entropy

MDPI

Article

Exploring Simplicity Bias in 1D Dynamical Systems

Kamal Dingle [1,2,*], Mohammad Alaskandarani [1], Boumediene Hamzi [3,4,*] and Ard A. Louis [2]

[1] Centre for Applied Mathematics and Bioinformatics, Department of Mathematics and Natural Sciences, Gulf University for Science and Technology, Hawally 32093, Kuwait; gust0020604@gust.edu.kw
[2] Rudolf Peierls Centre for Theoretical Physics, University of Oxford, Parks Road, Oxford OX1 3PU, UK; ard.louis@physics.ox.ac.uk
[3] Department of Computing and Mathematical Sciences, California Institute of Technology, Caltech, CA 91125, USA
[4] The Alan Turing Institute, London NW1 2DB, UK
* Correspondence: dingle.k@gust.edu.kw (K.D.); bhamzi@turing.ac.uk (B.H.)

Abstract: Arguments inspired by algorithmic information theory predict an inverse relation between the probability and complexity of output patterns in a wide range of input–output maps. This phenomenon is known as *simplicity bias*. By viewing the parameters of dynamical systems as inputs, and the resulting (digitised) trajectories as outputs, we study simplicity bias in the logistic map, Gauss map, sine map, Bernoulli map, and tent map. We find that the logistic map, Gauss map, and sine map all exhibit simplicity bias upon sampling of map initial values and parameter values, but the Bernoulli map and tent map do not. The simplicity bias upper bound on the output pattern probability is used to make a priori predictions regarding the probability of output patterns. In some cases, the predictions are surprisingly accurate, given that almost no details of the underlying dynamical systems are assumed. More generally, we argue that studying probability–complexity relationships may be a useful tool when studying patterns in dynamical systems.

Keywords: dynamical systems; algorithmic probability; simplicity bias; time series

Citation: Dingle, K.; Alaskandarani, M.; Hamzi, B.; Louis, A.A. Exploring Simplicity Bias in 1D Dynamical Systems. *Entropy* **2024**, *26*, 426. https://doi.org/10.3390/e26050426

Academic Editor: José F. F. Mendes

Received: 23 March 2024
Revised: 11 May 2024
Accepted: 13 May 2024
Published: 16 May 2024

1. Introduction

In recent years, several studies of *simplicity bias* have been made in input–output maps, in which a general inverse relationship between the complexity of outputs and their respective probabilities has been observed [1,2]. More specifically, using arguments inspired by *algorithmic information theory* [3–5] (AIT), and specifically *algorithmic probability*, an upper bound on the probability $P(x)$ of observing output pattern x was presented [1], with the bound depending on the estimated Kolmogorov complexity of x. The upper bound implies that complex output patterns must have low probabilities, while high-probability outputs must be simple. Example systems in which simplicity bias has been observed include RNA structures [1,6], differential equation solutions [1], finite state transducers [2], time series patterns in natural data [7], and natural protein structures [8], among others.

Which systems will and will not show simplicity bias has yet to be determined, but the phenomenon is expected to appear in a wide class of input–output maps, under fairly general conditions. Some of these conditions were suggested in ref. [1], including (1) that the number of inputs should be much larger than the number of outputs, (2) the number of outputs should be large, and (3) that the map should be 'simple' (technically of $O(1)$ complexity) to prevent the map itself from dominating over inputs in defining output patterns. Indeed, if an arbitrarily complex map was permitted, outputs could have arbitrary complexities and probabilities, and thereby remove any connection between probability and complexity. Finally (4), because many AIT applications rely on approximations of Kolmogorov complexity via standard lossless compression algorithms [9,10] (but see [11,12] for a fundamentally different approach), another condition that has been proposed is that the map should not generate pseudo-random outputs like $\pi = 3.1415\ldots$, which standard

compressors cannot handle effectively. The presence of such outputs may yield high-probability outputs which appear 'complex', apparently violating simplicity bias, but which are in fact simple.

To further explore the presence of simplicity bias in dynamical systems and physics, and also in maps which test the boundaries of the conditions for simplicity bias described above, we examine the output trajectories of a selection of 1D maps from chaos theory, namely the logistic map, the Gauss ("mouse") map, the sine map, the Bernoulli map, and the tent map. Starting with its popularisation by Robert May [13] in the 1970s, the logistic map has been heavily studied. The map is a textbook example of a simple system which can produce simple, complex, and chaotic patterns via iterations of the map [14]. This map is related to a very wide range of nonlinear models with applications in epidemiology, economics, physics, time series, etc. [15]. Due to the popularity of the logistic map, and because its trajectory outputs can depict simple as well as complex, chaotic, and even pseudo-random patterns, we focus primarily on the logistic map, but we also analyse the other mentioned maps.

Although we are not restricted to studying binary strings, most AIT results are framed in the context of binary strings and hence applications are easier in the same framework. Thus, in this work, we will study simplicity bias in digitised binary trajectories of these example 1D dynamical systems. Our main conclusions are that simplicity bias appears in the logistic map, the Gauss ("mouse") map, and the sine map, and hence we suggest that simplicity bias may also appear in natural dynamical systems more generally. As a broader context of motivation, this work contributes to research at the intersection of dynamical systems, AIT, and machine learning.

2. Background and Problem Set-Up

2.1. Background Theory and Pertinent Results

2.1.1. AIT and Kolmogorov Complexity

We will now provide some basic background information regarding AIT, and describe simplicity bias in more detail. Note that the current work will not involve detailed AIT or related theory, so we only give a brief survey of relevant results without giving many formal details here. There are many standard texts which the interested reader can refer to if needed, e.g., refs. [16–19].

Within computer science, *algorithmic information theory* [3–5] (AIT) directly connects computation, computability theory, and information theory. The central quantity of AIT is *Kolmogorov complexity*, $K(x)$, which measures the complexity of an individual object x as the amount of information required to describe or generate x. $K(x)$ is more technically defined as the length of a shortest program which runs on an (optimal prefix) *universal Turing machine* (UTM) [20], generates x, and halts. More formally, the Kolmogorov complexity $K_U(x)$ of a string x with respect to U is defined [3–5] as

$$K_U(x) = \min_p \{|p| : U(p) = x\} \tag{1}$$

where p is a binary program for a prefix optimal UTM U, and $|p|$ indicates the length of the program p in bits. Due to the invariance theorem [16] for any two optimal UTMs U and V, $K_U(x) = K_V(x) + O(1)$ so that the complexity of x is independent of the machine, up to additive constants. Hence, we conventionally drop the subscript U in $K_U(x)$, and speak of 'the' Kolmogorov complexity $K(x)$. Informally, $K(x)$ can be defined as the length of the shortest program that produces x, or simply as the size in bits of the compressed version of x (assuming we have a perfect compressor). If x contains repeating patterns like $x = 1010101010101010$ then it is easy to compress, and hence $K(x)$ will be small. On the other hand, a randomly generated bit string of length n is highly unlikely to contain any significant patterns, and hence can only be described via specifying each bit separately without any compression, so that $K(x) \approx n$ bits. Other more expressive names for $K(x)$ are *descriptional complexity*, *algorithmic complexity*, and *program-size complexity*, each

of which highlight the idea that $K(x)$ is measuring the amount of information to describe or generate x precisely and unambiguously. Note that Shannon information and Kolmogorov complexity are related [21], but differ fundamentally in that Shannon information quantifies the information or complexity of a random source, while Kolmogorov complexity quantifies the information of individual sequences or objects.

An increasing number of studies show that AIT and Kolmogorov complexity can be successfully applied in physics, including thermodynamics [22–25], quantum physics [26], and entropy estimation [27,28]. Further, they also have numerous potential applications in biology [8,29,30], other natural sciences [31], and engineering [16,32,33].

2.1.2. The Coding Theorem and Algorithmic Probability

An important result in AIT is Levin's [34] coding theorem, which establishes a fundamental connection between $K(x)$ and probability predictions. Mathematically, it states that

$$P(x) = 2^{-K(x)+O(1)} \tag{2}$$

where $P(x)$ is the probability that an output x is generated by a (prefix optimal) UTM fed with a random binary program. Thus, high complexity outputs have exponentially low probability, and simple outputs must have high probability. This is a profound result which links notions of data compression and probability in a direct way. $P(x)$ is also known as the *algorithmic probability* of x. Given the wide-reaching and striking nature of this theorem, it is somewhat surprising that it is not more widely studied in the natural sciences. The reason, in part, for this inattention is that AIT results are often difficult to apply directly in real-world contexts due to a number of issues, including the fact that $K(x)$ is formally uncomputable and also in light of the ubiquitous use of UTMs.

2.1.3. The Simplicity Bias Bound

Coding theorem-like behaviour in real-world input–output maps has been studied recently, leading to the observation of a phenomenon called *simplicity bias* [1] (see also ref. [35]). Simplicity bias is captured mathematically as

$$P(x) \leq 2^{-a\tilde{K}(x)-b} \tag{3}$$

where $P(x)$ is the (computable) probability of observing output x on random choice of inputs, and $\tilde{K}(x)$ is the approximate Kolmogorov complexity of the output x: complex outputs from input–output maps have lower probabilities, and high-probability outputs are simpler. The constants $a > 0$ and b can be fit with little sampling and often even predicted without recourse to sampling [1]. We will assume that $b = 0$ in Equation (3) throughout this work, which is a default assumption, as argued and discussed in ref. [1]. There is also a conditional version of the simplicity bias equation [36].

The ways in which simplicity bias differs from Levin's coding theorem include that it does not assume UTMs, uses approximations of complexities, and for many outputs, $P(x) \ll 2^{-K(x)}$. Hence, the abundance of low complexity, low-probability outputs [2,37] is a sign of simplicity bias.

2.1.4. Estimating Pattern Complexity

To estimate complexity, we follow the guidance of ref. [1] and use

$$C_{LZ}(x) = \begin{cases} \log_2(n), & x = 0^n \text{ or } 1^n \\ \log_2(n)[N_w(x_1 \ldots x_n) + N_w(x_n \ldots x_1)]/2, & \text{otherwise} \end{cases} \tag{4}$$

where $N_w(x)$ comes from the 1976 Lempel and Ziv complexity measure [9], and where the simplest strings 0^n and 1^n are separated because $N_w(x)$ assigns complexity $K = 1$ to the string 0 or 1, but complexity 2 to 0^n or 1^n for $n \geq 2$, whereas the true Kolmogorov complexity of such a trivial string actually scales as $\log_2(n)$ for typical n, because one only

needs to encode n. Having said that, the minimum possible value is $K(x) \approx 0$ for a simple set, and so, e.g., for binary strings of length n, we can expect $0 \leq K(x) \leq n$ bits. Because for a random string of length n the value $C_{LZ}(x)$ is often much larger than n, especially for short strings, we scale the complexity so that a in Equation (3) is set to $a = 1$ via

$$\tilde{K}(x) = \log_2(M) \cdot \frac{C_{LZ}(x) - \min_x(C_{LZ})}{\max_x(C_{LZ}) - \min_x(C_{LZ})} \tag{5}$$

where M is the maximum possible number of output patterns in the system, and the min and max complexities are over all strings x which the map can generate. $\tilde{K}(x)$ is the approximation to Kolmogorov complexity that we use throughout. This scaling results in $0 \leq \tilde{K}(x) \leq n$ which is the desirable range of values.

2.2. Digitised Map Trajectories

The AIT coding theorem is framed in terms of random inputs or 'programs' and the resultant output strings or patterns. The core idea of the simplicity bias bound is that for a large range of systems, uniformly sampling input parameter values ('programs') yields an exponential variation in output pattern probabilities, with high-probability patterns having low complexities. While dynamical systems may not appear to fit this input–output framework, they can be viewed as input–output functions if the initial values and parameters are considered input 'programs' which are then computed to generate 'output' dynamical system trajectories. Because output pattern complexities and probabilities are more easily calculated if outputs are binary strings (and as above, AIT is mainly framed in terms of binary strings), we will digitise the real-valued trajectories into binary strings.

To illustrate this input–output framework, consider binary sequence trajectories resulting from digitised realisations of the logistic map

$$x_{k+1} = \mu x_k(1 - x_k) \tag{6}$$

where the inputs are the pair of values $x_0 \in (0.0, 1.0)$ and $\mu \in (0.0, 4.0]$. For the outputs, the map is first iterated to obtain a sequence of n real values $x_1, x_2, x_3, \ldots, x_n$ in $[0, 1]$. Similar to the field of symbolic dynamics [38], the real-valued trajectory is digitised to become a binary string output sequence x by applying a threshold, and writing 1 if $x_k \geq 0.5$ and 0 otherwise [39]. Hence, a binary sequence output x of n bits is generated for each input pair μ and x_0.

By way of example, consider choosing $\mu = 3.8$ and $x_0 = 0.1$ with $n = 25$, then from iterating Equation (6) with $k = 1, 2, \ldots, 25$ we obtain the real-valued trajectory

$$x_1, x_2, \ldots, x_{25} = 0.34, 0.86, \ldots, 0.25, 0.72, 0.77, 0.67$$

which, after digitisation, becomes the binary string

$$x = 0101011011111011010110111$$

Figure 1 illustrates the trajectory and digitisation procedure. The choice of n strikes a balance, as it is long enough that many different patterns can emerge, and short enough that decent frequency and probability estimates can be made without the need for excessive sampling. Different μ and x_0 inputs can yield different binary strings $x \in \{0, 1\}^n$.

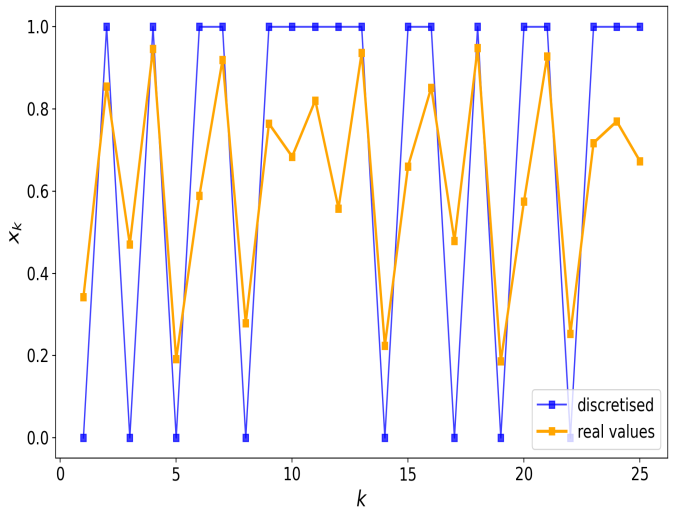

Figure 1. An example of a real-valued (orange) and digitised (blue) trajectory of the logistic map, with $\mu = 3.8$ and $x_0 = 0.1$. The discretisation is defined by writing 1 if $x_k \geq 0.5$ and 0 otherwise, resulting in the pattern $x = 010101101111011010110111$, which has a length of $n = 25$ bits.

Note that throughout the paper, we will ignore the first 50 iterations of a map so as to discard the transient dynamics of the iterated maps. This step of ignoring the initial iterations is common practice [40].

3. Results

3.1. Logistic Map

The logistic map in Equation (6) is possibly the best known and most iconic 1D map studied in dynamical systems and chaos theory. To our knowledge, there have not been many studies investigating the complexity of digitised trajectories of the logistic map. A notable exception is the work by Kaspar and Schuster [41], who also studied the estimated complexity of resulting binary string trajectories. However, their work was fundamentally different in that it was not concerned with simplicity bias or estimating the probability of different outputs.

3.1.1. Parameter Intervals

The intervals $x_0 \in (0.0, 1.0)$ and $\mu \in (0.0, 4.0]$ are standard ranges in which the logistic map is studied and it can be shown [40], for example, that for $\mu > 4.0$, very few trajectories are confined to $[0, 1]$; similarly, if $x_0 \notin (0, 1)$, the behaviour can be unbounded or trivial. For some large values of μ (e.g., $\mu = 4.0$), almost all initial values x_0 yield complex and chaotic outputs [14], and the distribution over digitised trajectories is roughly uniform, with little bias. Note that we use the word 'bias' to describe a strongly non-uniform distribution of the probability to obtain a given binary string output.

In Figure 2a, the bifurcation diagram of the logistic map is shown, in which for different values of μ the asymptotic x_k values are depicted. The diagram shows fixed points, oscillations, and non-period behaviour.

The value 0.5 is also added as a red line, highlighting the digitising threshold. It is known [14] that if $\mu \in [0.0, 1.0]$, then the trajectories tend toward 0. Because we truncate at 0.5, the corresponding binary string would be $x = \ldots 0000$. If $\mu \in (1.0, 2.0]$, then the trajectories tend toward $1 - (1/\mu)$, which means that we expect to see binary strings $x = \ldots 0000$. For $\mu \in (2.0, 3.0]$, the resulting pattern would be $x = \ldots 1111$ because $1 - (1/\mu)$ is greater than 0.5. For μ from 3.0 to $\approx$3.3, we still expect $x = \ldots 1111$ because although the first bifurcation appears at $\mu = 3.0$, and oscillations with period two begin, until about 3.3, both values of the oscillations are larger than 0.5. Figure 2b shows the same bifurcation diagram, but zooms in on the larger values of μ. From about 3.3 to about 3.5,

oscillations between two and four values appear, and larger values of μ can yield oscillation periods of 8, 16, etc., which will yield patterns such as $x = 0101\ldots$. Chaotic trajectories do not occur until [14] $\mu \geq 3.56994567\ldots \approx 3.57$ (https://oeis.org/A098587 (accessed on 1 September 2023)), and so the μ interval from 3.0 to about 3.5699 contains period-doubling bifurcations, in which oscillations of exponentially increasing frequency occur, but there are no truly chaotic trajectories. Finally, for μ between ≈3.57 and 4.0, more complex and chaotic patterns can emerge, but also 'islands' of stability appear in this interval, as can be observed in Figure 2b.

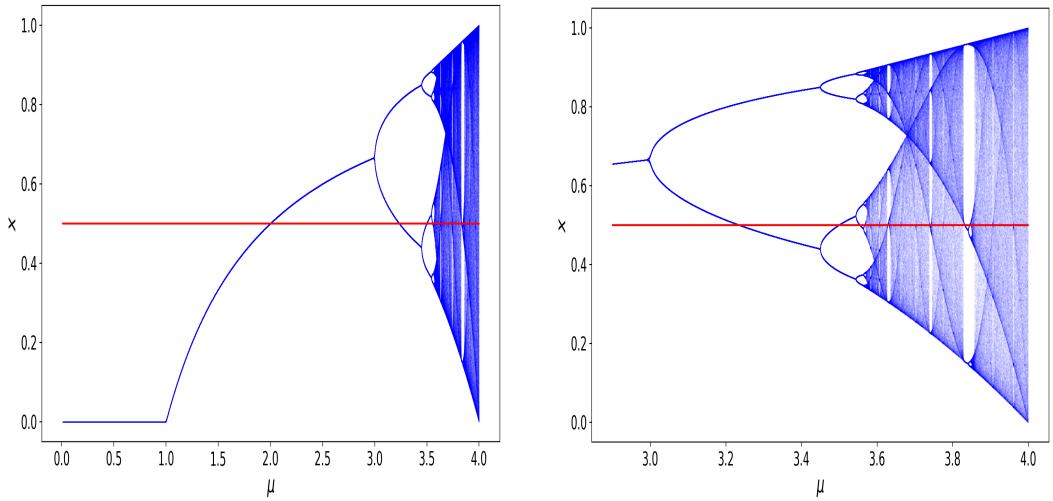

Figure 2. A bifurcation diagram for the logistic map. In (**a**), we see the diagram for parameters $\mu \in (0, 4.0]$; and in (**b**), we see the diagram for values $\mu \in (2.9, 4.0]$. The value 0.5 has been highlighted in red, to indicate the cut-off threshold used to digitise trajectories by a value of 0 if the output is below the threshold, and a value of 1 if it is greater than or equal to the threshold.

In our numerical simulations, we will separately investigate various intervals for μ, namely μ sampled uniformly from (0.0, 4.0], [3.0, 4.0], [3.57, 4.0], and finally also fixing $\mu = 4.0$. The motivation for choosing these intervals is as follows: given that the interval $\mu \in (0.0, 4.0]$ is the standard range in which the map is studied, the most 'natural' sampling strategy is to sample uniformly across this interval. As for choosing intervals [3.0, 4.0] and [3.57, 4.0], these are interesting test cases because we can expect to see complex patterns appearing more frequently at these intervals. Finally, when fixing $\mu = 4.0$, most trajectories will be highly complex, and so we might expect simplicity bias to disappear.

3.1.2. Connection of Simplicity and Probability

Reflecting on the bifurcation diagram and the above-related comments, it is easily seen by uniformly sampling μ from (0.0, 4.0] that some simple binary patterns will have high probability, and highly complex patterns will have low probability, just because most of the interval (0.0, 4.0] yields fixed points or low-period oscillations. Hence, some general form of bias towards simplicity is expected for the logistic map. However, what is not a priori obvious is whether or to what extend the simplicity bias bound in Equation (3) will be followed or will have predictive value.

3.1.3. Simplicity Bias Appears When Bias Appears

Following the protocol for generating binary strings via logistic map trajectories described above, we now examine the probability $P(x)$ that the digitised trajectory x is produced on random sampling of input parameters μ and x_0, which is performed here using 10^6 samples over uniformly chosen random parameters. Using Equation (3), we can make an a priori prediction regarding the upper-bound decay upon sampling of the input

parameters. (Recall that we ignore the first 50 iterations of the map in order to exclude the transient dynamics).

Figure 3a shows that the upper-bound prediction (the black line) agrees remarkably well with the probability and complexity data for different binary string output (blue dots); we see that the logistic map displays simplicity bias when uniform sampling $\mu \in (0.0, 4.0]$, such that high-probability outputs have low complexities, and complex outputs have low probability. The gradient prediction ($a = 1$) of the black line used no information about the map, except that we assumed that almost all 2^n outputs of length n bits can be realised. An upper-bound fit to the $\log_{10} P(x)$ data gives the slope as -0.18. Note that many output strings fall below the upper-bound prediction, as we expected based on earlier studies [1,37], but nonetheless it is known that randomly generated outputs tend to be close to the bound [2]. Put differently, even though many output strings (blue dots) appear to be far below the bound, most of the probability mass for each complexity value is concentrated close to the bound. Thus, this simple bound predicts $P(x)$ quite accurately by using the complexity values of output strings while otherwise completely ignoring the details of the underlying dynamical system.

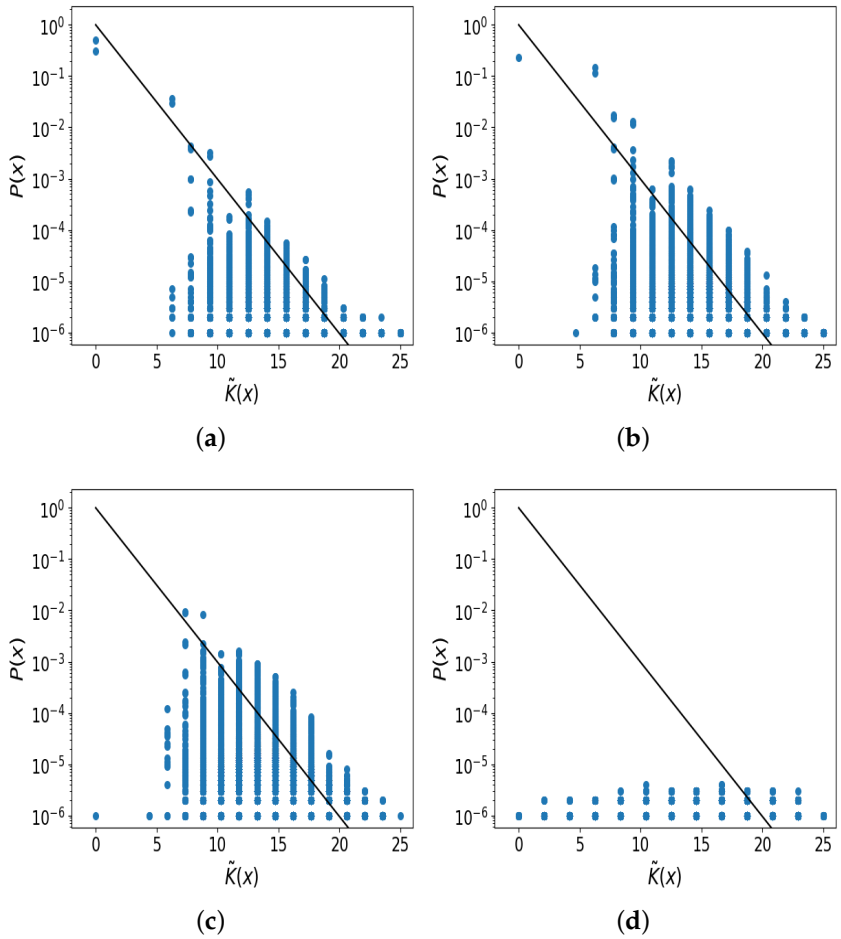

Figure 3. Simplicity bias in the digitised logistic map from random samples with $x_0 \in (0, 1)$ and μ sampled in different intervals. Each blue data-point corresponds to a different binary digitised trajectory x of 25 bits in length. The black line is the upper-bound prediction of Equation (3). (**a**) Clear simplicity bias for $\mu \in (0.0, 4.0]$ with $P(x)$ closely following the upper bound, except for low frequency and high complexity outputs which suffer from increased sampling noise; (**b**) simplicity bias is still present for $\mu \in [3.0, 4.0]$; (**c**) the distribution of $P(x)$ becomes more flat (less biased) and simplicity bias is much less clear when $\mu \in [3.57, 4.0]$ due to constraining the sampling to μ-regions more likely to show chaos; (**d**) the distribution of $P(x)$ is roughly uniform when using $\mu = 4.0$, with almost no bias, and hence no possibility of simplicity bias.

In Figure 3b we make a similar plot to panel (a) except that we restrict the sampling to $\mu \in [3.0, 4.0]$. The qualitative pattern in (b) is similar to (a), although the simplicity bias is slightly less clear than in (a). An upper-bound fit to the $\log_{10} P(x)$ data gives the slope as -0.17. For Figure 3c, the sampling interval was chosen so that $\mu \in [3.57, 4.0]$ which is the region containing some truly chaotic trajectories [40]. Both bias and simplicity bias are less clear in this plot. An upper-bound fit to the $\log_{10} P(x)$ data gives a slope of -0.04. In Figure 3d a plot is given in which μ is no longer sampled, but is fixed at $\mu = 4.0$, so that almost all x_0 values yield chaotic trajectories. As expected, there is very little bias in the distribution, i.e., the distribution of $P(x)$ is roughly uniform, and hence no simplicity bias can be observed. An upper-bound fit to the $\log_{10} P(x)$ data gives a slope of 0.02. Figure 3d still shows some very simple strings, which can be understood based on the fact that $x_i \approx 0 \Rightarrow x_{i+j} \approx 0$ for $j \approx 1$ even if the trajectories may become much more complex for $j \gg 1$. In other words, if the trajectory reaches a value close to zero, then it tends to stay close to zero for several iterations. These initial short trajectories with many 0 s will have very low complexity.

As argued above, the fact that simple patterns occur with high probability when sampling μ from (0.0, 4.0] is not in itself remarkable, and can be rationalised based on the bifurcation diagram. However, what is noteworthy in our investigation is that we do not just see a vague general inverse relation between complexity and probability (as the preceding discussion might predict), but rather that we see an exponential decay in probability with linearly increasing complexity, which follows the upper bound of Equation (3) quite closely.

See the Appendix A for an illustration of the effects of using the different values of n. See Appendix B for the same plots as in Figure 3, but with semi-transparent data points to highlight the distribution of data points.

3.1.4. Distribution of Complexities

By itself, simplicity bias does not necessarily imply that the distribution of complexity values, $P(K(x) = r)$, will be skewed towards low complexity values. Rather, simplicity bias means that the individual probability of some simple strings will be higher than that of more complex strings. Because the probability of observing a given complexity r depends on the product of the number of strings of complexity r and their individual probabilities, the distribution $P(K(x) = r)$ may in fact peak at high complexities, or low complexities, or have no peak at all possibly.

To investigate this distribution in the logistic map, in Figure 4, we plot the distribution of complexities for the different sampling intervals of μ. As can be seen in Figure 4a, when sampling from [0.0, 4.0] there is a bias towards lower complexities. As for (b), when μ is sampled from [3.0, 4.0] the distribution of complexities is roughly uniform (at least on a log scale). In Figure 4c, the distribution is also somewhat uniform but peaks around medium complexity values. In (d), there is a bias towards higher complexity values. For comparison, in (e) we also plot the distribution of complexities resulting from sampling purely random binary strings, and this distribution is very similar to that in (d), which is obtained when $\mu = 4.0$. It is noteworthy that in some of these cases, while there is some evidence of bias toward higher or lower complexities, the distributions still display spread and are not narrowly peaked (at least on a log scale). We note that if the logistic map produced binary output strings with a uniform probability over strings, in theory, the frequency would grow exponentially with complexity r. However, for Lempel–Ziv complexity with short strings, the distribution is not quite exponential; see [1,42].

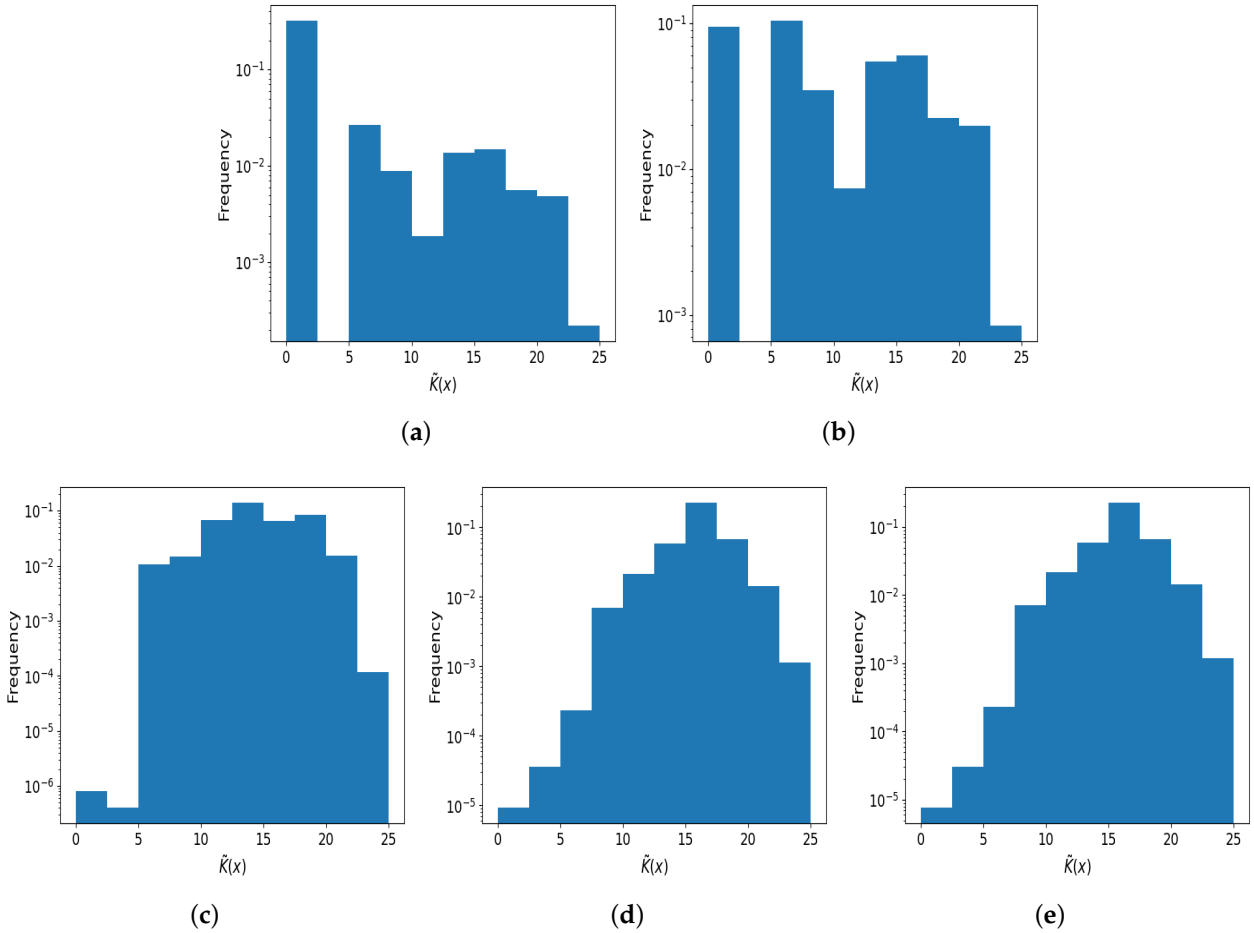

Figure 4. The distribution $P(\tilde{K}(x) = r)$ of output complexity values, with $x_0 \in (0.0, 1.0)$ and μ sampled from different intervals. (**a**) A roughly uniform complexity distribution for $\mu \in (0.0, 4.0]$, with some bias towards lower complexities (mean is 3.4 bits); (**b**) close to uniform distribution of complexities for $\mu \in [3.0, 4.0]$, mean is 10.3 bits; (**c**) the distribution leans toward higher complexities when $\mu \in [3.57, 4.0]$, mean is 14.1 bits; (**d**) the distribution is biased to higher complexity values when $\mu = 4.0$ (mean is 16.4 bits); (**e**) for comparison, purely random binary strings of 25 bits in length were generated (mean is 16.2 bits). The distributions of complexity values in (**d**,**e**) are very similar, but (**a**–**c**) show distinct differences. Calculating and comparing $P(K)$ is an efficient way of checking how simplicity-biased a map is.

The appearance of a distribution that is much less peaked towards high complexity than that of randomly chosen strings can be rationalised from AIT arguments. To the first order, there are $\sim 2^r$ strings of complexity r, each with probability $\sim 2^{-r}$, such that the product $2^r 2^{-r} = O(1)$ is independent of r, and is presumably uniform. For prefix complexity, the number of strings with complexity r is slightly less than 2^r, and hence for a prefix UTM, the distribution would be biased to lower r values to some extent. Nonetheless, the brief rough argument outlined is still valid as a first-order approximation for distributions with large spread, as we see in Figure 4. See ref. [8] for similar arguments and results regarding the distribution of complexities in a biological setting, as well as ref. [42] for the machine learning settings.

Finally, we note that in practice, far fewer samples are needed to produce a $P(\tilde{K}(x) = r)$ distribution than a $P(x)$ distribution, because many strings have the same $\tilde{K}$. Comparing the sampled distribution to a null model of random strings may be the quickest and easiest way to diagnose simplicity bias in a dynamical system.

3.1.5. Complex and Pseudo-Random Outputs

The occurrence of complex patterns may prompt a question: since the logistic map in Equation (6) is itself simple with low Kolmogorov complexity, it might be supposed that any output x it generates should be simple as well. So, how can we obtain large variations in complexity which would lead to large variations in probability? Or are all the apparently complex outputs merely pseudo-random patterns? We can address these analytically via bounding the complexity $K(x)$ of a discretised trajectory written as an n-bit binary string x using the following inequality

$$K(x) \leq K(\text{logistic map}) + K(\mu, x_0) + K(n) + O(1) \tag{7}$$

This bound follows from the fact that any x can be described precisely by first describing: (a) the logistic map function in Equation (6) with only a few bits because $K(\text{logistic map}) = O(1)$ bits, (b) a pair of values μ and x_0 which yield x with $K(\mu, x_0)$ bits, and (c) the length of the string (i.e., the number of iterations to perform) via n, which is at most $\log_2(n)$ bits (up to loglog terms). From this upper bound, we see that if μ and x_0 are simple values with short descriptions, then $K(x) \ll n$, so x must be simple. However, because μ and x_0 are (truncated) real numbers randomly chosen from their intervals, they need not be simple—rather the opposite—because almost all decimal numbers sampled here are given to d decimal places, and so will have high complexity values of $\sim \log_2(10)d$ bits. Therefore, outputs need not be simple just because the map is simple, due to the presence of the complexity of the inputs in Equation (7). Nor are all outputs merely pseudo-random patterns. Indeed, this argument concurs with the fact that in Figure 3, we see that very many outputs x are realised, most of which must be (truly) complex because it is well known from AIT that almost all strings of length n are complex and only relatively few are simple.

Extending the preceding discussion on Equation (7), it is also interesting to ask if there are simple input parameters which generate high-entropy pseudo-random outputs. Indeed there are; for example, if $\mu = 4.0$ then almost all x_0 lead to chaotic trajectories, including some simple initial values like $x_0 = 1/3$, which can be described with only a few bits, so that $K(\mu, x_0) = O(1)$ and so $K(x) \leq \log_2(n) + O(1)$ (up to additive loglog terms), but at the same time $\tilde{K}(x) \approx n$ bits because the Lempel–Ziv complexity measure cannot compress pseudo-random strings.

On reflection, these pseudo-random strings with $K(x) \ll \tilde{K}(x)$ must be 'rare' in the space, and do not have individually high probability, otherwise we would not see simplicity bias, or at least we would see many points that strongly violate the upper bound in Figure 3a, which we do not.

3.1.6. Pre-Chaotic Regime

As discussed above, when $\mu > 3.5699\ldots$ some trajectories can be chaotic, but there are also intermittent 'islands' of stability, and in general the pattern of types of behaviour in the region 3.57 to 4.0 is intricate and complicated. However, in the region $\mu \in [0.0, 3.5699]$, the types of behaviour are more straightforward, with progressive period-doubling bifurcations but no true chaos. Feigenbaum [43,44] famously showed that the distance between successive bifurcations eventually decreases exponentially, as $\mu_{q+1} - \mu_q \approx 2.069/4.669^q$ [40], where μ_q is the value of the q^{th} bifurcation.

Because this interval $\mu \in [0.0, 3.5699]$ is relatively easy to understand, and it is similarly easy to determine if simplicity bias appears even without chaos, we also generated a complexity–probability plot for $\mu \in [0.0, 3.5699]$, which was uniformly sampled: Figure 5a shows the decay in probability with increasing complexity. As is apparent, there are fewer patterns (blue dots), which is because we are restricting the possible space of patterns by restricting to this interval of μ. We see that some simplicity bias is observed, but the bound is less clearly followed as compared to that observed when sampling across [0.0, 4.0]. For example, the upper bound on the data points' decay is less clearly linear.

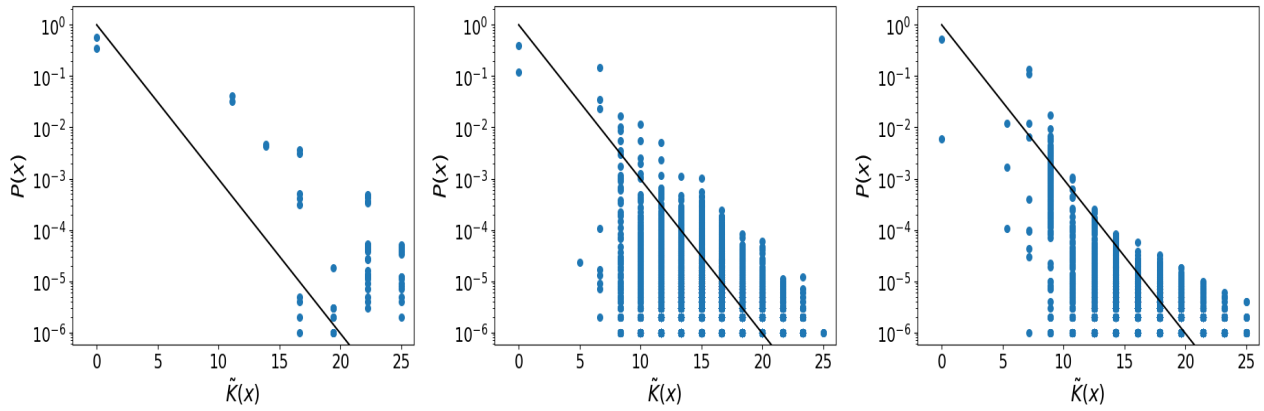

Figure 5. Simplicity bias in (**a**) the logistic map with μ sampled in [0.0, 3.5699], which is the non-chaotic period doubling regime (upper bound fitted slope is -0.17); (**b**) the Gauss map (upper bound fitted slope is -0.13); and (**c**) the sine map (upper bound fitted slope is -0.17).

To obtain a better understanding of the complexity of trajectories, we can find (WolframAlpha (https://www.wolframalpha.com/ (accessed on 1 September 2023)), using command "logistic map for r = [//number:3.4//] and x = [//number:0.2//]") the oscillation periods for some chosen values of μ: With $\mu = 3.4$, period 2; $\mu = 3.5$, period 4; $\mu = 3.56$, period 8; $\mu = 3.569$, period 32; $\mu = 3.5699$, period 128. So for the sampled interval [0.0, 3.5699] the highest period is 128. Because we use $n = 25$, any pattern with a period of 32 or more will not appear as periodic, because n needs to be large for the periodicity to become plain. From this brief analysis, we can see how the low-probability and high-complexity patterns have appeared in Figure 5a. In conclusion, some simplicity bias is observed for the interval $\mu \in [0.0, 3.5699]$, but it is not as pronounced as that for sampling $\mu \in [0.0, 4.0]$, which is presumably due to the presence of potentially more complex patterns in the interval 3.57 to 4.0.

3.2. Gauss Map ("Mouse Map")

Moving on from the logistic map, we now explore simplicity bias in another 1D map, namely the *Gauss map*, which is also known as the *mouse map* because the bifurcation diagram is reminiscent of a mouse [45]. The equation defining the dynamical system is

$$x_{k+1} = e^{-\alpha x_k^2} + \beta \tag{8}$$

The value $\alpha = 7.5$ is chosen and will be fixed. For many values of α, the bifurcation diagram is not sufficiently complex as to yield many varied trajectories; also the value 7.5 has been used previously [46]. The value β is the bifurcation parameter and the map is typically studied [45,47] with $\beta \in [-1.0, 1.0]$.

Similar to the logistic map example, we will sample initial values x_0 and values of β, then ignore the first 50 iterations (due to transient dynamics), and then digitise the real-valued trajectory to form binary string outputs. Because iterations of Equation (8) are not confined to [0.0, 1.0] and due to the form of the bifurcation diagram [45], we will sample $x_0 \in [-0.5, 0.5]$ uniformly. Also, due to the form of the bifurcation diagram, the digitisation threshold will be set at 0.2 (instead of 0.5, as it was in the case of the logistic map). Changing the threshold is implemented merely to avoid having too many trivial outputs of $x = 000\ldots000$. As stated above, we will use $n = 25$.

In Figure 5b, we see that there is clear simplicity bias in the Gauss map. The slope of the decay follows the predicted black line very closely, but the offset value b is not well approximated by $b = 0$. Nonetheless, there is clear simplicity bias, which is similar to the logistic map case. Hence, we can predict the relative log-probability of output strings with quite high accuracy.

3.3. Sine Map

We now study simplicity bias in the *sine map*. Wolfram [48] described the sine map

$$x_{k+1} = \mu \sin(\pi\sqrt{x_k}) \tag{9}$$

and further displayed the bifurcation diagram for this map, which is broadly similar to that of the logistic map, to illustrate Feigenbaum's discovery of universality. For this map, we sample $x_0 \in [0.0, 1.0]$ uniformly 10^6 times and return to the digitisation threshold of 0.5. The parameter μ will be sampled uniformly from [0.0, 1.0]. As stated before, the first 50 iterates will be ignored, and we will use $n = 25$. Note that there is another form of the sine map which is sometimes studied in dynamical systems [49,50], in which there is no square root on x_k.

Figure 5c shows the complexity–probability plot for the sine map, and simplicity bias is also present here. However, the upper bound is less clearly linear (on the log scale), and the black line prediction is not as closely followed. It is also worth noting that in the tail of the decay at higher complexities, the upper bound appears to curve upward slightly. It is not clear why this happens.

3.4. Bernoulli Map

Moving on to another prototypical chaotic map, the Bernoulli map (also known as the *dyadic map*, *bit shift map*, *doubling map*, or *sawtooth map*) is defined via the equation

$$x_{k+1} = (2x_k \mod 1) \tag{10}$$

with $x_0 \in (0.0, 1.0)$. This map shows sensitive dependence on initial conditions because the trajectories of two inputs x_0 and y_0 which differ by $|x_0 - y_0| \approx 0$ will eventually diverge for a large enough k.

Given that the Bernoulli map is also a 1D chaotic system with a simple ($O(1)$ complexity) map, it is interesting to ask whether it shows simplicity bias like the logistic map and others do. A little reflection shows that this map does not show simplicity bias nor any other type of bias, even for a digitised version. The trajectory is defined by multiplying 2 by x_0, ignoring the integer part, so if x_0 is a random real number then the trajectory will be random and incompressible because $(2x_0 \mod 1)$ will be another random number, assuming that x_0 is defined to a large number of decimal places. Multiplying a random number by two does not remove the randomness. Hence, the binary discretised trajectory sequence would look almost random, with small and quickly decaying autocorrelation (see Section 9.4 of ref. [17] for a similar conclusion). For bias and simplicity bias, it is necessary for random inputs to be able to lose a lot of their information and complexity (i.e., complex inputs must be able to produce simple outputs), but the Bernoulli map does not allow this. Hence, the behaviour of this map is similar to that of the logistic map with $\mu = 4.0$ in the sense that there is no bias and no simplicity bias. Indeed, this similarity is quite natural due to the conjugacy between the logistic map with $\mu = 4.0$ and the Bernoulli (doubling) map. This map does not have a bifurcation parameter μ.

3.5. Tent Map

The last map we look at is the *tent map*, which is quite well known and studied in dynamical systems research [14]. The iterated values follow the function

$$x_{k+1} = \begin{cases} 2x_k & 0 \leq x_k \leq 0.5 \\ 2 - 2x_k & 0.5 < x_k \leq 1.0 \end{cases}$$

with $x_k \in [0.0, 1.0]$, for $k = 0, 1, 2, 3, \ldots$. This map does not have a bifurcation parameter μ. Despite being a 1D dynamical system, this map does not lead to strong bias in the distribution of digitised binary string outputs x, and hence it cannot possibly show simplicity bias. Intuitively, this can be seen due to the fact that almost all values of x_0 will

yield complex paths, while simplicity bias typically arises when most inputs lead to relatively simple paths. Indeed, because the tent map is topologically conjugate to the logistic map [14] when $\mu = 4.0$, and we saw neither bias nor simplicity bias in the logistic map [14] when $\mu = 4.0$, this helps us to understand the absence of simplicity bias in the tent map.

4. Discussion

Arguments inspired by algorithmic information theory (AIT) predict that in many input–output maps, strongly non-uniform probability distributions over outputs will occur, with complex patterns having exponentially low probabilities, and some simple patterns having high probability; this phenomenon is known as *simplicity bias* [1]. Here, we numerically investigated the presence of simplicity bias in digitised trajectories arising from iterations of the logistic map, Gauss map, sine map, Bernoulli map, and tent map. By digitising the real-valued trajectories, we studied the probability and complexity of the resulting binary strings. Our main conclusions are that (i) we observe simplicity bias in the logistic map, Gauss map, and sine map, and also that in some cases, the probability of resulting binary strings can be predicted a priori with surprising accuracy; and (ii) we do not observe simplicity bias in the trajectories of the Bernoulli map and tent map, nor indeed any bias at all.

Due to the qualitatively different behaviours exhibited by the logistic map for different μ values, we separately studied different regimes by sampling μ from (0.0, 4.0], (3.0, 4.0], (3.57, 4.0] and also $\mu = 4.0$. In general, the simplicity bias and upper bound prediction accuracy was higher for μ sampled across the full range (0.0, 4.0] and decreased for smaller ranges, and completely disappeared for $\mu = 4.0$. The logistic map is perhaps the most iconic example of a dynamical system in chaos theory, and has been very extensively studied for decades. Here, we report a novel finding relating to this map, and one that is not (merely) a subtle higher-order effect, but rather a strong effect related to order-of-magnitude variations in pattern probability. This finding is also interesting, given that we did not necessarily expect to observe simplicity bias when outputs can be pseudo-random (Cf. ref. [1]). It appears that in this map, pseudo-random outputs are sufficiently rare that they do not cause strong violations of the simplicity bias upper bound. Additionally, we found simplicity bias can be 'tuned' via altering the μ interval: sampling from the full interval (0.0, 4.0] yields a biased (low-entropy) distribution over output strings along with simplicity bias, while sampling from higher values of $\mu \approx 4.0$ yields low-bias (high-entropy) distributions and little or no simplicity bias. While we observe simplicity bias similar to that predicted by AIT arguments in some of these maps, we want to clarify that we do not think that we have in any way proven that these patterns are in fact directly linked to the AIT arguments. To achieve that outcome, much more work is needed. Nevertheless, we argue that studying probability–complexity relationships, and looking for patterns such as simplicity bias, may be a fruitful perspective for dynamical systems research.

A weakness of our probability predictions is that they only constitute an upper bound on the probabilities, and, for example, Figure 3a shows that many output trajectory patterns x fall far below their respective upper bounds. Following the hypothesis from [2,37], these low-complexity, low-probability outputs are presumably patterns which the logistic map finds 'hard' to make, yet are not intrinsically very complex. Further, the presence of these low-complexity, low-probability patterns may indicate the non-universal power of the map [2]. A potential avenue for future work would be studying what types of patterns may occur far from the bound, and possible approaches to improving on their probability predictions [37].

The motivations for this work were to explore new examples of simplicity bias in physics and specifically dynamical systems, test the boundaries of relevance of simplicity bias, and explore how information theory and algorithmic probability can inform dynamical systems research. By extension, this work expands on the project of investigating the interaction between machine learning and dynamical systems because machine learning is intimately connected to information theory and algorithmics. The broader context of

our work is a research project that explores the interface of dynamical systems, machine learning, and AIT. Given the strong interest in applying machine learning to the analysis of dynamical systems, an open and important line of questioning, then, is the extent to which machine learning methods are applicable in dynamical systems problems, and what kinds of limitations or advantages this relatively novel approach may have. Since information theory and machine learning are inextricably linked and have even been referred to as "two sides of the same coin" [51], one perspective on these questions is to consider whether computation and information processing—fundamental components of machine learning—themselves might have limits of applicability, or in some other way constrain or inform dynamical behaviours. Some fascinating examples of such limits have been identified in relation to uncomputability of trajectories, meaning that some properties of dynamical trajectories may not be possible to calculate, even in principle. In a seminal paper, Moore [52] proved that the motion of a single particle in a potential can (in some specific settings) be sufficiently complex as to simulate a Turing machine, and thereby may yield trajectories with uncomputable quantitative properties. More recently, Watson et al. [53] proved for an example many-body system that even if the exact initial values of all system parameters were determined, the renormalisaton group trajectory and resultant fixed point is impossible to predict. This kind of unpredictability is stronger than the unpredictability of (merely) chaotic dynamics, in which the limiting factor is the accuracy with which initial conditions can be measured. See the works of Wolfram [54,55], Svozil [56], Lloyd [57], and Aguirre et al. [58] for more discussion of (un)computability and (un)predictability in physical systems. Naturally, if the dynamics of a particular system cannot be computed even in principle, the accuracy of machine learning approaches to prediction in these settings will be restricted (but see [59] for further discussion on this).

There may also be deep connections between deep neural networks (DNNs) and simplicity bias. Indeed, upon random sampling of parameters, DNNs exhibit an exponential bias towards functions with low complexity [60–62]. This property implies that they can learn simple Kolmogorov data fairly well, but will not generalise well on complex data. Interestingly, by changing the initialisation over parameters towards a more artefactual chaotic regime of DNNs, this simplicity bias becomes weaker [63]. It has been recently shown that in this regime, DNNs no longer generalise well on both simple and complex data [42] and tend to overfit. This is not unexpected, because DNNs are highly expressive, and classical bias-variance arguments suggest that they should be highly prone to overfitting. The big question is why standard DNNs do not fall prey to this problem. It was argued that the Occam's razor like *simplicity bias* toward simple functions observed in standard DNNs compensates the exponential growth of the number of possible functions with complexity, and that this compensation explains why such DNNs can be highly expressive without overfitting [42]. These principles imply that if a dynamical system exhibits some form of simplicity bias then the inbuilt Occam's razor inductive bias of DNNs should make it much easier to learn by DNNs than in the opposite case where the dynamical system does not have simplicity bias.

The word "complexity" can take on many meanings [64,65], and can be vague. In this work, we are precise about what we mean by complexity, which is Kolmogorov complexity, and in practice using lossless compression methods which is a standard and theoretically motivated approximation to the true uncomputable quantity. Bialek et al. [66,67] discuss complexity in relation to time series and argue that Kolmogorov complexity lacks in its intuitive appeal for a measure of the complexity of these types of sequential patterns. Many would agree that a series with a rich complicated structure and long-range correlations is truly complex, whereas a random string of bits is merely an irregular and, in a sense, trivial pattern. In contrast, Kolmogorov complexity assigns the highest complexity to such random strings, precisely because they do not contain any correlations or structure. Having noted this, the discussion does not directly bear upon our work, because we are not studying 'complexity' in a general sense or trying to argue for one or other metric. Rather, we are studying specifically simplicity bias and AIT inspired arguments as a mathematical

framework for understanding probability and dynamical systems. AIT allows one to make quantitative bounds and predictions about various systems, as we illustrate here, regardless of whether or not the term "complexity" is being used in it truest or most correct sense.

Mathematicians and physicists are fascinated by studying simple systems which can exhibit complex patterns, even while they are not technically random. In this context, Wolfram [55] investigated simple automata, and showed that some have high levels of computing power, leading to his conjecture that many natural systems can be Turing complete, i.e., that they can implement arbitrary algorithms, and hence produce complex patterns. Despite this focus on complexity, we observe in our work that complexity is, in a sense, not actually that common. Even though the logistic map is famous precisely due to its ability to produce complex behaviour, within the space of possible parameters, i.e., $\mu, x_0 \in \mathcal{R}$ and even restricting to $\mu \in (0.0, 4.0]$ and $x_0 \in (0.0, 1.0)$, chaos and 'complexity' only occur rarely (assuming uniform sampling of the parameter space). This observation is in agreement with with the fact that while Wolfram highlighted some rule sets that produce randomness, most of his automata rules sets do not produce complex pseudo-random patterns [55]. Coe et al. [68] analytically studied the question of when automata produce 'random' and complex dynamics, and also found that relatively few are random.

The study of random dynamical systems—in which dynamical systems are randomly perturbed in some way—is quite well established [69], including specifically dynamical systems with additive noise [70]. While deterministic dynamical systems have been studied due to their relevance to modelling problems in, e.g., ecology, physics, and economics, randomly perturbed dynamics are also common in science and hence are important to study, and arise naturally when modelling physical systems (e.g., [71]). This motivates studying simplicity bias in such random dynamical systems in future work. As a separate motivation, in the deterministic logistic map, many sampled μ values yield trajectories which quickly converge to fixed points or other fairly trivial patterns. However, by introducing some small random noise, these trajectories may be prevented from converging into trivial patterns, and therefore may show simplicity bias even while the deterministic counterpart does not. The relation of simplicity bias to random dynamical systems has been studied recently for the first time [72], but many questions on this topic remain unanswered.

In this work, we have used 1D maps—including the logistic map—as toy systems to explore simplicity bias in dynamical systems. The connection between AIT and dynamical systems has received some earlier attention from an analytical perspective [73–75], and computational perspective [72,76]. In future work it may be fruitful to investigate which, if any, properties of the logistic (or other chaotic) map can be predicted or bounded using the simplicity bias bound. That is, if we assume that the simplicity bias bound holds, what other dynamical properties of the map might follow from this? Even if these properties have already been determined, it would still be insightful to see if they could be derived (perhaps more simply) from AIT arguments. Further, the presence of simplicity bias in some of the 1D maps may lead to the simplicity bias bound being used as a trajectory prediction method. This is related to the a priori prediction of natural time series patterns attempted by Dingle et al. [7]. Another angle would be to study non-digitised trajectories of dynamical systems, which would require different complexities measures amenable to continuous curves, such as proposed in ref. [77]. More generally, exploring the use of arguments inspired by AIT in dynamical systems and chaos research is an attractive and potentially fruitful research avenue.

Author Contributions: Conceptualization, K.D., B.H. and A.A.L.; Methodology, K.D. and A.A.L.; Formal analysis, M.A.; Writing—original draft, K.D.; Writing—review & editing, B.H. and A.A.L.; Funding acquisition, K.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Gulf University for Science and Technology grant number ISG9.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Available from corresponding author on request.

Acknowledgments: This project has been partially supported by Gulf University for Science and Technology, including by project code: ISG Case 9. We thank G. Valle-Perez for valuable discussions.

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

Appendix A. The Impact of the Number of Iterations

In this brief section, we give examples of the impact of changing the length n of binary strings.

For very small numbers of iterations, simplicity bias is not pronounced. This is primarily due to the fact that the complexity measure is not very sensitive to small changes in complexity values. To illustrate, in Figure A1a a plot is shown for $n = 5$ iterations only. In this case, there is some evidence of simplicity bias (i.e., an inverse relation between complexity and probability), but the relation is not strong. A similar weak connection between complexity and probability was observed for very short sequences in an earlier time series study of simplicity bias [7]. In Figure A1b, clear simplicity bias is observed with $n = 25$. At this length, the complexity measure is sufficiently sensitive and we have sufficient sampling to reveal order-of-magnitude variations in probability.

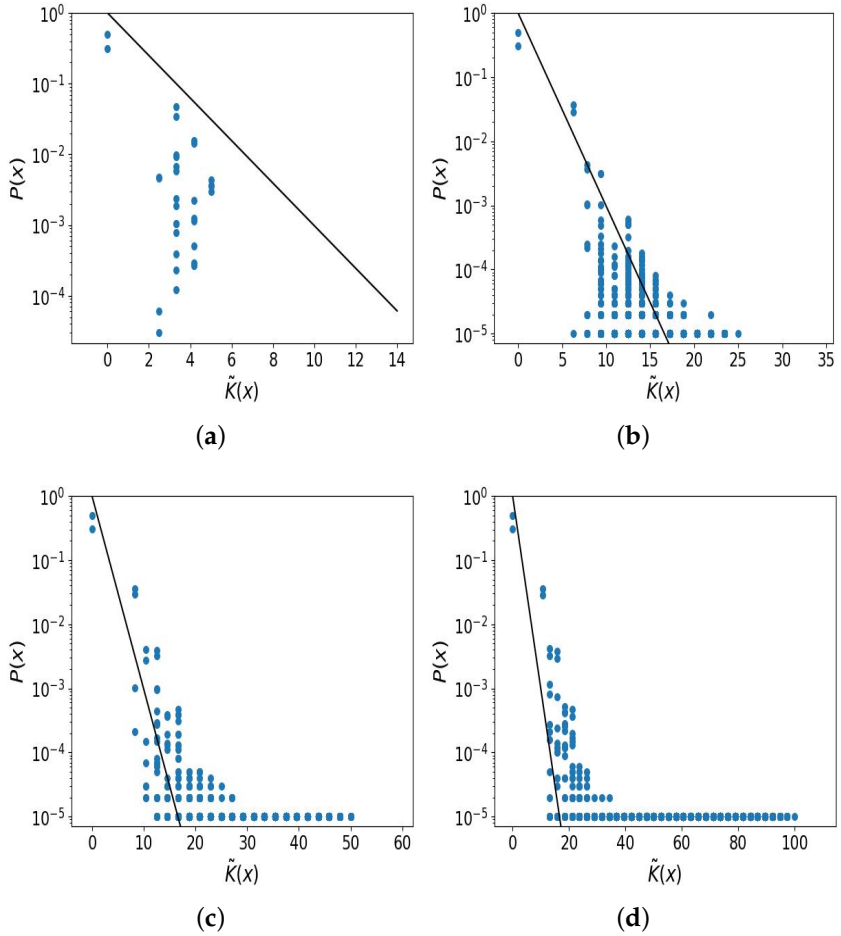

Figure A1. Simplicity bias with different number of iterations. (**a**) With $n = 5$ iterations, there is some simplicity bias but it is not pronounced; (**b**) with $n = 25$ iterations, the simplicity bias is very clear; with (**c**) $n = 50$ iterations there is still clear simplicity bias, but a long 'tail' begins to emerge, illustrating low-frequency patterns; (**d**) with $n = 100$ iterations, there is still some simplicity bias but the 'tail' has become more dominant and the simplicity bias is less clear.

In theory, for larger output complexities simplicity bias should be more clear [1], and in fact AIT results in general are typically stated up to $O(1)$ terms, which become irrelevant when pattern complexities become asymptotically large [16]. However, in this logistic map example, we find that for longer iterations, the pattern of simplicity bias becomes more noisy and less clear; see Figure A1c,d. For a larger n, the number of possible patterns becomes exponentially large and the one million samples produce many patterns of frequency ≈ 1, yielding a long flat 'tail' in the graph.

Appendix B. Alternate Figures Highlighting Density of Points

In this section, we repeat the some of the plots from the main text, but also provide semi-transparent data points to highlight the density of data points.

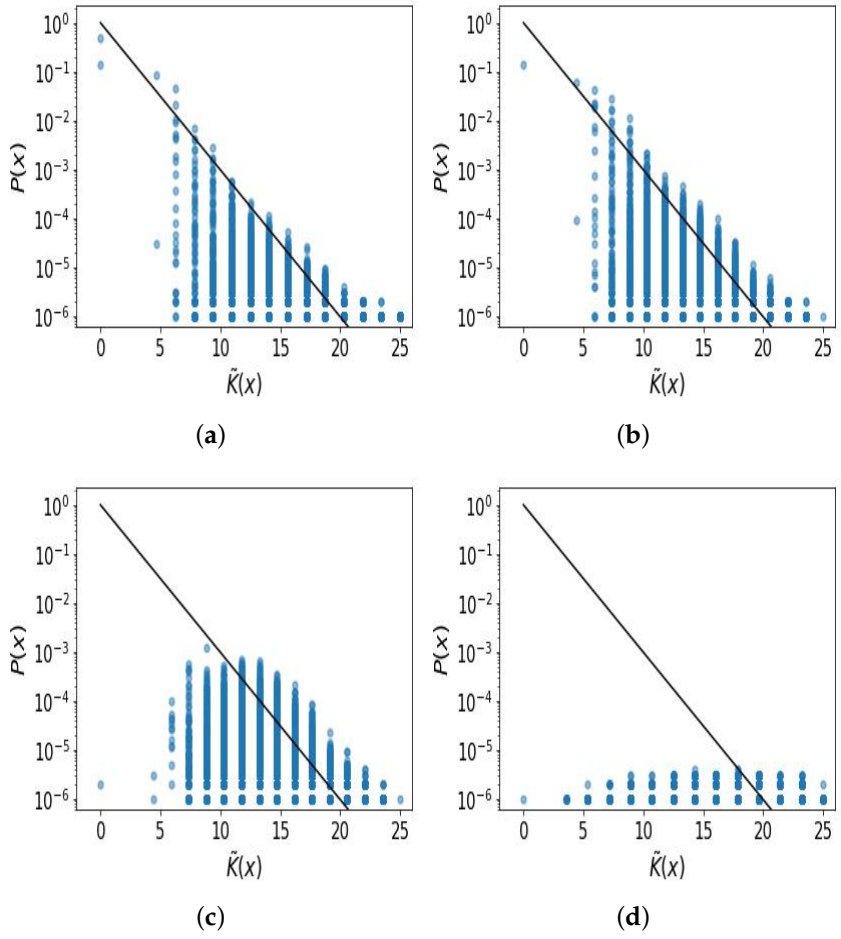

Figure A2. Simplicity bias in the logistic map, which is the same as in Figure 3, but with semi-transparent data points. (**a**) sampling $\mu \in [0.0, 4.0]$; (**b**) sampling $\mu \in [3.0, 4.0]$; (**c**) sampling $\mu \in [3.57, 4.0]$; (**d**) $\mu = 4.0$;

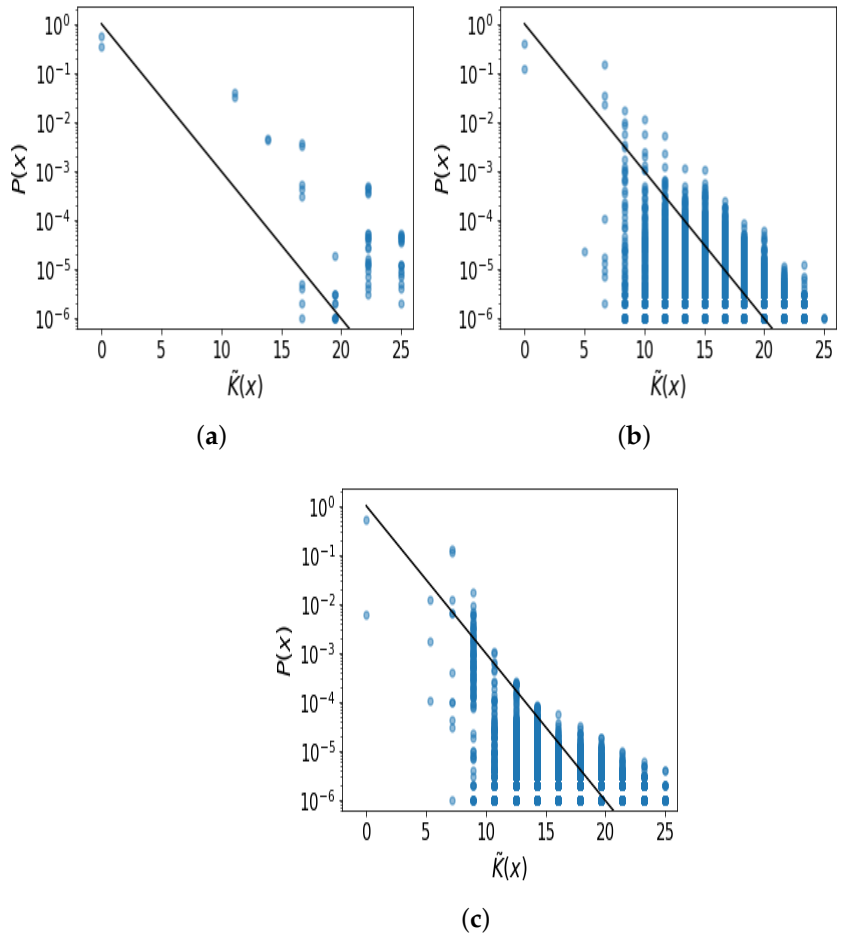

Figure A3. Simplicity bias in (**a**) the logistic, (**b**) Gauss map, and (**c**) sine map, the same as in Figure 5, but with semi-transparent data points.

References

1. Dingle, K.; Camargo, C.Q.; Louis, A.A. Input–output maps are strongly biased towards simple outputs. *Nat. Commun.* **2018**, *9*, 761. [CrossRef] [PubMed]
2. Dingle, K.; Pérez, G.V.; Louis, A.A. Generic predictions of output probability based on complexities of inputs and outputs. *Sci. Rep.* **2020**, *10*, 4415. [CrossRef]
3. Solomonoff, R.J. A preliminary report on a general theory of inductive inference (revision of report v-131). *Contract AF* **1960**, *49*, 376.
4. Kolmogorov, A.N. Three approaches to the quantitative definition of information. *Probl. Inf. Transm.* **1965**, *1*, 3–11. [CrossRef]
5. Chaitin, G.J. A theory of program size formally identical to information theory. *J. ACM* **1975**, *22*, 329–340. [CrossRef]
6. Dingle, K.; Batlle, P.; Owhadi, H. Multiclass classification utilising an estimated algorithmic probability prior. *Phys. D Nonlinear Phenom.* **2023**, *448*, 133713. [CrossRef]
7. Dingle, K.; Kamal, R.; Hamzi, B. A note on a priori forecasting and simplicity bias in time series. *Phys. A Stat. Mech. Its Appl.* **2023**, *609*, 128339. [CrossRef]
8. Johnston, I.G.; Dingle, K.; Greenbury, S.F.; Camargo, C.Q.; Doye, J.P.K.; Ahnert, S.E.; Louis, A.A. Symmetry and simplicity spontaneously emerge from the algorithmic nature of evolution. *Proc. Natl. Acad. Sci. USA* **2022**, *119*, e2113883119. [CrossRef]
9. Lempel, A.; Ziv, J. On the complexity of finite sequences. *IEEE Trans. Inf. Theory* **1976**, *22*, 75–81. [CrossRef]
10. Ziv, J.; Lempel, A. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory* **1977**, *23*, 337–343. [CrossRef]
11. Delahaye, J.P.; Zenil, H. Numerical evaluation of algorithmic complexity for short strings: A glance into the innermost structure of algorithmic randomness. *Appl. Math. Comput.* **2012**, *219*, 63–77. [CrossRef]
12. Soler-Toscano, F.; Zenil, H.; Delahaye, J.-P.; Gauvrit, N. Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines. *PLoS ONE* **2014**, *9*, e96223. [CrossRef] [PubMed]
13. May, R.M. Simple mathematical models with very complicated dynamics. *Nature* **1976**, *261*, 459–467. [CrossRef] [PubMed]

14. Hasselblatt, B.; Katok, A. *A First Course in Dynamics: With a Panorama of Recent Developments*; Cambridge University Press: Cambridge, UK, 2003.
15. Hilborn, R.C. *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*; Oxford University Press on Demand: New York, NY, USA, 2000.
16. Li, M.; Vitanyi, P.M.B. *An Introduction to Kolmogorov Complexity and Its Applications*; Springer: New York, NY, USA, 2008.
17. Calude, C.S. *Information and Randomness: An Algorithmic Perspective*; Springer: Berlin/Heidelberg, Germany, 2002.
18. Gács, P. *Lecture Notes on Descriptional Complexity and Randomness*; Boston University, Graduate School of Arts and Sciences, Computer Science Department: Boston, MA, USA, 1988.
19. Shen, A.; Uspensky, V.; Vereshchagin, N. *Kolmogorov Complexity and Algorithmic Randomness*; American Mathematical Society: Providence, RI, USA, 2022; Volume 220.
20. Turing, A.M. On computable numbers, with an application to the entscheidungsproblem. *J. Math.* **1936**, *58*, 345–363.
21. Grunwald, P.; Vitányi, P. Shannon information and Kolmogorov complexity. *arXiv* **2004**, arXiv:cs/0410002.
22. Bennett, C.H. The thermodynamics of computation—A review. *Int. J. Theor. Phys.* **1982**, *21*, 905–940. [CrossRef]
23. Kolchinsky, A.; Wolpert, D.H. Thermodynamic costs of turing machines. *Phys. Rev. Res.* **2020**, *2*, 033312. [CrossRef]
24. Zurek, W.H. Algorithmic randomness and physical entropy. *Phys. Rev. A* **1989**, *40*, 4731. [CrossRef]
25. Kolchinsky, A. Generalized zurek's bound on the cost of an individual classical or quantum computation. *arXiv* **2023**, arXiv:2301.06838.
26. Mueller, M.P. Law without law: from observer states to physics via algorithmic information theory. *Quantum* **2020**, *4*, 301. [CrossRef]
27. Avinery, R.; Kornreich, M.; Beck, R. Universal and accessible entropy estimation using a compression algorithm. *Phys. Rev. Lett.* **2019**, *123*, 178102. [CrossRef] [PubMed]
28. Martiniani, S.; Chaikin, P.M.; Levine, D. Quantifying hidden order out of equilibrium. *Phys. Rev. X* **2019**, *9*, 011031. [CrossRef]
29. Ferragina, P.; Giancarlo, R.; Greco, V.; Manzini, G.; Valiente, G. Compression-based classification of biological sequences and structures via the universal similarity metric: experimental assessment. *BMC Bioinform.* **2007**, *8*, 252. [CrossRef]
30. Adams, A.; Zenil, H.; Davies, P.C.W.; Walker, S.I. Formal definitions of unbounded evolution and innovation reveal universal mechanisms for open-ended evolution in dynamical systems. *Sci. Rep.* **2017**, *7*, 997. [CrossRef] [PubMed]
31. Devine, S.D. *Algorithmic Information Theory for Physicists and Natural Scientists*; IOP Publishing: Bristol, UK, 2020.
32. Vitányi, P.M. Similarity and denoising. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2013**, *371*, 20120091. [CrossRef] [PubMed]
33. Cilibrasi, R.; Vitányi, P.M.B. Clustering by compression. *IEEE Trans. Inf. Theory* **2005**, *51*, 1523–1545. [CrossRef]
34. Levin, L.A. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Probl. Peredachi Informatsii* **1974**, *10*, 30–35.
35. Buchanan, M. A natural bias for simplicity. *Nat. Phys.* **2018**, *14*, 1154. [CrossRef]
36. Dingle, K.; Novev, J.K.; Ahnert, S.E.; Louis, A.A. Predicting phenotype transition probabilities via conditional algorithmic probability approximations. *J. R. Soc. Interface* **2022**, *19*, 20220694. [CrossRef]
37. Alaskandarani, M.; Dingle, K. Low complexity, low probability patterns and consequences for algorithmic probability applications. *Complexity* **2023**, *2023*, 9696075. [CrossRef]
38. Lind, D. Marcus, B. *An Introduction to Symbolic Dynamics and Coding*; Cambridge University Press: Cambridge, UK, 1995.
39. Kanso, A.; Smaoui, N. Logistic chaotic maps for binary numbers generations. *Chaos Solitons Fractals* **2009**, *40*, 2557–2568. [CrossRef]
40. Berger, A. *Chaos and Chance: An Introduction to Stochastic Apects of Dynamics*; Walter de Gruyter: Berlin, Germany, 2001.
41. Kaspar, F.; Schuster, H.G. Easily calculable measure for the complexity of spatiotemporal patterns. *Phys. Rev. A* **1987**, *36*, 842. [CrossRef] [PubMed]
42. Mingard, C.; Rees, H.; Valle-Pérez, G.; Louis, A.A. Do deep neural networks have an inbuilt occam's razor? *arXiv* **2023**, arXiv:2304.06670.
43. Feigenbaum, M.J. The universal metric properties of nonlinear transformations. *J. Stat. Phys.* **1979**, *21*, 669–706. [CrossRef]
44. Feigenbaum, M.J. Universal behavior in nonlinear systems. *Phys. D Nonlinear Phenom.* **1983**, *7*, 16–39. [CrossRef]
45. Binous, H. Bifurcation Diagram for the Gauss Map from the Wolfram Demonstrations Project. 2011. Available online: https://demonstrations.wolfram.com/BifurcationDiagramForTheGaussMap/ (accessed on 1 September 2023).
46. Patidar, V. Co-existence of regular and chaotic motions in the gaussian map. *Electron. J. Theor. Phys.* **2006**, *3*, 29–40.
47. Suryadi, M.T.; Satria, Y.; Prawadika, L.N. An improvement on the chaotic behavior of the gauss map for cryptography purposes using the circle map combination. *J. Phys. Conf. Ser.* **2020**, *1490*, 012045. [CrossRef]
48. Wolfram, S. Mitchell Feigenbaum (1944–2019), 4.669201609102990671853203820... 2023. Available online: https://writings.stephenwolfram.com/2019/07/mitchell-feigenbaum-1944-2019-4-669201609102990671853203820/ (accessed on 1 September 2023).
49. Griffin, J. The Sine Map. 2013. Available online: https://people.maths.bris.ac.uk/~macpd/ads/sine.pdf (accessed on 1 September 2023).
50. Dong, C.; Rajagopal, K.; He, S.; Jafari, S.; Sun, K. Chaotification of sine-series maps based on the internal perturbation model. *Results Phys.* **2021**, *31*, 105010. [CrossRef]
51. MacKay, D.J. *Information Theory, Inference, and Learning Algorithms*; Cambridge University Press: Cambridge, UK, 2003.
52. Moore, C. Unpredictability and undecidability in dynamical systems. *Phys. Rev. Lett.* **1990**, *64*, 2354. [CrossRef]

53. Watson, J.D.; Onorati, E.; Cubitt, T.S. Uncomputably complex renormalisation group flows. *Nat. Commun.* **2022**, *1364*, 7618. [CrossRef]
54. Wolfram, S. Undecidability and intractability in theoretical physics. *Phys. Rev. Lett.* **1985**, *54*, 735. [CrossRef] [PubMed]
55. Wolfram, S. *A New Kind of Science*; Wolfram Media: Champaign, IL, USA, 2002.
56. Svozil, K. *Randomness & Undecidability in Physics*; World Scientific: Singapore, 1993.
57. Lloyd, S. Uncomputability and physical law. In *The Incomputable: Journeys beyond the Turing Barrier*; Springer: Cham, Switzerland, 2017; pp. 95–104.
58. Aguirre, A.; Merali, Z.; Sloan, D. *Undecidability, Uncomputability, and Unpredictability*; Springer: Cham, Switzerland, 2021.
59. Lathrop, R.H. On the learnability of the uncomputable. In *ICML*; Citeseer: Forest Grove, OR, USA, 1996; pp. 302–309.
60. Valle-Perez, G.; Camargo, C.Q.; Louis, A.A. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv* **2018**, arXiv:1805.08522.
61. Mingard, C.; Skalse, J.; Valle-Pérez, G.; Martínez-Rubio, D.; Mikulik, V.; Louis, A.A. Neural networks are a priori biased towards boolean functions with low entropy. *arXiv* **2019**, arXiv:1909.11522.
62. Bhattamishra, S.; Patel, A.; Kanade, V.; Blunsom, P. Simplicity bias in transformers and their ability to learn sparse boolean functions. *arXiv* **2022**, arXiv:2211.12316.
63. Yang, G.; Salman, H. A fine-grained spectral perspective on neural networks. *arXiv* **2019**, arXiv:1907.10599.
64. Lloyd, S. Measures of complexity: a nonexhaustive list. *IEEE Control. Syst. Mag.* **2001**, *21*, 7–8.
65. Mitchell, M. *Complexity: A Guided Tour*; Oxford University Press: Oxford, UK, 2009.
66. Bialek, W.; Nemenman, I.; Tishby, N. Complexity through nonextensivity. *Phys. A Stat. Mech. Its Appl.* **2001**, *302*, 89–99. [CrossRef]
67. Bialek, W.; Nemenman, I.; Tishby, N. Predictability, complexity, and learning. *Neural Comput.* **2001**, *13*, 2409–2463. [CrossRef]
68. Coe, J.B.; Ahnert, S.E.; Fink, T.M.A. When are cellular automata random? *EPL Europhys. Lett.* **2008**, *84*, 50005. [CrossRef]
69. Arnold, L.; Jones, C.K.; Mischaikow, K.; Raugel, G.; Arnold, L. *Random Dynamical Systems*; Springer: Berlin/Heidelberg, Germany, 1995.
70. Doan, T.S.; Engel, M.; Lamb, J.S.W.; Rasmussen, M. Hopf bifurcation with additive noise. *Nonlinearity* **2018**, *31*, 4567. [CrossRef]
71. Dingle, K.; Lamb, J.S.W.; Lázaro-Camí, J.-A. Knudsen's law and random billiards in irrational triangles. *Nonlinearity* **2012**, *26*, 369. [CrossRef]
72. Hamzi, B.; Dingle, K. Simplicity bias, algorithmic probability, and the random logistic map. *Phys. D Nonlinear Phenom.* **2024**, *463*, 134160. [CrossRef]
73. White, H.S. Algorithmic complexity of points in dynamical systems. *Ergod. Theory Dyn. Syst.* **1993**, *13*, 807–830. [CrossRef]
74. Brudno, A.A. The complexity of the trajectories of a dynamical system. *Russ. Math. Surv.* **1978**, *33*, 197. [CrossRef]
75. V'yugin, V.V. Ergodic theorems for algorithmically random points. *arXiv* **2022**, arXiv:2202.13465.
76. Zenil, H.; Kiani, N.A.; Marabita, F.; Deng, Y.; Elias, S.; Schmidt, A.; Ball, G.; Tegnér, J. An algorithmic information calculus for causal discovery and reprogramming systems. *Iscience* **2019**, *19*, 1160–1172. [CrossRef]
77. Terry-Jack, M.; O'keefe, S. Fourier transform bounded kolmogorov complexity. *Phys. D Nonlinear Phenom.* **2023**, *453*, 133824. [CrossRef]