generation for FPGAs in the cloud. In Proceedings of the 2020 International Conference on Field-Programmable Technology (ICFPT), Maui, HI, USA, 9–11 December 2020; pp. 112–119.
11. Truong, N.D.; Haw, J.Y.; Assad, S.M.; Lam, P.K.; Kavehei, O. Machine learning cryptanalysis of a quantum random number generator. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 403–414. [[CrossRef](#)]
12. Yang, J.; Zhu, S.; Chen, T.; Ma, Y.; Lv, N.; Lin, J. Neural network based min-entropy estimation for random number generators. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Singapore, 8–10 August 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 231–250.
13. Lv, N.; Chen, T.; Zhu, S.; Yang, J.; Ma, Y.; Jing, J.; Lin, J. High-efficiency min-entropy estimation based on neural network for random number generators. *Secur. Commun. Netw.* **2020**, *2020*, 4241713.
14. Li, C.; Zhang, J.; Sang, L.; Gong, L.; Wang, L.; Wang, A.; Wang, Y. Deep learning-based security verification for a random number generator using white chaos. *Entropy* **2020**, *22*, 1134. [[CrossRef](#)]
15. Feng, Y.; Hao, L. Testing randomness using artificial neural network. *IEEE Access* **2020**, *8*, 163685–163693. [[CrossRef](#)]
16. Li, H.; Zhang, J.; Li, Z.; Liu, J.; Wang, Y. Improvement of Min-entropy evaluation based on pruning and quantized deep neural network. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 1410–1420. [[CrossRef](#)]
17. Kelsey, J.; McKay, K.A.; Sönmez Turan, M. Predictive models for min-entropy estimation. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2015: 17th International Workshop, Saint-Malo, France, 13–16 September 2015; Proceedings 17; Springer: Berlin/Heidelberg, Germany, 2015; pp. 373–392.
18. Zhu, S.; Ma, Y.; Chen, T.; Lin, J.; Jing, J. Analysis and improvement of entropy estimators in NIST SP 800-90B for non-IID entropy sources. In *IACR Transactions on Symmetric Cryptology*; Ruhr University Bochum: Bochum, Germany, 2017; pp. 151–168.
19. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the Advances in Cryptology—EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; Proceedings 23; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.
20. Jentsch, C.; Reichmann, L. Generalized binary time series models. *Econometrics* **2019**, *7*, 47. [[CrossRef](#)]
21. Shannon, C.E. Prediction and entropy of printed English. *Bell Syst. Tech. J.* **1951**, *30*, 50–64. [[CrossRef](#)]
22. Cachin, C. Entropy Measures and Unconditional Security in Cryptography. Ph.D. Thesis, ETH Zurich, Zurich, Switzerland, 1997.
23. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
24. Vijayakumar, A.K.; Cogswell, M.; Selvaraju, R.R.; Sun, Q.; Lee, S.; Crandall, D.; Batra, D. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv* **2016**, arXiv:1610.02424.
25. Shao, L.; Gouws, S.; Britz, D.; Goldie, A.; Strobe, B.; Kurzweil, R. Generating high-quality and informative conversation responses with sequence-to-sequence models. *arXiv* **2017**, arXiv:1701.03185.
26. Fan, A.; Lewis, M.; Dauphin, Y. Hierarchical neural story generation. *arXiv* **2018**, arXiv:1805.04833.

27. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The curious case of neural text degeneration. *arXiv* **2019**, arXiv:1904.09751.
28. Ackley, D.H.; Hinton, G.E.; Sejnowski, T.J. A learning algorithm for Boltzmann machines. *Cogn. Sci.* **1985**, *9*, 147–169.
29. Li, J.; Monroe, W.; Jurafsky, D. Learning to decode for future success. *arXiv* **2017**, arXiv:1701.06549.
30. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
31. Serdyuk, D.; Ke, N.R.; Sordoni, A.; Trischler, A.; Pal, C.; Bengio, Y. Twin networks: Matching the future for sequence generation. *arXiv* **2017**, arXiv:1708.06742.
32. Lawrence, C.; Kotnis, B.; Niepert, M. Attending to future tokens for bidirectional sequence generation. *arXiv* **2019**, arXiv:1908.05915.
33. Qi, W.; Yan, Y.; Gong, Y.; Liu, D.; Duan, N.; Chen, J.; Zhang, R.; Zhou, M. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv* **2020**, arXiv:2001.04063.
34. Gloeckle, F.; Idrissi, B.Y.; Rozière, B.; Lopez-Paz, D.; Synnaeve, G. Better & Faster Large Language Models via Multi-token Prediction. *arXiv* **2024**, arXiv:2404.19737.
35. Stern, M.; Shazeer, N.; Uszkoreit, J. Blockwise parallel decoding for deep autoregressive models. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 10107–10116.
36. Cai, T.; Li, Y.; Geng, Z.; Peng, H.; Lee, J.D.; Chen, D.; Dao, T. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv* **2024**, arXiv:2401.10774.
37. Ching, W.K.; Ng, M.K. Markov chains. In *Models, Algorithms and Applications*; International Series in Operations Research & Management Science; Springer: New York, NY, USA, 2006; Volume 83, pp. 1–208.
38. Raftery, A.E. A model for high-order Markov chains. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1985**, *47*, 528–539. [[CrossRef](#)]
39. Bozorgmanesh, H.; Hajarian, M. Convergence of a transition probability tensor of a higher-order Markov chain to the stationary probability vector. *Numer. Linear Algebra Appl.* **2016**, *23*, 972–988. [[CrossRef](#)]
40. Bagini, V.; Bucci, M. A design of reliable true random number generator for cryptographic applications. In *Proceedings of the Cryptographic Hardware and Embedded Systems: First International Workshop, CHES'99, Worcester, MA, USA, 12–13 August 1999*; Proceedings 1; Springer: Berlin/Heidelberg, Germany, 1999; pp. 204–218.
41. Stipčević, M.; Ursin, R. An on-demand optical quantum random number generator with in-future action and ultra-fast response. *Sci. Rep.* **2015**, *5*, 10214. [[CrossRef](#)]
42. QURSA-UC3M. qursa-uc3m/ml-predictors-min-entropy-estimation. 2024. Available online: <https://github.com/qursa-uc3m/ml-predictors-min-entropy-estimation/tree/0.1.0> (accessed on 28 June 2024).
43. OpenSSL Project. RAND_bytes(3)—OpenSSL 3.4. 2024. Available online: https://www.openssl.org/docs/man3.4/man3/RAND_bytes.html (accessed on 28 June 2024).
44. Meeker, W.Q.; Hahn, G.J.; Escobar, L.A. *Statistical Intervals: A Guide for Practitioners and Researchers*; John Wiley & Sons: Hoboken, NJ, USA, 2017; Volume 541.
45. Face, H. Transformers: State-of-the-Art Machine Learning for PyTorch, TensorFlow, and JAX. Available online: <https://huggingface.co/docs/transformers/index> (accessed on 28 June 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.