Santa Fe Institute Working Paper 16-10-022
arxiv.org:1610.07685 [cond-mat.stat-mech]

Minimum Memory for Generating Rare Events

Cina Aghamohammadi* and James P. Crutchfield†
Complexity Sciences Center and Department of Physics,
University of California at Davis, One Shields Avenue, Davis, CA 95616
(Dated: January 28, 2017)

We classify the rare events of structured, memoryful stochastic processes and use this to analyze sequential and parallel generators for these events. Given a stochastic process, we introduce a method to construct a new process whose typical realizations are a given process' rare events. This leads to an expression for the minimum memory required to generate rare events. We then show that the recently discovered classical-quantum ambiguity of simplicity also occurs when comparing the structure of process fluctuations.

PACS numbers: 02.50.-r 89.70.+c 02.50.Ey 02.50.Ga
Keywords: stochastic processes, large deviation theory, computational mechanics, fluctuation spectra

I. INTRODUCTION

One of the most critical computations today is identifying the statistically extreme events exhibited by large-scale complex systems. Whether in the domains of geology, finance, or climate, or whether in natural or designed systems (earthquakes and hurricanes versus market crashes and internet route flapping), one can argue that this class of problem is rapidly coming to define our present scientific and technological era [1]. Success in understanding the origins and occurrence of extreme events will have a major impact on social infrastructure and its sustainability.

Large deviation theory [2–7] is a relatively new and key tool for analyzing a process' full range of statistical fluctuations. Presaged by Shannon-McMillman-Breiman type theory in communication theory [8, 9], the mathematical development of large deviations was first pursued by Donsker and Varadhan [10]. In essence, it can be seen as a refinement of the Central Limit Theorem [11] or as a generalization of Einstein's fluctuation theory [12, 13]. Today, large deviation theory enters into physics in many different circumstances [7]. One can also formulate statistical mechanics in the language of large deviation theory [7, 14, 15]. And, it appears in abstract dynamical systems under the rubric of the thermodynamic formalism [16].

The following analyzes the memory resources required to generate, and so study, extreme events in structured temporal processes. It extends large deviation theory in a constructive way that leads to exact calculations of the spectrum of fluctuations for processes generated by finite-state hidden Markov models. Fortunately, in this setting the generation and fluctuation problems can be simply stated. And so, we first give a suitably informal introduction to process generators and fluctuation theory, leaving technical results for later.

* caghamohammadi@ucdavis.edu
† chaos@ucdavis.edu

II. MARKOV PROCESSES AND GENERATORS

A discrete-time, discrete-value *stochastic process* [17, 18] is the probability space $\mathcal{P} = \left\{\mathcal{A}^{\infty}, \Sigma, \mathbb{P}(\cdot)\right\}$. Here, $\mathbb{P}(\cdot)$ is the probability measure over the bi-infinite chain $X_{-\infty:\infty} = \ldots X_{-2}X_{-1}X_0X_1X_2\ldots$, where random variables X_i take values in a finite discrete alphabet $\mathcal{A}$ and Σ is the σ-algebra generated by the cylinder sets in $\mathcal{A}^{\infty}$. The following only considers ergodic stationary processes; that is, $\mathbb{P}(\cdot)$ is invariant under time translation—$\mathbb{P}(X_{i_1}X_{i_2}\cdots X_{i_m}) = \mathbb{P}(X_{i_1+n}X_{i_2+n}\cdots X_{i_m+n})$ for all n—and over successive realizations. A familiar important property of stochastic processes is *Markov order* [19]. This is the minimum history length R required by any generator to correctly generate the process. Specifically, R is the smallest integer such that:

$$\mathbb{P}(X_t|\ldots, X_{t-2}, X_{t-1}) = \mathbb{P}(X_t|X_{t-R}, \ldots, X_{t-2}, X_{t-1}) \ .$$

To keep matters uncomplicated, consider a process consisting of time series $\ldots 10010011\ldots$ of binary symbols. Having raw sequences in hand does represent the process' behaviors, but in and of themselves the sequences are not that useful. For example, how can we predict future symbols? What mechanisms drive the process' behaviors? Much more helpful in answering such questions is an algorithm that can produce the process' sequences. And, a good one can be used to simulate the process—generating example sequences, perhaps not even in the original data, but statistically similar—that allow one to predict future sequences, gain insight into the process' internal mechanisms, and estimate statistical properties.

Note that in most cases representing a process by spec-

ifying the probability measure $\mathbb{P}(\cdot)$ is impossible due to the infinite number of possible sequences. So, how should we represent processes? Is there a more compact way than specifying in-full the probability measure on the sequence sigma algebra? In a rather direct sense, *Markov chains* and *hidden Markov models* provide constructive answers. The quality of those answers depends, of course, on how useful these representations are. We now fill in their technical details, so that we can work with them.

Markov chains (MCs) [19, 20] and hidden Markov models (HMMs) [18, 21, 22] are widely-used algorithms for generating stochastic processes. Both consist of a set $\boldsymbol{\mathcal{S}}$ of states and a set of state-transition probabilities. Formally, both MCs and HMMs are specified by a tuple $\{\boldsymbol{\mathcal{S}}, \mathcal{A}, \{T^{(x)}, x \in \mathcal{A}\}\}$. In this, $\boldsymbol{\mathcal{S}}$ is a finite set of states, $\mathcal{A}$ is a finite alphabet, and $\{T^{(x)}, x \in \mathcal{A}\}$ is a set of $|\boldsymbol{\mathcal{S}}| \times |\boldsymbol{\mathcal{S}}|$ substochastic symbol-labeled transition matrices whose sum $T = \sum_{x \in \mathcal{A}} T^{(x)}$ is an stochastic matrix. In MCs states are past words, whereas in HMMs states and words are distinct. Hence, their states are hidden—not directly unobserved.

Consider an example HMM where $\boldsymbol{\mathcal{S}} = \{A, B\}$, $\mathcal{A} = \{0, 1\}$, $T^{(0)} = \begin{bmatrix} p & 0 \\ 0 & 0 \end{bmatrix}$, and $T^{(1)} = \begin{bmatrix} 0 & 1-p \\ 1 & 0 \end{bmatrix}$. An HMM such as this is graphically depicted via its state-transition diagram—a directed graph with labeled edges. $\boldsymbol{\mathcal{S}}$ is the set of graph nodes and the edge from node i to j is labeled by $p|x$ corresponding to the HMM transition with probability $p = T^{(x)}_{ij}$ that goes from state i to j and generates symbol x. Fig. 1 shows the state-transition diagram for a two-state HMM that generates a process called the binary-symbol *Even Process* [23].

The Even Process highlights why HMMs are such useful algorithms. Since the HMM states are tied to be individual past words, HMMs can be arbitrarily more compact than MCs for the same process. In this case, the Even Process is an infinite Markov order process since its current state can depend on arbitrarily long histories. (If only 1s have been observed, it can be in either state A or state B.) Said in terms of algorithmic complexity, the MC representing the Even Process requires an infinite number of Markov states, each associated with a history 1^k0, $k = 0, 1, 2, \ldots$. In contrast, as the figure makes plain, the Even Process' HMM takes only two states. This is why HMMs are preferred algorithms compared to MCs when it comes to generating processes.

When using HMMs as process generators we can restrict attention to those that are *unifilar*: the current state and next symbol uniquely determine the next state. Unifilar HMMs are important since they are perfect predictors of their process. (The same is not generally true of a process' nonunifilar HMM generators. We return to the important, but subtle distinction between prediction and generation using HMMs at the end.) For any given process there is an infinite number of unifilar HMM generators; so the restriction imposes no loss of representational generality. Given all of the alternative HMMs, though, which do we choose?

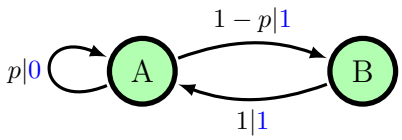


FIG. 1: State-transition diagram for the hidden Markov generator of the Even Process, which consists of random binary sequences with an even number of 1s separated by arbitrary-length blocks of 0s.

III. OPTIMAL SERIAL AND PARALLEL GENERATORS

Let's say Alice wants to generate the Even Process. The previous remarks indicate that she should not use an MC algorithm since it has infinite states and, as a result, needs an infinite amount of memory to generate the process. Ands so, she uses an HMM algorithm, which is finite. To do this, she writes a computer program: If the current state is A, with probability p the program emits symbol 0 and stays at state A and with probability $1 - p$ it emits symbol 1 and goes to state B. However, if the current state is B, it generates symbol 1 and goes to the state A. The program continues in this fashion, again and again, and in the long run generates a realization of the Even Process. Moreover, if Alice chooses to start in A or B using the asymptotic state probability distribution π, then the resulting realization is stationary.

Imagine that a long time has passed and the HMM is in state A. Alice decides to stop the program for now and return tomorrow to continue generating the same realization. She must make a decision, does she use the realization generated today or start all over again tomorrow? Not wanting to waste the effort already invested, she decides to use today's realization tomorrow and simply concatenate newly generated symbols.

The next day, though, can she randomly pick a state and continue generating? The answer is no. If she randomly picks state B, then there is a chance that after concatenating the old and new realizations together, the sequence has odd number of 1s between two 0s. However, she knows that the Even Process never generates such subsequences. Thus, if she wants to use today's realization tomorrow then, she must record the HMM's current state and continue generating from that state to-

morrow.[1]

Information theory [8] tells us that to record the current state Alice needs $\log_2 |\boldsymbol{\mathcal{S}}|$ bits of memory. This is the cost of *sequential generation*. And, it gives a quantitative way to compare algorithms across the infinite number of alternatives. If Alice wants to use less memory, she selects the HMM with the minimum number of states. Which representation achieves this?

Before answering, let's contrast another scenario, that for *simultaneous generation*. Now, Alice wants to generate $N \gg 1$ realizations for a given process simultaneously, but insists that the individual sequences be statistically independent. The latter means that she cannot simply generate a single realization and copy it N times. At first blush, it seems that she needs $N \log_2 |\boldsymbol{\mathcal{S}}|$ bits of memory. According to Shannon's source coding theorem [8, 24], though, she can compress the sequence information and, for large N, she needs only $N \mathrm{H}[\boldsymbol{\mathcal{S}}] \leq N \log_2 |\boldsymbol{\mathcal{S}}|$ bits of memory, where $\mathrm{H}[\boldsymbol{\mathcal{S}}] = -\sum_{\sigma \in \boldsymbol{\mathcal{S}}} \pi(\sigma) \log_2 \pi(\sigma)$ is the Shannon entropy of the stationary probability distribution $\pi(.)$ over the HMM's states. That is, on average Alice needs $\mathrm{H}[\boldsymbol{\mathcal{S}}]$ bits of memory to generate each realization. So, if Alice wants to use less memory, she selects the process HMM with the minimum $\mathrm{H}[\boldsymbol{\mathcal{S}}]$ in the set of unifilar HMMs. Again, which representation achieves this?

Crutchfield and Young [25] showed that over all unifilar HMMs that generate a given process, there is a unique HMM with the minimum number of states. Surprisingly, this same HMM is also the one with the minimum entropy over its states. It is now known as the *ϵ-machine* [26, 27] and its state entropy is the process' *statistical complexity* C_μ [25, 26]. The consequence is that, for a given stochastic process, the minimum memory required for any unifilar HMM to sequentially generate it is $\log_2 |\boldsymbol{\mathcal{S}}_\epsilon|$ bits, where $\boldsymbol{\mathcal{S}}_\epsilon$ is the set of states in the process' ϵ-machine. And, for simultaneous generation the average minimum required memory for each realization is C_μ.

Today, C_μ is often used as a measure of structural complexity for stochastic processes, from stochastic resonance [28] to hydrodynamic flows [29], atmospheric turbulence [30], geomagnetic volatility [31], and single-molecule dynamics [32–34]. In short, we use ϵ-machines and C_μ to measure the memory inherent in a stochastic process. And, by the preceding argument we now know how they determine the memory required for sequential and parallel generation.

[1] The time period over which Alice pauses generation can be set to any duration—an hour, a minute, or a second. In particular, the period can be that required to generate a single symbol. In this case, after every symbol emitted Alice must know in what state the generator is. In short, Alice needs to remember the current state during generation.

IV. TYPICAL AND ATYPICAL BEHAVIORS

So far, the discussion implicitly assumed that models captured a process' typically observed behaviors. However, most stochastic processes exhibit statistical fluctuations and so occasionally generate atypical, statistically extreme behaviors. Now, we turn to define what we mean by typical and atypical behaviors. Once done, we finally state our problem: How much memory is needed to generate a process' atypical behaviors.

So, what does it mean that a process exhibits statistical fluctuations? Let's say Alice has a biased coin, meaning that when she flips it, the probability p of seeing heads is greater than one half. Alice now flips the coin $n \gg 1$ times and see k heads. The Strong Law of Large Numbers [35] guarantees that for large n, the ratio k/n almost surely converges to p:

$$\mathbb{P}\left(\lim_{n\to\infty} \frac{k}{n} = p\right) = 1 \ .$$

Informally, for large n the *typical sequence* has close to p percent Heads. This does not mean that Alice never sees long runs of all Heads or all Tails, for example. It simply means that the latter are rare events.

We now show that a process' typically observed realizations are those sequences in its so-called typical set. Consider a given process and let $\mathcal{A}^n$ denote the set of length-n sequences. Then, for an arbitrary $\epsilon > 0$ the process' *typical set* [8, 36, 37] is:

$$A^n_\epsilon = \{w : 2^{-n(h_\mu+\epsilon)} \leq \mathbb{P}(w) \leq 2^{-n(h_\mu-\epsilon)}, w \in \mathcal{A}^n\}, \quad (1)$$

where h_μ is the process' *metric entropy* (Shannon entropy rate) [38]:

$$h_\mu(\mathcal{P}) = -\lim_{n\to\infty} \frac{1}{n} \sum_{w\in\mathcal{A}^n} \mathbb{P}(w) \log_2 \mathbb{P}(w) \ .$$

According to the *Shannon-McMillan-Breiman theorem* [24, 39, 40], for a given $\epsilon \ll 1$ and sufficiently large n:

$$\mathbb{P}(w \notin A^n_\epsilon, w \in \mathcal{A}^n) \leq \epsilon \ . \quad (2)$$

There are two important lessons here. First, coming from Eq. (1), all sequences in the typical set have approximately the same probability. Second, coming from Eq. (2), for large n the probability of sequences falling outside the typical set is close to zero—they are rare.

One consequence is that sequences generated by a stationary ergodic process fall into one of three partitions;

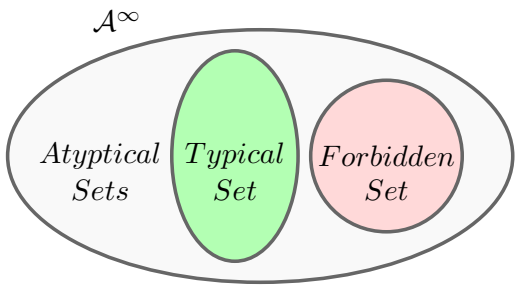


FIG. 2: For a given process, the space $\mathcal{A}^\infty$ of its realizations is partitioned into forbidden sequences, sequences in the typical set, and sequences in atypical sets.

see Fig. 2. The first contains those that are never generated by a process—sequences with zero probability. (For example, the Even Process cannot generate realizations containing a subsequence in $\{01^{2k+1}0\}$, $k = 0, 1, 2, \ldots$—those with an odd number of 1s between 0s.) These are the *forbidden sequences*. The second partition consists of those in the typical set—the set with probability close to one, as in Eq. (1). And, the last contains sequences in a family of atypical sets—realizations that are rare to different degrees. We now refine this classification.

Mirroring the familiar *Boltzmann weight* in statistical physics [41], in the $n \to \infty$ limit, we define the subsets $\Lambda_U^{\mathcal{P}} \subset \mathcal{A}^\infty$ for a process $\mathcal{P}$ as:

$$\Lambda_{U,n}^{\mathcal{P}} = \left\{ w : -\frac{\log_2 \mathbb{P}(w)}{n} = U, \ w \in \mathcal{A}^n \right\}$$
$$\Lambda_U^{\mathcal{P}} = \lim_{n\to\infty} \Lambda_{U,n}^{\mathcal{P}} \ . \tag{3}$$

In effect, this partitions $\mathcal{A}^\infty$ into subsets $\Lambda_U^{\mathcal{P}}$ in which all $w \in \Lambda_U^{\mathcal{P}}$ have the same probability decay rate U. Physics vernacular would speak of the sequences having the same *energy density* U.[2] Figure 3 depicts these subsets as "bubbles" of equal energy. (Though, to be clear about their "shape", these subsets are isomorphic to Cantor sets.) The definition guarantees that any bi-infinite sequence $\mathcal{P}$ generates belongs to one of these sets. Equation (1) says the typical set is that bubble with energy equal to the process' entropy rate: $U = h_\mu$. All the other bubbles contain rare events.

When Alice uses a process' HMM to generate realizations, what she does is generate sequences in the typical set with probability close to one and, rarely, atypical sequences. Imagine, though, that Alice is interested in a particular class of rare sequences, those in a different isoenergy bubble; say, those with energy U in the set $\Lambda_U^{\mathcal{P}}$. How can Alice efficiently generate these rare sequences? We now show that she can find a new process $\mathcal{P}^U$ whose typical set is $\Lambda_U^{\mathcal{P}}$.

[2] U, considered as a random variable, is sometimes called a *self process* [5].

V. GENERATING RARE EVENTS

To do this, we return to considering HMMs for a given process. With suitable HMMs and a precise definition of a process' atypical sequences we can now ask, How much memory is required to generate them? How does this compare to the memory required to generate typical behaviors?

Given a process $\mathcal{P}$ and its ϵ-machine $M(\mathcal{P})$, How do we construct an ϵ-machine $M(\mathcal{P}^U)$ that generates $\mathcal{P}$'s atypical sequences at some energy $U \neq h_\mu$? Here, we answer this question by constructing a map $\mathcal{B}_\beta : \mathcal{P} \to \mathcal{P}_\beta$ from the original $\mathcal{P}$ to a new process $\mathcal{P}_\beta$. The latter is parametrized by $\beta \in \mathbb{R}/\{0\}$ which indexes the atypical set of interest. Both processes $\mathcal{P} = \{\mathcal{A}^\infty, \Sigma, \mathbb{P}(\cdot)\}$ and $\mathcal{P}_\beta = \{\mathcal{A}^\infty, \Sigma, \mathbb{P}_\beta(\cdot)\}$ are defined on the same measurable sequence space. The measures differ, but their supports (allowed sequences) are the same. We refer to $\mathcal{B}_\beta$ as the *β-map*.

Assume we are given $M(\mathcal{P}) = \{\boldsymbol{\mathcal{S}}, \mathcal{A}, \{T^{(x)}, x \in \mathcal{A}\}\}$. We will now show that for every probability decay rate or energy U, there exists a particular β such that $M(\mathcal{P}_\beta)$ typically generates the words in $\Lambda_{U,n}^{\mathcal{P}}$ for large n. The β-map which establishes this is calculated by a construction that relates $M(\mathcal{P})$ to $M(\mathcal{P}_\beta) = \{\boldsymbol{\mathcal{S}}, \mathcal{A}, \{\mathbf{S}_\beta^{(x)}, x \in \mathcal{A}\}\}$—the HMM that generates $\mathcal{P}_\beta$:

1. For each $x \in \mathcal{A}$, construct a new matrix $\mathbf{T}_\beta^{(x)}$ for which $\left(\mathbf{T}_\beta^{(x)}\right)_{ij} = \left(\mathbf{T}^{(x)}\right)_{ij}^{\beta}$.
2. Construct a new matrix $\mathbf{T}_\beta = \sum_{x\in\mathcal{A}} T_\beta^{(x)}$.
3. Calculate $\mathbf{T}_\beta$'s maximum eigenvalue $\widehat{\lambda}_\beta$ and corresponding right eigenvector $\widehat{\mathbf{r}}_\beta$.
4. For each $x \in \mathcal{A}$, construct new matrices $\mathbf{S}_\beta^{(x)}$ for which:

$$\left(\mathbf{S}_\beta^{(x)}\right)_{ij} = \frac{\left(\mathbf{T}_\beta^{(x)}\right)_{ij} (\widehat{\mathbf{r}}_\beta)_j}{\widehat{\lambda}_\beta (\widehat{\mathbf{r}}_\beta)_i} \ . \tag{4}$$

Theorem 1. *For the new process $\mathcal{P}_\beta$ in the limit $n \to \infty$ the probability of the set $\Lambda_{U,n}^{\mathcal{P}}$ converges to one $\lim_{n\to\infty} \mathbb{P}_\beta(\Lambda_{U,n}^{\mathcal{P}}) = 1$ where:*

$$U = \beta^{-1}\left(h_\mu(\mathcal{P}_\beta) - \log_2 \widehat{\lambda}_\beta\right) \ . \tag{5}$$

Also, in the same limit the process $\mathcal{P}_\beta$ assigns equal energies over all the members of the set $\Lambda_{U,n}^{\mathcal{P}}$.

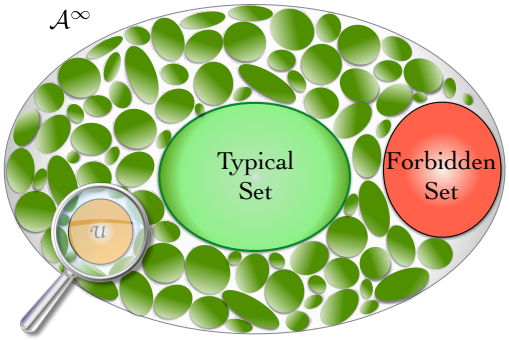


FIG. 3: $\mathcal{A}^\infty$ partitioned into Λ_Us—isoenergy or equal probability-decay-rate bubbles—in which all sequences in the same Λ_U have the same energy U. The typical set is one such bubble with energy equal to metric entropy: $U = h_\mu$. Another important partition is that of the forbidden sequences, in which all sequences have zero probability. The forbidden set can also be interpreted as the subset of sequences with infinite energy.

Proof. *See the appendix.*

As a result, for large n the process $\mathcal{P}_\beta$ typically generates the set $\Lambda^{\mathcal{P}}_{U,n}$, where $U = \beta^{-1}\big(h_\mu(\mathcal{P}_\beta) - \log_2 \widehat{\lambda}_\beta\big)$. And so, there is a one-to-one relationship between β and U and we can denote the process $\mathcal{P}_\beta$ by $\mathcal{P}^U$. More formally, every word in $\Lambda^{\mathcal{P}}_U$ with probability measure one is in the typical set of process $\mathcal{P}_\beta$.

This says that changing β controls which class of rare events we focus on. Informally, the β-map acts like a magnifier (Fig. 3) by enhancing particular isoenergy bubbles. That is, changing β moves the magnifier from one bubble to another. The β-map construction guarantees that the HMMs $M(\mathcal{P})$ and $M(\mathcal{P}_\beta)$ have the same states and transition topology: $\left(\mathbf{T}^{(x)}_\beta\right)_{ij} \neq 0 \iff \left(\mathbf{S}^{(x)}_\beta\right)_{ij} \neq 0$. The only difference is in their transition probabilities. Thus, $M(\mathcal{P}_\beta)$ is also a unifilar HMM, but not necessarily an ϵ-machine, since the latter requires a minimal set of states. Minimality is not guaranteed by the β-map. Typically, though, $M(\mathcal{P}_\beta)$ is an ϵ-machine and there is only a finite number of βs for which it is not. (More detailed development along these lines will appear in a sequel.)

Historically, a similar map was found for the first time in 1961 by Miller [42], but only for Markov order-one processes. In the setting of continuous-time first-order Markov evolution a similar map was introduced by Refs. [43, 44] (*s-ensemble*), by Ref. [45] (*biased ensemble*), and Ref. [46, 47] (*exponential tilting*). In these settings $\mathcal{P}_\beta$ is sometimes called an *auxiliary process* [45].

The β-map for unifilar HMMs and, consequently, for finite- or infinite-order discrete-time discrete-value Markov processes was introduced for the first time in 1993 [4]. A proof was not provided, which we remedy here, explaining why this β-map works so generally. There $\mathcal{P}_\beta$ was called the *twisted distribution*.

VI. MEMORY SPECTRA

For an arbitrary stochastic process $\mathcal{P}$, using its ϵ-machine the last section presented a method to construct a (unifilar) generator whose typical set is the process $\mathcal{P}^U$—the rare events of the original $\mathcal{P}$. Now, we determine the minimum memory required to generate $\mathcal{P}^U$. Recalling the earlier coding-theoretic arguments, this is rather straightforward to answer. The minimum memory to generate $\mathcal{P}^U$ is determined by the size of its ϵ-machine. (As noted, this is the size of $M(\mathcal{P}_U)$ except for finite number of U.)

And so, except for a finite number of rare-event classes, to sequentially generate sequences in a given rare class, one requires the same memory—the number $|\boldsymbol{\mathcal{S}}|$ of states—as that to generate the original process. This is our first result on the minimum Markov memory for a process' rare events.

The story differs markedly, however, for simultaneous generation. The minimum required memory for simultaneous generation of $\mathcal{P}^U$ is $C_\mu(\mathcal{P}^U)$, putting the earlier coding argument together with last section's calculations. More to the point, this is generally not equal to $C_\mu(\mathcal{P})$. To better appreciate this result, let us examine three examples.

First, consider the Two-Biased Coins (TBC) Process with $p = 1/3$, whose ϵ-machine is shown in Fig. 4(a)(top left). To generate its realizations one flips a biased coin repeatedly. At first, label Heads a 0 and Tails a 1. After flipping, switch the labels and call a Head 1 and Tail 0. A TBC process sequence comes from repeating these steps endlessly. As Fig. 4(a) makes clear, there is a symmetry in the process. In the stationary distribution π, state A has probability half, as does state B, and this is independent of p. This gives $C_\mu(\mathcal{P}) = 1$ bit. Recalling the β-map construction, we see that changing β does not change the ϵ-machine topology. All that changes is p. This means, in turn, that the symmetry in states remains and $C_\mu(\mathcal{P}^U) = 1$ is constant over allowed Us (or βs); $C_\mu(U)$ versus U is the horizontal line shown in Fig. 4(c).

What energies are allowed? The TBC Process has a finite energy range: $U \in [\approx 0.586, \approx 1.584]$. From Eq. (3) we see that the maximum $U_{\max}$ corresponds to the bubble with the rarest sequences that can be generated. Conversely, $U_{\min}$ corresponds to the bubble with the most probable. The energy extremes delimit the domain of the $C_\mu(\mathcal{P}^U)$ curves in Fig. 4(c). In addition, the U asso-

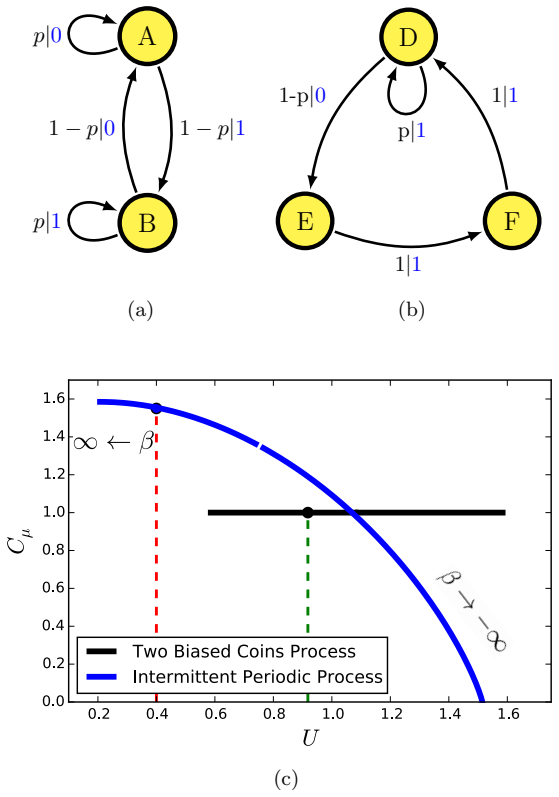


FIG. 4: (a) Two-Biased Coins (TBC) Process ϵ-machine generator. (b) Intermittent Periodic Process (IPP) ϵ-machine generator. (c) Statistical complexity C_μ versus energy U (or fluctuation class) for each, along with the energies U^* at which their typical sets are found (vertical dashed lines).

ciated with $\mathcal{P}$'s typical set is marked in the figure with a dashed (green) vertical line near $U \approx 0.9183$.

The difference between the typical set and that with $U_{\min}$ is important to appreciate. The typical set is that *set* of sequences with probability close to one and with energy $U = h_\mu$. The latter is generally different from $U_{\min}$. That is, typical sequences are not necessarily the most probable sequences, considered individually, but rather they belong to the most probable subset—the typical set.

As a result of this analysis for this example, one 1 bit of memory is uniformly required for generating the TBC Process' events, rare or not and independent of which class of rare events we examine.

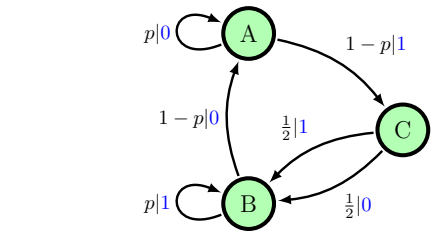
Second, this is not the general case, since $C_\mu(\mathcal{P}^U)$ can be a nonconstant function of U, as we now show. Consider the Intermittent Periodic Process (IPP) with $p = 0.35$; its ϵ-machine is given in Fig. 4(b) (top right). It gets its name since when $p = 0$, it periodically emits the subsequence 101 and when $p > 0$, it randomly inserts 1s. Using the β-map and Thm. 1 we can find the processes $\mathcal{P}^U$ and calculate their C_μ. Fig. 4(c) shows how their $C_\mu(\mathcal{P}^U)$ depends on U. The IPP is similar to the TBC Process in that it also has a finite energy range; IPP energies $U \in [\approx 0.207, \approx 1.515]$. It turns out that for any process with a finite ϵ-machine the allowed energy range is also finite. In addition, the U associated with $\mathcal{P}$'s typical set is marked in the figure with a dashed (red) vertical line near $U \approx 0.406$.

Thus, IPP's $C_\mu(\mathcal{P}^U)$ is a nontrivial function of U. Practically, this means that generating various rare-sequence classes requires less memory than for other classes. For example, for events with $U_{\max}$—$p = 1$ and $\beta \to -\infty$—ones needs no memory, since the class of maximum energy has only one sequence—the all-1s sequence. This can be generated by an IID process that emits only 1s. Generally, due to its IID character we do not need to remember or store the process' current state. In other words, the ϵ-machine $M(\mathcal{P}^U)$ that generates this class only has one state and so $C_\mu = 0$ bits there. For $U_{\min}$, occurring at $p = 0$ and $\beta \to \infty$, there are three "ground state" sequences—the three shifts of $\ldots 101101 \ldots$ and three equally probable states. Thus, $C_\mu(U_{\min}) = \log_2 3 \approx 1.585$ bits are necessary for generation.

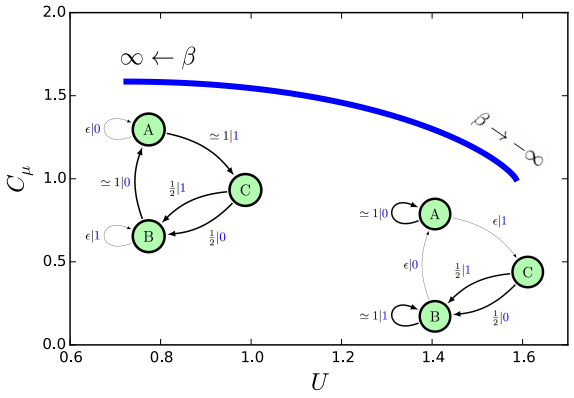
Third and finally, for a more complex example consider the process generated by the ϵ-machine with $p = 1/3$ given in Fig. 5(a) (top). Using the β-map and Thm. 1 we again find the processes $\mathcal{P}^U$ and calculate their C_μ, as shown in Fig. 5(b) (bottom). The difference between this process and IPP is that at no inverse temperature β do we have an IID process $\mathcal{P}_\beta$. As a consequence $C_\mu(\mathcal{P}^U)$ is nonzero for all allowed U.

The insets in Fig. 5(b) (bottom) highlight the details of the process' ϵ-machines for two limits of β. In the limit $\beta \to \infty$ the probability of B's self-transition vanishes and the probability of transiting from state B to A goes to one. Similarly, the probability of A's self-transition vanishes and the A-to-C transition probability goes to one. As a consequence, as shown in Fig. 5(b), the extreme process generates 0 then 1, then flips a coin to decide the outcome and then repeats the same steps again and again. The physical interpretation is that this limit captures the process' "ground states" and they have positive entropy density and memory.

In the complementary limit $\beta \to -\infty$, an interesting property emerges. The process breaks into two distinct subprocesses that link to each other only arbitrarily weakly. The first process consists of state B with a deterministic self-transition that generates 1s. And, the second subprocess consists of state A with a deterministic self-transition that and generates 0s. In other words, the



(a)



(b)

FIG. 5: (a) Process ϵ-machine generator. (b) Statistical complexity C_μ versus energy U for the ϵ-machine generator. Insets (bottom) display ϵ-machines for the processes generating the fluctuation extremes at $\beta \to \infty$ and $\beta \to -\infty$.

process has two phases that rarely switch between themselves. As a result, over moderate durations the process exhibits nonergodic behavior. We note that this has profound effects on predictability: substantial resources are required for predicting nonergodic processes [48], despite their requiring finite resources for generation.

VII. CONCLUDING REMARKS

To generate the rare behaviors of a stochastic process one can wait, if one wants, for exponentially long times for them to occur. Here, we introduced an alternative to rare-event generation from large deviation theory and its predecessors. Given a process, we first classified its events into those that are forbidden, typical, and atypical. And, then we refined the atypical class. For any chosen rare class we introduced an algorithm that constructs a new process, and its unifilar HMM, that typically generates those rare events. Appealing to the optimality of computational mechanics' ϵ-machines then allowed us to analyze the minimal memory costs of implementing rare-event generators. Depending on the goal—producing a single correct sample (sequential generation) or a large number of correct of samples (simultaneous generation) from the rare class of interest—memory cost differs. We studied both costs. Taken together the three examples analyzed give a complete survey of applying the method and how memory costs vary across classes of rare events.

There are two main types of algorithms for generating stochastic processes: Monte Carlo versus finite-state machine algorithms. Monte Carlo algorithms are appropriate if the process can be written as a probability distribution generated by a Hamiltonian system and if what we are interested are macroscopic statistics. For a given process, finding a compact Hamiltonian generator can be challenging. In addition, to generate long realizations using Monte Carlo algorithms one needs corrspondingly long initial data. This data, which changes during the algorithm, must be stored by the algorithm. And so, this approach can be memory intensive. These limitations do not exist for finite-state machine algorithms.

The introduction emphasized that we only focused on unifilar HMMs as process generators and then we constructed the minimal unifilar generator for a given class of rare events. The unifilar condition is necessary when using a process' past behavior to optimally predict its future [49]. However, one may not be interested in prediction, only generation for which unifilarity is not required. While removing unifilarity expands the space of HMMs, it greatly complicates finding minimal generators. For one, *nonunifilar* HMMs can be more memory efficient than unifilar HMMs for a given process [18, 50, 51]. For another, constructing a minimal nonunifilar HMM for a general process is still an open and hard question [52–54].

The required memory $C_\mu(\mathcal{P})$ for (unifilarly) generating realizations of a given process $\mathcal{P}$ has been used as a measure of structural complexity for over two decades. It places a total order over stochastic-process space, ranking processes by the difficulty to generate them. The theorem introduced here extends the measure $C_\mu(\mathcal{P})$ to the full memory spectrum $C_\mu(\mathcal{P}^U)$ to generate fluctuations.

As one consequence, this structural accounting introduces the new phenomenon of the *ambiguity of simplicity* [55] to the domain of fluctuation theory. Say that process A is simpler than process B, since it requires less memory to generate: $C_\mu(A) < C_\mu(B)$. However, if instead we are interested in the rarest events at U, we showed that it is possible that A is more complex than process B since it requires more memory for that event class: $C_\mu(A^U) > C_\mu(B^U)$. As Ref. [55] notes, this fundamental ambiguity flies in the face of appeals to simplicity via

Occam's Razor and practically impacts employing statistical model selection as it relies on a total order of model complexity.

The same fluctuation theory has recently been used to identify fluctuations in macroscopic thermodynamic functioning in Maxwellian Demons [56]. Moreover, the method can be applied to many stochastic systems to explore their rare behaviors, from natural processes observed in fluid turbulence [57, 58], physiology [59, 60], surface science [61, 62], meteorological processes [63], cosmic microwave background radiation [64], seismic time series [65] to designed systems found in finance [66–69], renewable energy [70, 71], and traffic [72, 73]. It gives a full description of a process, from its typical to its rare behaviors. And, it determines how difficult it is to simulate a process' rare events.

Finally, there is another potentially important application domain. The rapid progress in quantum computation and information suggest that, perhaps soon even, one will be able to generate processes, both classical and quantum, using programmable quantum systems. The equivalent memory C_q for the simultaneous quantum simulation of processes also has already been introduced [49, 74–77]. And so, a sequel will analyze quantum memory fluctuation spectra $C_q(U)$ and how they differ from the classical spectra introduced here.

ACKNOWLEDGMENTS

We thank Mehrnaz Anvari and John Mahoney for useful conversations. This material is based upon work supported by U. S. Army Research Laboratory and the U.S. Army Research Office under contract W911NF-13-1-0390.

PROOF OF THE THEOREM

This appendix establishes the main theorem via a single lemma relying on a process' cryptic order.

Cryptic order is a recently introduced topological property of stochastic processes [78] that is bounded by, but is rather different in motivation from, the more familiar Markov order [79]. Formally, given a process' ϵ-machine, its *cryptic order* is $K = \inf\left\{l : \mathrm{H}[\mathcal{S}_l | X_0 X_1 \cdots] = 0,\ l \in \mathbb{Z}\right\}$. Informally, this means that if we observe an infinite length realization, we can be certain about in which state the ϵ-machine is in after the K^{th} symbol [80].

Lemma 1. *For any given process with finite states and cryptic order, for every U and $\beta \in \mathbb{R}/0$ we have:*

$$\Lambda_U^{\mathcal{P}} = \Lambda^{\mathcal{P}_\beta}_{\beta U - \log_2 \widehat{\lambda}_\beta} \ .$$

Proof. *Consider an arbitrary word $w = x_0 x_1 \ldots x_{n-1} \in \mathcal{A}^n$ generated by process $\mathcal{P}$ where $n \gg 1$. Since the ϵ-machine is unifilar, immediately after choosing the initial state, all the successor states are uniquely determined. Using this, we can decompose w to two parts: The first part w_K is the first K symbols and the second part is w's remainder. Knowing w, the state σ_K and all successor states following $\sigma_{K+1}, \sigma_{K+2}, \ldots$ are uniquely determined. As a consequence, the probability of process $\mathcal{P}$ generating w can be written as:*

$$\mathbb{P}(w) = \mathbb{P}(w_K) \prod_{i=K}^{n-1} \left(\boldsymbol{T}^{(x_i)} \right)_{\sigma_i \sigma_{i+1}} \ .$$

We can adapt the energy definition in Eq. (3) to finite-length sequences. Then, w's energy is:

$$\begin{aligned} \mathcal{U}(w) &= -\frac{\log_2 \mathbb{P}(w)}{n} \\ &= -\frac{\log_2 \mathbb{P}(w_K)}{n} - \frac{\log_2 \left(\prod_{i=K}^{n-1} \left(\boldsymbol{T}^{(x_i)} \right)_{\sigma_i \sigma_{i+1}} \right)}{n} \ . \end{aligned}$$

Now, consider the same word, but this time generated by the ϵ-machine $M(\mathcal{P}_\beta)$. Then, the probability of generating w is:

$$\begin{aligned} \mathbb{P}_\beta(w) &= \mathbb{P}_\beta(w_K) \prod_{i=K}^{n-1} \left(\boldsymbol{S}_\beta^{(x_i)} \right)_{\sigma_i \sigma_{i+1}} \\ &= \mathbb{P}_\beta(w_K) \prod_{i=K}^{n-1} \frac{\left(\boldsymbol{T}_\beta^{(x_i)} \right)_{\sigma_i \sigma_{i+1}} (\widehat{\mathbf{r}}_\beta)_{\sigma_{i+1}}}{\widehat{\lambda}_\beta (\widehat{\mathbf{r}}_\beta)_{\sigma_i}} \\ &= \mathbb{P}_\beta(w_K) \frac{(\widehat{\mathbf{r}}_\beta)_{\sigma_n}}{(\widehat{\mathbf{r}}_\beta)_{\sigma_K}} \left(\widehat{\lambda}_\beta \right)^{n-K} \prod_{i=K}^{n-1} \left(\boldsymbol{T}_\beta^{(x_i)} \right)_{\sigma_i \sigma_{i+1}} \\ &= \mathbb{P}_\beta(w_K) \frac{(\widehat{\mathbf{r}}_\beta)_{\sigma_n}}{(\widehat{\mathbf{r}}_\beta)_{\sigma_K}} \left(\widehat{\lambda}_\beta \right)^{n-K} \left(\prod_{i=K}^{n-1} \left(\boldsymbol{T}^{(x_i)} \right)_{\sigma_i \sigma_{i+1}} \right)^\beta \ . \end{aligned}$$

The new energy for the same word is:

$$\begin{aligned} \mathcal{U}_\beta(w) &= -\frac{\log_2 \mathbb{P}(w)}{n} \\ &= -\frac{\log_2 \left(\mathbb{P}_\beta(w_K) \frac{(\widehat{\mathbf{r}}_\beta)_{\sigma_n}}{(\widehat{\mathbf{r}}_\beta)_{\sigma_K}} \right)}{n} - \frac{n-K}{n} \log_2 \widehat{\lambda}_\beta \\ &\quad - \beta \frac{\log_2 \left(\prod_{i=K}^{n-1} \left(\boldsymbol{T}^{(x_i)} \right)_{\sigma_i \sigma_{i+1}} \right)}{n} \ . \end{aligned}$$

In the limit of large n the first terms in $\mathcal{U}(w)$ and $\mathcal{U}_\beta(w)$ vanish and we have $\mathcal{U}_\beta(w) = \beta \mathcal{U}(w) - \log_2 \widehat{\lambda}_\beta$. Thus, for any two long sequences $w_1, w_2 \in \mathcal{A}^n$, if $\mathcal{U}(w_1) = \mathcal{U}(w_2)$, then $\mathcal{U}_\beta(w_1) = \mathcal{U}_\beta(w_2)$. And, the partitions induced by

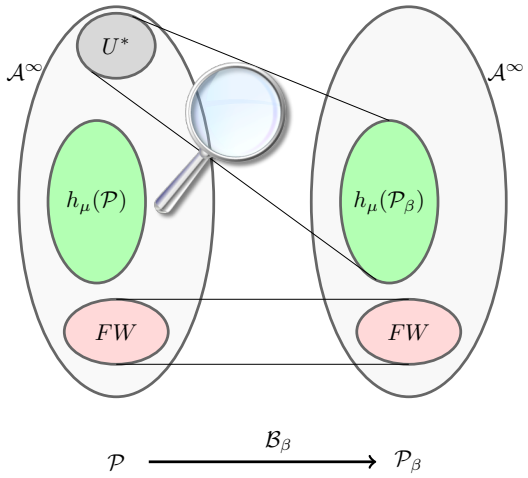


FIG. 6: The β-map acts like a magnifier: In the parlance of large deviation theory, it "twists" or "tilts" the sequence distribution in a way that focuses on the probability of a chosen rare-event class. Fixing β, the β-map changes the energy U of a class to $U_\beta = \beta U - \log_2 \widehat{\lambda}_\beta$. In particular, a subset with energy U^* maps to the typical set of a new process that has energy $h_\mu(\mathcal{P}_\beta)$. The set FW of forbidden sequences is invariant under the β-map.

Eq. (3) are invariant under the β-map. In other words, the energy of an arbitrary bubble after β-mapping changes from U to U_β, where:

$$U_\beta = \beta U - \log_2 \widehat{\lambda}_\beta \ .$$

This completes the lemma's proof.

This demonstrates how the β-map changes bubble energy: $U \to \beta U - \log_2 \widehat{\lambda}_\beta$. So, now we ask for the bubble (and its energy) that maps to the typical set of the new process $\mathcal{P}_\beta$. That is, we use the β-map to find the class $\Lambda_U^{\mathcal{P}}$ of rare sequences typically generated by $M(\mathcal{P}_\beta)$.

This sets up the theorem's proof. Using the fact that the process' metric entropy is the typical set's energy, the energy of $\mathcal{P}_\beta$'s typical set is $h_\mu(\mathcal{P}_\beta)$. (Refer to Fig. 6.) The lemma tells us how the β-map changes energy. Using this, we can identify the bubble with energy U^* that is typically generated by $M(\mathcal{P}_\beta)$, it has:

$$h_\mu(\mathcal{P}_\beta) = \beta U^* - \log_2 \widehat{\lambda}_\beta \ .$$

This completes the theorem's proof.

[1] J. P. Crutchfield. In P. Alsina and J. Perello, editors, *Cultures of Change — Changing Cultures*, pages 98–111, Barcelona, Spain, 2009. ACTAR Publishers. 1

[2] J. D. Deuschel and D. W. Stroock. *Large deviations*. Academic Press, New York, New York, 1989. 1

[3] J. A. Bucklew. *Large Deviation Techniques in Decision, Simulation, and Estimation*. Wiley-Interscience, New York, New York, 1990.

[4] K. Young and J. P. Crutchfield. *Chaos, Solitons, and Fractals*, 4:5 – 39, 1994. 5

[5] H. Touchette. *Physics Reports*, 478:1–69, 2009. 4

[6] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*, volume 38 of *Stochastic Modelling and Applied Probability*. Springer, New York, New York, second edition, 2009.

[7] R. S. Ellis. *Entropy, Large Deviations, and Statistical Mechanics*, volume 271 of *A Series of Comprehensive Studies in Mathematics*. Springer, New York, New York, 2012. 1

[8] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, second edition, 2006. 1, 3

[9] P. H. Algoet and T. M. Cover. *Ann. Prob.*, 16(2):899–909, 1988. 1

[10] M. D. Donsker and S. R. S. Varadhan. *Comm. Pure Appl. Math.*, 28:1–47, 1975. 1

[11] W. Feller. *An Introduction to Probability Theory and its Applications*. Wiley, New York, third, revised edition, 1970. 1

[12] A. Einstein. *Ann. d. Phys.*, 17:549–560, 1905. 1

[13] A. Einstein. *Ann. d. Phys.*, 19:371–381, 1906. 1

[14] Y. Oono. *Prog. Theo. Phys.*, 99:165, 1989. 1

[15] O. E. Lanford. In *Statistical Mechanics and Mathematical Problems*, pages 1–113. Springer, 1973. 1

[16] D. Ruelle. *Thermodynamic Formalism*. Addison-Wesley, Reading, 1978. 1

[17] N. F. Travers. *Bounds on Convergence of Entropy Rate Approximations in Hidden Markov Processes*. PhD thesis, University of California, Davis, 2013. 1

[18] D. R. Upper. *Theory and Algorithms for Hidden Markov Models and Generalized Hidden Markov Models*. PhD thesis, University of California, Berkeley, 1997. Published by University Microfilms Intl, Ann Arbor, Michigan. 1, 2, 7

[19] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, New York, New York, 2009. 1, 2

[20] J. R. Norris. *Markov Chains*, volume 2. Cambridge University Press, Cambridge, UK, 1998. 2

[21] L. R. Rabiner and B. H. Juang. *IEEE ASSP Magazine*, January:4–16, 1986. 2

[22] L. R. Rabiner. *Proc. IEEE*, 77(2):257–286 2

[23] J. P. Crutchfield and D. P. Feldman. *CHAOS*, 13(1):25–54, 2003. 2

[24] C. E. Shannon. *Bell Sys. Tech. J.*, 27:379–423, 623–656, 1948. 3

[25] J. P. Crutchfield and K. Young. *Phys. Rev. Let.*, 63:105–108, 1989. 3

[26] J. P. Crutchfield. *Nature Physics*, 8(1):17–24, 2012. 3

[27] C. R. Shalizi and J. P. Crutchfield. *J. Stat. Phys.*, 104:817–879, 2001. 3

[28] A. Witt, A. Neiman, and J. Kurths. *Phys. Rev. E*, 55:5050–5059, 1997. 3

[29] W. M. Gonçalves, R. D. Pinto, J. C. Sartorelli, and M. J. de Oliveira. *Physica A*, 257(1-4):385–389, 1998. 3

[30] A. J. Palmer, C. W. Fairall, and W. A. Brewer. *IEEE Trans. Geosci. Remote Sens.*, 38:2056–2063, 2000. 3

[31] R. W. Clarke, M. P. Freeman, and N. W. Watkins. *Phys. Rev. E*, 67:160–203, 2003. 3

[32] C.-B. Li and T. Komatsuzaki. *Phys. Rev. Lett.*, 111:058301, 2013. 3

[33] D. Nerukh, C. H. Jensen, and R. C. Glen. *J. Chem. Phys.*, 132(8):084104, 2010.

[34] D. Kelly, M. Dillingham, A. Hudson, and K. Wiesner. *PLoS One*, 7(1):e29703, 01 2012. 3

[35] R. Durrett. *Probability: Theory and examples*. Cambridge University Press, Cambridge, UK, 2010. 3

[36] S. Kullback. *Information Theory and Statistics*. Dover, New York, 1968. 3

[37] R. W. Yeung. *Information Theory and Network Coding*. Springer, New York, 2008. 3

[38] G. Han and B. Marcus. *IEEE Trans. Info. Th.*, 52(12):5251–5266, 2006. 3

[39] B. McMillan. *Ann. Math. Stat.*, 24:196–219, 1953. 3

[40] L. Breiman. *Ann. Math. Stat.*, 28(3):809–811, 1957. 3

[41] L. Boltzmann. *Lectures on Gas Theory*. Dover Publications, New York, 2013. 4

[42] H. D. Miller. *An. Math. Stat.*, pages 1260–1270 5

[43] J. P. Garrahan, R. L. Jack, V. Lecomte, E. Pitard, K. van Duijvendijk, and F. van Wijland. *J. Phys. A: Math. Theo.*, 42(7):075007 5

[44] L. O. Hedges, R. L. Jack, J. P. Garrahan, and D. Chandler. *Science*, 323(5919):1309–1313, 2009. 5

[45] R. L. Jack and P. Sollich. *Prog. Theo. Phys. Supp.*, 184:304–317, 2010. 5

[46] J. Van Campenhout and T. M. Cover. *IEEE Trans. Info. Th.*, 27(4):483–489, 1981. 5

[47] I. Csiszár. *Ann. Prob.*, pages 768–793, 1984. 5

[48] J. P. Crutchfield and S. Marzen. *Phys. Rev. E*, 91:050106(R), 2015. 7

[49] J. R. Mahoney, C. Aghamohammadi, and J. P. Crutchfield. *Scientific Reports*, 6:20495, 2016. 7, 8

[50] J. P. Crutchfield. *Physica D*, 75:11–54, 1994. 7

[51] W. Löhr and N. Ay. In *International Conference on Complex Sciences*, pages 265–276. Springer, New York, 2009. 7

[52] W. Löhr and N. Ay. *Adv. Complex Sys.*, 12(02):169–194, 2009. 7

[53] W. Löhr. *J. Systems Sci. Complexity*, 25(1):30–45, 2012.

[54] P. Gmeiner. *arXiv:1108.5303*, 2011. 7

[55] C. Aghamohammadi, J. R. Mahoney, and J. P. Crutchfield. The ambiguity of simplicity. *Phys. Lett. A.*, in press, 2016. arXiv:1602.08646. 7

[56] J. P. Crutchfield and C. Aghamohammadi. *arXiv:1609.02519*, 2016. 8

[57] M. Anvari, C. Aghamohammadi, H. Dashti-Naserabadi, E. Salehi, E. Behjat, M. Qorbani, M. K. Nezhad, M. Zirak, A. Hadjihosseini, and J. Peinke. *Phys. Rev. E*, 87(6):062139, 2013. 8

[58] Y. Tsuji and T. Ishihara. *Phys. Rev. E*, 68:026309, 2003. 8

[59] T. Kuusela. *Phys. Rev. E*, 69:031916, 2004. 8

[60] J. Prusseit and K. Lehnertz. *Phys. Rev. Lett.*, 98:138103, 2007. 8

[61] M. Waechter, F. Riess, T. Schimmel, U. Wendt, and J. Peinke. *Euro. Phys. J. B*, 41(2):259–277, 2004. 8

[62] A. L. S. Chua, C. A. Haselwandter, C. Baggio, and D. D. Vvedensky. *Phys. Rev. E*, 72:051103, 2005. 8

[63] P. Sura. *J. Atmos. Sci.*, 60:654–666, 2003. 8

[64] M. S. Movahed, F. Ghasemi, S. Rahvar, and M. R. R. Tabar. *Phys. Rev. E*, 84:021103, 2011. 8

[65] P. Manshour, S. Saberi, M. Sahimi, J. Peinke, A. F. Pacheco, and M. R. R. Tabar. *Phys. Rev. Lett.*, 102:014101, 2009. 8

[66] A. H. Shirazi, C. Aghamohammadi, M. Anvari, A. Bahraminasab, J. Tabar, M. R. R. Peinke, M. Sahimi, and M. Marsili. *J. Stat. Mech.: Th. Exp.*, 2013(2):1–11, 2013. 8

[67] C. Aghamohammadi, M. Ebrahimian, and H. Tahmooresi. *Physica A: Stat. Mech. App.*, 413:25–30, 2014.

[68] F. Ghasemi, M. Sahimi, J. Peinke, R. Friedrich, G. R. Jafari, and M. R. R. Tabar. *Phys. Rev. E*, 75:060102(R), 2007.

[69] J. P. Huang. *Physics Reports*, 564:1–56, 2015. 8

[70] M. R. R. Tabar, M. Anvari, G. Lohmann, D. Heinemann, M. Wächter, P. Milan, E. Lorenz, and J. Peinke. *Euro. Phys. J. Special Topics*, 223(12):2637–2644, 2014. 8

[71] M. Anvari, G. Lohmann, M. Wächter, P. Milan, E. Lorenz, D. Heinemann, M. R. R. Tabar, and J. Peinke. *New J. Physics*, 18(6):063027, 2016. 8

[72] S. Kriso, J. Peinke, R. Friedrich, and P. Wagner. *Phys. Lett. A*, 299(2-3):287–291, 2002. 8

[73] T. Nagatani. *Phys. Rev. E*, 61:3534–3540, 2000. 8

[74] P. M. Riechers, J. R. Mahoney, C. Aghamohammadi, and J. P. Crutchfield. *Phys. Rev. A*, 93(5):052317, 2016. 8

[75] M. Gu, K. Wiesner, E. Rieper, and V. Vedral. *Nature Comm.*, 3:762, 2012.

[76] R. Tan, D. R. Terno, J. Thompson, V. Vedral, and M. Gu. *Euro. Phys. J. Plus*, 129(9):1–12, 2014.

[77] C. Aghamohammadi, J. R. Mahoney, and J. P. Crutchfield. *arXiv:1609.03650*, 2016. 8

[78] J. P. Crutchfield, C. J. Ellison, and J. R. Mahoney. *Phys. Rev. Lett.*, 103(9):094101, 2009. 8

[79] N. Merhav, M. Gutman, and J. Ziv. *IEEE Trans. Info. Theo.*, 35(5):1014–1019, 1989. 8

[80] C. J. Ellison, J. R. Mahoney, and J. P. Crutchfield. *J. Stat. Phys.*, 136(6):1005–1034, 2009. 8