Title

Many roads to synchrony: Natural time scales and their algorithms

Journal

Physical Review E, 89(4)

ISSN

2470-0045

Authors

James, Ryan G
Mahoney, John R
Ellison, Christopher J
et al.

Publication Date

2014-04-01

DOI

10.1103/physreve.89.042135

Peer reviewed

Many roads to synchrony: Natural time scales and their algorithms

Ryan G. James,[1,2,*] John R. Mahoney,[3,†] Christopher J. Ellison,[1,4,‡] and James P. Crutchfield[1,5,§]
[1]*Complexity Sciences Center and Department of Physics, University of California at Davis, One Shields Avenue, Davis, California 95616, USA*
[2]*Department of Computer Science, University of Colorado at Boulder, Boulder, Colorado 80309, USA*
[3]*School of Natural Sciences, University of California, Merced, California 95344, USA*
[4]*Center for Complexity and Collective Computation, Wisconsin Institute for Discovery, University of Wisconsin—Madison, Madison, Wisconsin 53715, USA*
[5]*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501, USA*
(Received 23 December 2010; revised manuscript received 20 December 2013; published 18 April 2014)

We consider two important time scales—the Markov and cryptic orders—that monitor how an observer synchronizes to a finitary stochastic process. We show how to compute these orders exactly and that they are most efficiently calculated from the ϵ-machine, a process's minimal unifilar model. Surprisingly, though the Markov order is a basic concept from stochastic process theory, it is not a probabilistic property of a process. Rather, it is a topological property and, moreover, it is not computable from any finite-state model other than the ϵ-machine. Via an exhaustive survey, we close by demonstrating that infinite Markov and infinite cryptic orders are a dominant feature in the space of finite-memory processes. We draw out the roles played in statistical mechanical spin systems by these two complementary length scales.

DOI: 10.1103/PhysRevE.89.042135 PACS number(s): 02.50.Ga, 05.45.Tp, 02.50.Ey

I. INTRODUCTION

Stochastic processes are frequently characterized by the spatial and temporal length scales over which correlations exist. In physics, the range of correlations is a structural property giving, for example, the distance over which significant energetic coupling exists among a system's degrees of freedom [1]. In time-series analysis, knowing the temporal scale of correlations is key to successful forecasting [2]. In biosequence analysis, the decay of correlations along DNA base pairs determines in some measure the difficulty faced by a replicating enzyme as it "decides" to begin transcribing a gene [3]. In multiagent systems, one of an agent's first goals is to detect useful states in its environment [4]. The common element in these is that the correlation scale determines how quickly an observer—analyst, forecaster, enzyme, or agent—*synchronizes* to a process; that is, how it comes to know a relevant structure of the stochastic process.

We recently showed that there are a number of distinct, though related, length scales associated with synchronizing to stationary stochastic processes [5]. Here we show that these length scales are *topological*, depending only on the underlying graph topology of a canonical representation of the stochastic process. This reveals deep ties between the structure of a process's minimal sufficient statistic and synchronization of an observer. We also recently introduced another class of synchronization length scales based, not on state-based models, but on the convergence of sequence statistics [6]. We briefly compare these to the Markov and cryptic orders in Appendix D.

Specifically, we investigate measures of synchronization and their associated lengths scales for hidden Markov models (HMMs)—a particular class of processes with an internal (hidden) Markovian dynamic that produces an observed sequence. We focus on two such measures—the Markov order and the cryptic order—and show through a series of incremental steps how they can be efficiently and accurately computed from the process's minimal sufficient statistic, the ϵ-machine.

Our development proceeds as follows. After briefly outlining the required background in Sec. II, we introduce the two primary measures of interest in Sec. III and demonstrate their calculation via naïve methods in Sec. IV. Reflecting on a surprising finding in Sec. V, Sec. V B shows to how alleviate several weaknesses in the naïve approach. Then, borrowing relevant data structures from formal language theory, Sec. V C resolves the last of the issues. Together these result in an efficient algorithm for exactly calculating the Markov order when it is finite and for determining (in finite steps) when it is infinite. Building on this new understanding, Sec. VI goes on to show how to compute the second time scale—the cryptic order—through similar means. Leveraging this computational efficiency, we survey the Markov and cryptic orders among ϵ-machines in Sec. VII and conclude that infinite correlation is a dominate property in the space of memoryful stationary processes. The implication is that observer synchronization can be difficult, taking an arbitrarily long time in principle. However, Ref. [7] shows that synchronization occurs exponentially fast for the family of processes considered here. To illustrate how these time scales apply in practice, Sec. VIII characterizes correlations in one-dimensional spin systems. Finally, we conclude by discussing how these time scales compare to other measures of interest and by suggesting applications where they and their algorithms will prove useful.

II. BACKGROUND

We assume the reader has introductory knowledge of information theory and finite-state machines, such as that found in the first few chapters of Ref. [8] and Ref. [9], respectively. Our development makes particular use of ϵ-machines, a natural

*rgjames@ucdavis.edu
†jmahoney3@ucmerced.edu
‡cellison@cse.ucdavis.edu
§chaos@ucdavis.edu

representation of a process that makes many properties directly and easily calculable; for a review see Ref. [10]. A cursory understanding of symbolic dynamics, such as found in the first few chapters of Ref. [11], is useful for several results.

We denote subsequences in a time series as $X_{a:b}$, where $a \leqslant b$, to refer to the random variable sequence $X_a X_{a+1} X_{a+2} \cdots X_{b-1}$, which has length $b - a$. We drop an index when it is infinite. For example, the *past* $X_{-\infty:0}$ is denoted $X_{:0}$ and the *future* $X_{0:\infty}$ is denoted $X_{0:}$. We generally use w to refer to a *word*—a sequence of symbols drawn from an alphabet $\mathcal{A}$. We place two words, u and v, adjacent to each other to mean concatenation: $w = uv$. We define a *process* to be a joint probability distribution over $X_{:} = X_{:0}X_{0:}$.

A *presentation* of a given process is any state-based representation that generates the process. A process's ϵ-*machine* is its unique, minimal unifilar presentation [12]. The recurrent states of a process's ϵ-machine are known as the *causal states* and, at time t, are denoted $\mathcal{S}_t$. The causal states are the minimal sufficient statistic of $X_{:0}$ about $X_{0:}$. For a thorough treatment on presentations see Ref. [5].

III. PROBLEM STATEMENT

When confronted with a process, there are a number of natural questions to ask. How much memory does it have? Is it like coin flips or die rolls with no memory? Does it alternate between two values, requiring that the process remember its phase? Does it express patterns that are arbitrarily long, requiring an equally long memory? This type of memory is quantified by the Markov order as follows:

$$R \equiv \min\{\ell \,|\, \Pr(X_0|X_{-\ell:0}) = \Pr(X_0|X_{:0})\}. \tag{1}$$

To put it colloquially, how many prior observations must one remember to predict as well as remembering the infinite past? Markov chains have $R = 1$ by their very definition. In hidden Markov models, though their internal dynamics are Markovian ($R = 1$), their observed behavior can range from memoryless ($R = 0$) to infinite ($R = \infty$). A major goal in the following is to show how to compute a process's R efficiently and accurately given its ϵ-machine. In this vein it is prudent to recast Eq. (1) using causal states as follows:

$$\begin{aligned}
\Pr(X_0|X_{-R:0}) &= \Pr(X_0|X_{:0}) \\
\Rightarrow X_{:0} &\sim_\epsilon X_{-R:0} \\
\Rightarrow H[\mathcal{S}_0|X_{-R:0}] &= 0 \\
\Rightarrow R &= \min\{\ell \,|\, H[\mathcal{S}_0|X_{-\ell:0}] = 0\} \\
&= \min\{\ell \,|\, H[\mathcal{S}_\ell|X_{0:\ell}] = 0\},
\end{aligned} \tag{2}$$

where $X_{:0} \sim_\epsilon X_{-R:0}$ means that the infinite and the finite past of length R provide equivalent predictions of future behavior. In effect, since the past R observations predict just as well as the infinite past, the causal states are a function of length-R pasts.

The second primary length scale we discuss is the *cryptic order* k_χ [13]. Its definition builds from Eq. (2) as follows:

$$k_\chi \equiv \min\{\ell \,|\, H[\mathcal{S}_\ell|X_{0:}] = 0\}. \tag{3}$$

The difference between the two is that cryptic order is conditioned on the *infinite* future, as opposed to a finite one. This provides our interpretation of the cryptic order: k_χ is the number of causal states that cannot be *retrodicted*. That is, no matter how many future symbols we know, the first k_χ internal states the process visited cannot be inferred. Though it may not be obvious, it has been shown that $k_\chi \leqslant R$, and we provide an alternative proof of this in the remark following Proposition 2 in Appendix B.

There exist other length scales defined in a similar vein, relating to the information measures discussed in Ref. [6]. At present there is not an understanding of these measures comparable to that for the Markov and cryptic orders, and so, algorithms to calculate them do not currently exist. However, we do discuss several of their features in Appendix D.

IV. NAÏVE APPROACH

To illustrate a direct method of determining a process's Markov and cryptic orders, we appeal to yet another form of their definitions [5],

$$R = \min\{\ell | H[X_{0:\ell}] = \mathbf{E} + \ell\, h_\mu\}, \tag{4}$$

$$k_\chi = \min\{\ell | H[X_{0:\ell}, \mathcal{S}_\ell] = \mathbf{E} + \ell\, h_\mu\}, \tag{5}$$

where $\mathbf{E} = \mathrm{I}[X_{:0}; X_{0:}]$ is known as the *excess entropy* and $h_\mu = H[X_0|X_{:0}]$ is known as the *entropy rate* [14]. The intuition for these is identical to those above: Once we reach Markov (cryptic) order, we predict as accurately as possible. It is worth noting that these definitions only hold for *finitary* ($\mathbf{E} < \infty$), stationary processes.

These definitions lead to a simple way of determining a process's Markov and cryptic orders. To compute the Markov order, we calculate the entropy $H[X_{0:\ell}]$ of longer and longer blocks of contiguous observations until it begins to grow linearly. We call this function of ℓ the *block entropy curve*. The first ℓ at which $H[X_{0:\ell}]$ matches its linear asymptote is the Markov order. To compute the cryptic order, we perform a similar test, but rather than calculating the entropy of blocks of observations alone, we calculate the entropy $H[X_{0:\ell}, \mathcal{S}_\ell]$ of

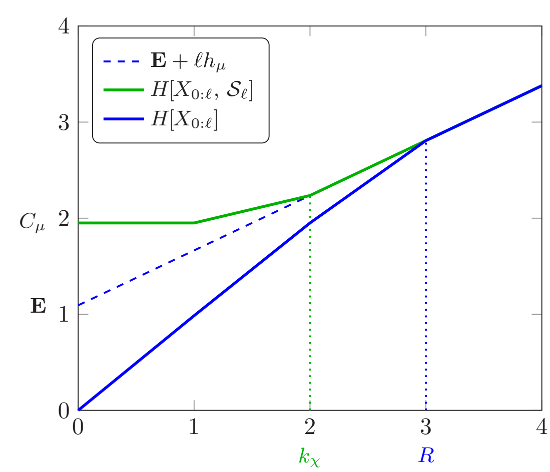

FIG. 1. (Color online) Block entropy and block-state entropy for the PSB process of Fig. 2: The block entropy curve reaches its asymptotic behavior ($\mathbf{E} + \ell\, h_\mu$) at $\ell = 3$, indicating a Markov order $R = 3$. The block-state entropy curve reaches the same asymptote at $\ell = 2$ and so the process is cryptic order $k_\chi = 2$.

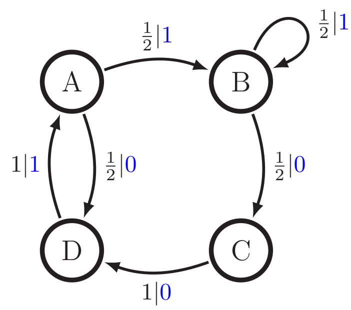

FIG. 2. (Color online) The phase-slip backtrack (PSB) process: Edges are labeled $p|s$ where p is the probability of an edge being followed and s is the symbol emitted upon traversing it.

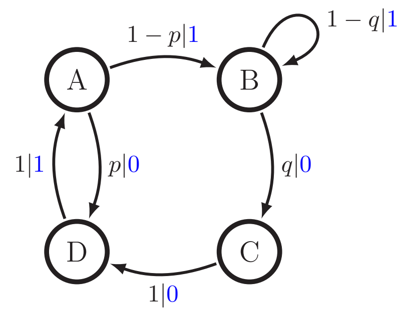

FIG. 3. (Color online) Phase-slip backtrack process with parametrized transition probabilities.

those blocks along with the causal states that are induced by those observations. We call this function of ℓ the *block-state entropy curve*. The cryptic order is the length at which the block-state entropy curve reaches its asymptotic linear behavior. This view of the two orders is shown in Fig. 1. The data for the block entropy and block-state entropy curves shown there come from the *phase-slip backtrack* (PSB) process show in Fig. 2.

It is important to point out the weaknesses of this approach. They are at least fourfold: One must (i) know h_μ exactly, (ii) know $\mathbf{E}$ exactly, (iii) be able to differentiate the block entropies being *exactly on* the asymptote from *less than machine precision away from* the asymptote, and (iv) be able to "guess" when R or k_χ are infinite in order to terminate the calculation. The first two are not prohibitive. The entropy rate h_μ can be computed exactly from any unifilar model of the process, and so its calculation can be done fairly easily [15]. Similarly, the excess entropy $\mathbf{E}$ can be computed if the joint distribution over both a unifilar model of the process and a unifilar model of the reverse of the process, at least one of which gauge-free, is on hand [16].

The last two weaknesses do not have such direct solutions. How are we to know if our entropy calculation at length ℓ is exactly equal to $\mathbf{E} + \ell h_\mu$? Or, instead, are the curve and linear asymptote so close that finite-precision estimates cannot differentiate them? Compounding this, what if $H[X_{0:\ell}]$ has not equaled $\mathbf{E} + \ell h_\mu$ by $\ell = 10^6$? Can one assume that it ever will? Perhaps the process is Markov order $R = 10^8$. These are the two particular weaknesses that need to be overcome.

V. MARKOV ORDER IS TOPOLOGICAL

In order to overcome the weaknesses of the naïve approach, we now assume that we possess the process's ϵ-machine. Its structure encodes the information needed to proceed. We start with the somewhat surprising observation that Markov order is not a probabilistic property, as seemingly suggested by Eq. (1), but rather a topological one. The first hint at this comes, though, in an empirical study. The question then becomes just how is this so. By way of answering it, we solve the fundamental problems noted with the naïve approach to Markov order. Several examples serve to drive home the idea and illustrate the calculation methods.

A. An observation

The first step forward in solving the two main problems encountered in the naïve Markov order method is to take a step back. Rather than considering the particular process generated by the machine in Fig. 2, we study the family of processes generated when its transition probabilities are varied while the structure remains the same. This family is shown by the parametrized machine of Fig. 3. If we compute block and block-state entropy curves for a random ensemble of processes from this family and plot the derivative of those curves (subtracting out their asymptotic behavior), we arrive at the block and block-state entropy convergence shown in Fig. 4.

As it dramatically demonstrates, the Markov and cryptic orders are *independent* of the transition probabilities in the machine's structure. Thus, any pattern relevant for prediction is encoded by the ϵ-machine's topology. The topological nature of the Markov order had previously been discovered and a loose upper bound provided [17]; our algorithm computes it exactly.

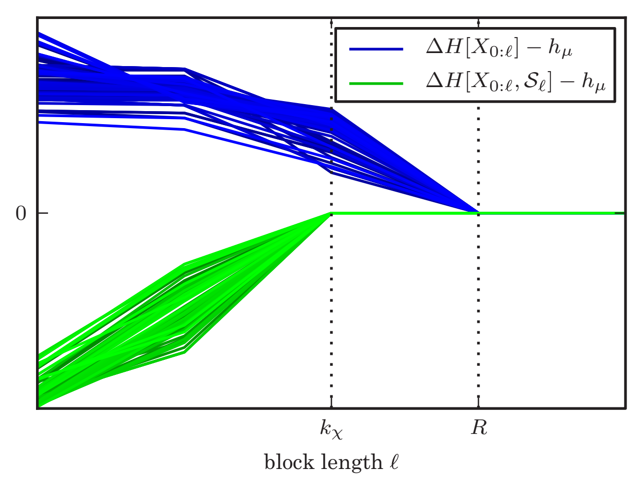

FIG. 4. (Color online) Entropy convergence curves versus block length ℓ for Fig. 3's family of processes with several dozen random values for p and q. The linear asymptotic behavior (h_μ) has been subtracted out of each curve (see the inset). The Markov order R and cryptic order k_χ are the lengths ℓ at which the blue (darker) and green (lighter) lines, respectively, reach zero. Thus, both orders are independent of the generating machine's probability parameters.

B. Synchronizing words

On careful inspection of Eq. (2), however, it is not surprising that the Markov order is a topological property. A conditional entropy $H[X|Y]$ vanishes only if X is a deterministic function of Y. In our case, $H[\mathcal{S}_R|X_{0:R}] = 0$ means that each length-R word determines a unique state of the model. We say that each word of length R is *synchronizing* [5]. (Particularly, we consider only *prefix-free* synchronizing words—those which have no initial subword that also synchronizes.) If one observes a process having no inkling as to which state its hidden Markov model began in, then after observing R symbols the exact state will be known. For a more formal treatment of synchronizing words, see Appendix A 1.

This provides an improved method of determining the Markov order. Enumerate all words of increasing length noting which have synchronized and which have not. When all the words at the current length have synchronized, then that length is the Markov order R. This procedure has been completed for the PSB process in Fig. 5. It can be verified that at lengths 0, 1, and 2 it is possible to still have ambiguity as to which state this system is in. For example, if the two symbols 10 are observed, the system may be in either state C or state D. One more observation is required to disambiguate which it is. Therefore, as observed previously, the Markov order for this process is $R = 3$. This method is improved by lexicographically enumerating words of increasing length *until* they synchronize to a single state. The longest such word—a prefix-free synchronizing word—is the Markov order R, since by that point every shorter word will have synchronized and, therefore, the causal states will be determined uniquely by words of that length.

This method addresses several weaknesses of the naïve approach. Now, neither $\mathbf{E}$ nor h_μ are needed, nor do we need to concern ourselves with the details of comparing nearly equal numerical values. However, the method relies on enumerating prefix-free synchronizing words, and it is quite possible for a process to have an infinite number of prefix-free synchronizing words. In these situations, it is not feasible to enumerate them all, hoping to identify the longest. To address this problem, we turn to formal language theory [9].

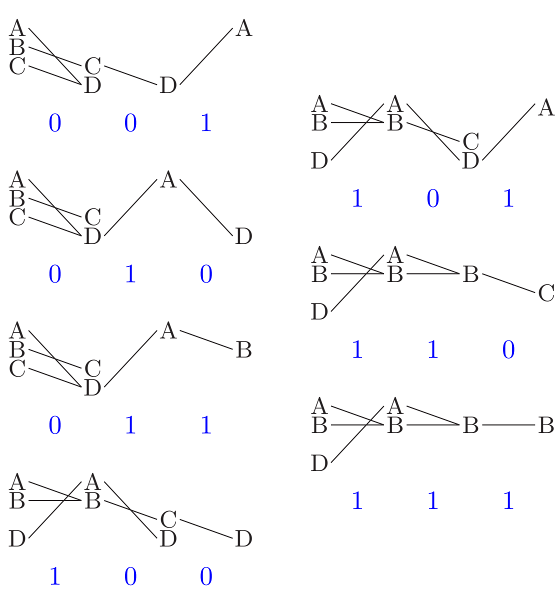

FIG. 5. (Color online) All observable words of length 3 for the PSB process. Each word has been annotated with the paths through which that word induces synchrony. It is not until the observation of three symbols that in all cases there is only a single possible state. There are, however, some words that induce synchrony more quickly.

C. State subset construction

The remaining problem is to find the longest prefix-free synchronizing word without having to enumerate them all. This can be accomplished with a standard algorithm from the theory of finite automata. We construct an object known as the *power automaton* (PA), so named since its states are elements of the power set of a given automaton's states.

Construction of the power automaton begins with a single state: the set of all states from the ϵ-machine. This is the PA's *start* state. Then, recursively, for each state in the PA and each symbol, consider all ϵ-machine states that can be reached by any ϵ-machine state within the current PA state on the currently considered symbol. A new PA state consisting of the set of ϵ-machine successor states is added, along with a directed edge from the current to the new PA state, labeled with the current symbol. Once the successors to each PA state have been determined, there will be a subgraph of the PA that is isomorphic to the recurrent ϵ-machine. This subgraph is the PA's *recurrent* component. When the ϵ-machine generates an ergodic process, this subgraph is the only strongly connected component with no outgoing edges. The remainder of the PA consists of *transient* states.

Synchronizing words are associated with particular PA paths. Each path begins in the start state and traverses edges in PA's transient portion. Eventually, the path continues to a PA recurrent state. Prefix-free synchronizing words have paths that end as soon as they reach a recurrent PA state. To find the longest prefix-free synchronizing word, we weight each edge in the PA's transient part with the value -1 and each edge in its recurrent part with 0. With these modifications, the Bellman-Ford algorithm can be employed to discover the path of least weight from the start state to any recurrent state. Due to the chosen weighting, the path of least weight is the longest.

The alternative Floyd-Warshall algorithm can also be used; see Ref. [18] for details regarding both. We choose the Bellman-Ford algorithm for two reasons. First, it works on graphs with negative weight and, second, it detects negative-weight cycles. A negative weight cycle here implies that the longest path is arbitrary (infinite) in length. For a more pedagogical statement of the algorithm, see Appendices C 1, C 2, and C 3.

This specifies a complete method for computing a process's Markov order efficiently and accurately from its ϵ-machine. First, construct the power automaton. Then weight the edges according to their status as transient or recurrent. Last, find the path of least weight from the start to a recurrent state. It runs in $O(|\mathcal{A}|2^{2N})$ time, where $\mathcal{A}$ is the number of observable symbols and N is the number of recurrent states. This quantity is exponential but finite. And, it depends only on integer calculations. In this way, it circumvents all the computational difficulties encountered in the naïve approach. Thus, if one can infer an accurate model from observations of a system, the problem of computing that system's Markov order is solved.

This method also provides a solution to weakness (iv) of the naïve algorithm (Sec. IV). When finite, the Markov order depends on the longest path through the transient states of the power automaton and, for an n state recurrent ϵ-machine, there are at most $f(n) := 2^n - n - 1$ transient states (subtracting n recurrent states and also the empty set). Since loops in the transient structure imply infinite Markov order, it follows that the longest possible path is one that visits each of the transient states. Thus, if the Markov order has not been found by $L = f(n)$, then it is safe to conclude that the Markov order is infinite. Since, the Markov order bounds the cryptic order, the same bound works for the cryptic order. As previously mentioned, Ref. [17] provides a tighter bound that is polynomial in the number of states: $f(n) = n(n+1)/2$. It is an open problem to find a tight upper bound for the Markov and cryptic orders in terms of both the number of states and alphabet symbols.

D. Examples

A variety of qualitatively different behaviors can be exhibited by the Markov order algorithm. Here, we illustrate the typical cases. Applying it to the PSB process, the algorithm produces the fairly simple transient structure consisting of three nodes—PA states ABCD, AB, and CD—seen in Fig. 6. There are two longest paths starting from PA start state ABCD and ending in a recurrent node: ABCD $\xrightarrow{1}$ AB $\xrightarrow{0}$ CD $\xrightarrow{1}$ A, which is traversed with the word 101, and ABCD $\xrightarrow{1}$ AB $\xrightarrow{0}$ CD $\xrightarrow{0}$ D, traversed with the word 100. This means that the longest prefix-free synchronizing words are 101 and 100, both of length 3, and therefore the PSB process's Markov order is $R = 3$.

The second process we analyze is shown in Fig. 7. Its PA has a slightly more complicated transient structure than that of the PSB process. Of particular note is the self-loop on PA state AB. This loop exists because ϵ-machine states A and B transition to each other on producing a 0. As a consequence, we cannot determine the state until observing a 1. The existence of nonsynchronizing words of arbitrary length implies that this process is non-Markovian; that is, $R = \infty$. The Bellman-Ford algorithm terminates as soon as it detects the corresponding negative-weight cycle in the transients.

Our third example is the Nemo process, shown in Fig. 8. Its transient structure is particularly simple: a single state represents all the recurrent states. Since the recurrent states simply permute on observing a 0, the word 0000... never reveals the current state. This is indicated by the self-loop on PA state ABC. This once again means that the process is non-Markovian and has $R = \infty$. This condition is detected by the algorithm as well.

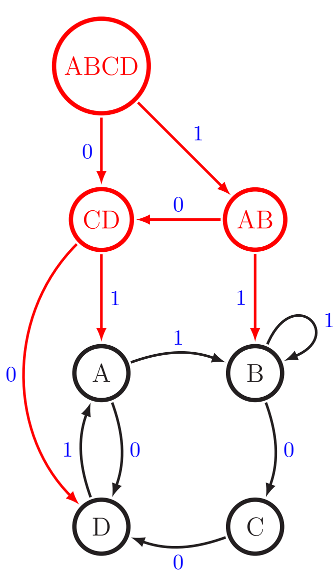

FIG. 6. (Color online) PSB process power automaton. The longest path beginning from state ABCD, traversing transient (red) edges, and ending in a recurrent (black) state is of length 3: ABCD $\xrightarrow{1}$ AB $\xrightarrow{0}$ CD $\xrightarrow{1}$ A(or $\xrightarrow{0}$ D).

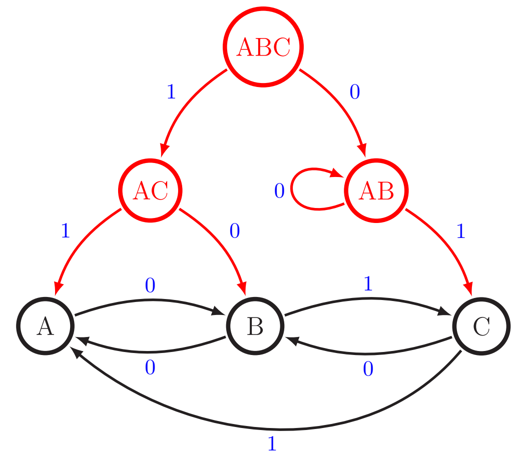

FIG. 7. (Color online) Typical complications in the PA for a finite-state non-Markovian process. The signature is the loop AB $\xrightarrow{0}$ AB in the transient structure. This means there is the possibility of an arbitrarily long series of observations that never synchronize and that, in turn, cause Markov order to diverge. Generically, loops in the transient structure can consist of more than one PA state.

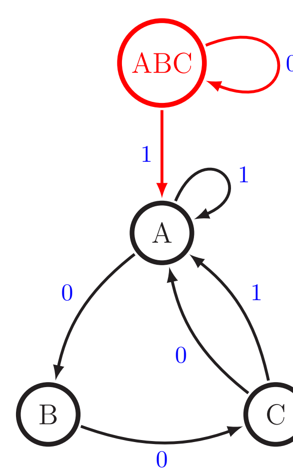

FIG. 8. (Color online) Like Fig. 7's process, the Nemo process here is non-Markovian. The Nemo process makes this perhaps clearer, however, since the recurrent states permute into each other on observing a 0. The transient structure captures this explicitly: ABC maps back to itself on a 0.

VI. CRYPTIC ORDER

We now turn to calculating a process's cryptic order k_χ. Recall that Eq. (3) involves a condition on the *infinite* future. With probability 1, each infinite future synchronizes for exactly synchronizing ϵ-machines [7]. We can then consider the problem of calculating k_χ to be that of determining as much of a state history as possible, given a prefix-free synchronizing word and the state to which it synchronized. The maximum number of states we cannot retrodict is then the cryptic order.

A. Calculation

Figure 9 depicts how the cryptic order is determined. Only the paths in Fig. 5 that survive all the way to synchrony (at the Markov order) are reproduced. From these, we determine how many symbols into each word we must parse (from the left) before the ϵ-machine is in one state only. The maximum such length is the cryptic order k_χ.

As with the Markov order, we need only consider prefix-free synchronizing words. However, we are again faced with the prospect that there may be an infinite number of prefix-free synchronizing words. Fortunately, a better method is available, and it too begins by constructing the power automaton. Now we examine the "veracity" of each transient edge. Take as an example the edge ABC $\xrightarrow{1}$ A in Fig. 8. It states that on producing a 1 from the superposition of states A, B, and C, the system can only transition to state A. For the cryptic order, we now condition on the fact that we are in state A and ask what states could have transitioned to A on a 1. Upon inspection, its clear that the system could have only transitioned from state A or state C on a 1. The core of the cryptic order algorithm is to inspect each transient edge in the power automaton in this manner, updating the PA's structure to "honestly" reflect the process's dynamics. In this instance, we create a state AC that transitions to state A on a 1 instead of transitioning from ABC on a 1.

After creating such a state, the automaton must be made consistent. To do this, subset construction is applied to include any newly added states. Generally, this creates new edges as well. And these, too, must be analyzed by use of the cryptic order algorithm. Once every edge has been inspected, some transient structure will remain. Once again, the longest path is the key, and the same edge-weighting method (Bellman-Ford) is employed to find it and so give the cryptic order. For a more instructional presentation of the algorithm, see Appendix C 4.

B. Examples

The ways in which the cryptic order algorithm modifies the power automaton are diverse. Each example from Sec. V D above illustrates a different behavior.

First, consider its behavior on the PSB process (Fig. 6), the final result of which is shown in Fig. 10. The edge CD $\xrightarrow{0}$ D in Fig. 6 can be removed since it does not represent a path that is true. To see why, note that to get to D on a 0, one must come from either state A or state C. However, since we are assuming CD, the process must be in either state C or D. The intersection of those two sets is state C and it is, therefore, the only possible state the system could have *actually* been in. Thus, CD $\xrightarrow{0}$ D is a misrepresentation from the cryptic order perspective and, in fact, it corresponds to the edge C $\xrightarrow{0}$ D, which already exists in the PA. So, the edge CD $\xrightarrow{0}$ D is removed.

This is not all, however. We must maintain the path's provenance. The edges that came into CD must be redirected to C (add edges ABCD $\xrightarrow{0}$ C and AB $\xrightarrow{0}$ C), since those are

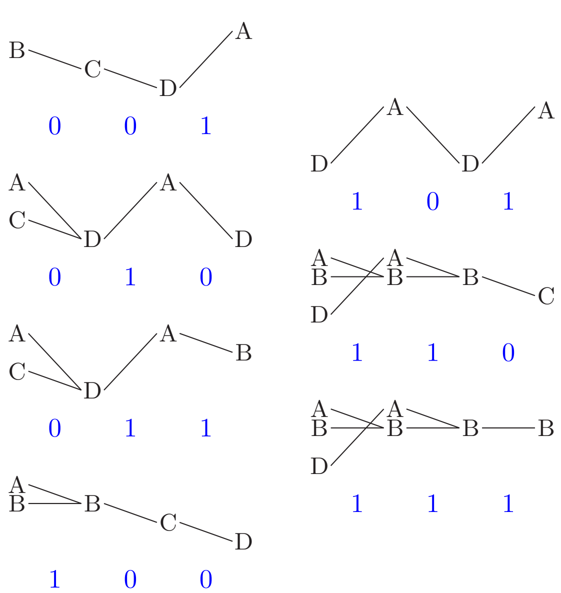

FIG. 9. (Color online) Key paths for determining cryptic order k_χ: We start with the paths in Fig. 5, except we remove paths that do not survive to the end of the sync word. The surviving paths give us the cryptic order: They each identify a single state by length $\ell = 2$ and so $k_\chi = 2$.

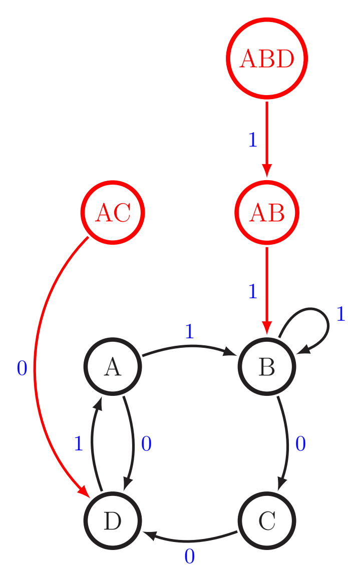

FIG. 10. (Color online) Cryptic order algorithm applied to the PSB process: The power automaton in Fig. 6 suggests that the word 11 could originate in any of A, B, C, or D. Careful inspection of the recurrent structure, though, shows that C cannot be the originator of 11, whereas the other three states can. The cryptic order algorithm accounts for such constraints. The longest path from a transient state to a recurrent state is ABD $\xrightarrow{1}$ AB $\xrightarrow{1}$ B and, therefore, $k_\chi = 2$.

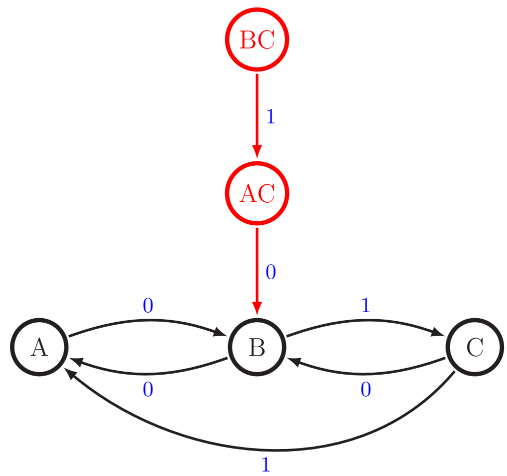

FIG. 11. (Color online) Cryptic order analysis of Fig. 7's process: The transient structure branch shown there—ABC $\xrightarrow{0}$ (AB $\xrightarrow{0}$ AB)* $\xrightarrow{1}$ C, with the arbitrarily long synchronizing word 00*1—can be perfectly retrodicted. Moreover, only a fragment of the left branch of the transient structure remains. This fragment has a length of 2 and so $k_\chi = 2$.

the edges that would have been traversed immediately prior to CD $\xrightarrow{0}$ D. Note that these edges are later removed in this recursive algorithm and so do not appear in Fig. 10. In the end, we see that the longest path from a start state to the recurrent states is 2 and, therefore, $k_\chi = 2$, one less than the Markov order $R = 3$.

Next consider the example from Fig. 7. The final output of the cryptic order algorithm is shown in Fig. 11. This process's PA consists of two major branches: one with a maximum depth of 2 and the other containing a loop. The cryptic order algorithm discovers that the branch with a loop is completely retrodictable. AB $\xrightarrow{1}$ C is actually B $\xrightarrow{1}$ C, and this creates edges AB $\xrightarrow{0}$ B and ABC $\xrightarrow{0}$ B, again to maintain provenance. The first of these newly added edges is also retrodictable: AB $\xrightarrow{0}$ B can only be A $\xrightarrow{0}$ B. The second, ABC $\xrightarrow{0}$ B, is in fact AC $\xrightarrow{0}$ B. Along this branch of the transient structure, we are thus only unable to retrodict the word 01, of which the 1 can be retrodicted, simply leaving us with AC $\xrightarrow{0}$ B. The previous branch is more easily analyzed, leaving us with BC $\xrightarrow{1}$ AC $\xrightarrow{0}$ B, the latter part of which was already in the PA from analyzing the other branch. This leaves a longest path of length 2, making $k_\chi = 2$. Thus, we see that this process is an example with infinite Markov order but finite cryptic order.

The last example to consider is the Nemo process. Recall that it is infinite Markov, as observed in Fig. 8. Applying the cryptic order algorithm results in the structure shown in Fig. 12. In this case, the transient structure grows under the algorithm. The edge ABC $\xrightarrow{1}$ A, connecting the transient to the recurrent structure in the power automaton, is modified by the algorithm since B cannot transition to A on a 1. The state AC is created and connected to A. Completing the power automaton structure from this state results in states AB and BC being added, forming the cycle AC $\xrightarrow{0}$ AB $\xrightarrow{0}$ BC $\xrightarrow{0}$ AC.

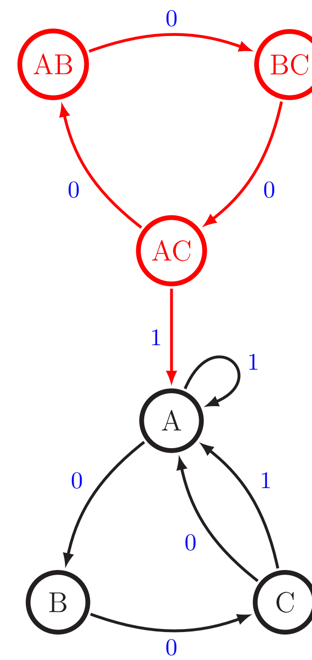

FIG. 12. (Color online) Cryptic order analysis of the Nemo process: Its power automaton (Fig. 8) contains the edge ABC $\xrightarrow{1}$ A. However, on closer inspection only states A and C can transition to A on a 1. This creates the AC state. When emitting a 0, AC becomes AB and on a second 0 that becomes BC. A third 0 completes the cycle. The edges indicate legitimate transitions as well: States that actually lead to AC on a 0 are BC and those that lead to BC are AB, and so on. This leads to a cycle in the cryptic order algorithm's calculated transient structure. Therefore, one concludes that $k_\chi = \infty$.

The algorithm terminates when the cycle is detected in this way. The cycle is valid as far as the cryptic order is concerned: Each of its states can be transitioned to from the recurrent state associated with the prior state in the cycle. The cycle results in an arbitrarily long path and, therefore, $k_\chi = \infty$.

VII. SURVEY

We illustrate the above results and algorithms, and their usefulness, by empirically answering several simple, but compelling, questions about the space of finitary processes. In particular, how typical are infinite Markov order and infinite cryptic order?

Restricting ourselves to topological ϵ-machines—those ϵ-machines with a distinct set of allowed transitions and equiprobable transition probabilities—we enumerate all binary-alphabet processes with a given number of states to which one can exactly synchronize. Reference [19] details their definition, the enumeration algorithm, and how it gives a view of the space of structured stochastic processes. For each of these ϵ-machines, we compute its Markov and cryptic orders. The result for all of the 1 132 613 six-state ϵ-machines is shown in Fig. 13.

The number of ϵ-machines that share a (R, k_χ) pair is encoded by the size of the circle at that (R, k_χ). The vast majority of processes—in fact, 98%—are non-Markovian at this state-size (six states). Furthermore, most (85%) of those

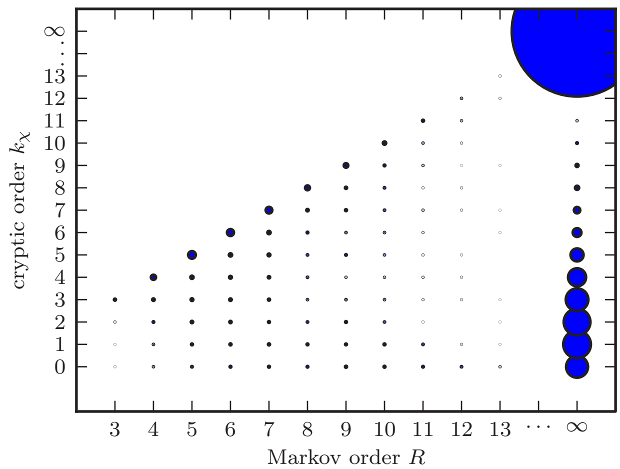

FIG. 13. (Color online) Distribution of Markov order R and cryptic order k_χ for all 1 132 613 six-state, binary-alphabet, exactly synchronizing ϵ-machines. Marker size is proportional to the number of ϵ-machines within this class at the same (R, k_χ).

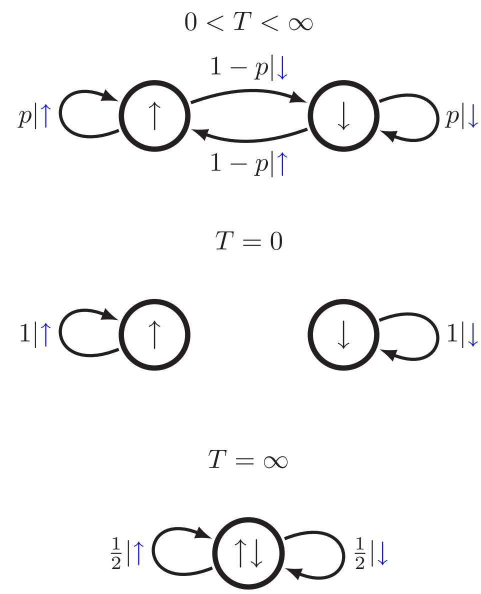

FIG. 14. (Color online) ϵ-Machines for a one-dimensional ferromagnetic Ising model as a function of temperature T, where $p = \frac{1}{2}(1 + \tanh \beta)$, the external field $B = 0$, and $J = k_B = 1$.

non-Markovian processes are also ∞ cryptic. However, this does not imply that synchronization is difficult; quite the contrary: Synchronization occurs exponentially quickly [7]. What this does mean is that with growing state size it becomes predominately likely that a given process has particular sequences which will not induce state synchronization.

Also of interest are the "forbidden" (R, k_χ) pairs within the space of six-state topological ϵ-machines. For example, ϵ-machines with $k_\chi = 4,5,8,10,11$ do not occur with $R = 13$. Also, processes with infinite Markov order and finite cryptic order appear to have a maximum cryptic order of $k_\chi = 11$, despite the fact that larger finite cryptic orders exist for finite Markov-order processes. These forbidden pairs provide insight into the space of minimal unifilar information sources. The space of processes generated by all six-state information sources, regardless of minimality or unifilarity, is in some sense "smooth." The restrictions of unifilarity and minimality fracture this space, making it harder to reason about. These forbidden pairs provide us with a probe into this space's organization.

VIII. SPIN CHAINS AND BEYOND

Although our primary goal was to precisely define length scales, several being new, and to present efficient calculation methods for them, it will be helpful to briefly draw out the physical meaning of Markov and cryptic orders by analyzing their role in spin chains and related systems. (A sequel will delve into this topic in greater depth.)

To start, recall that Ref. [20] showed that the Markov order R of an ϵ-machine representing a (one-dimensional) Ising spin system is upper bounded by the interaction range specified in a system's Hamiltonian. Consider first the ferromagnetic, one-dimensional, nearest-neighbor Ising model at different temperatures T. The ϵ-machines for this family of systems are shown in Fig. 14. As just noted, since the system has nearest-neighbor interactions, the Markov order should be $R = 1$. This is straightforward to see from the first ϵ-machine, which is for a finite temperature T. Without an observation there are two possible causal states the system could be in: ↑ or ↓. Once a single spin has been observed, however, the causal state is known exactly. This changes markedly at the temperature limits, though. At $T = 0$, the system is in a ground configuration of either all up spins or all down spins. Without an external field to break this symmetry an observation must be made to determine in which of these states it is and so the Markov order is still $R = 1$. In the presence of an external field, however, there is only a single ground state—that aligned with the field—and no observation is required to know in which state the system is. Thus, $R = 0$. At $T = \infty$ the system collapses to a single causal state where the next spin is entirely determined by thermal fluctuations and so the Markov order is $R = 0$.

As a second case, consider the antiferromagnetic, one-dimensional, nearest-neighbor Ising model, which is similar enough that it makes for a useful contrast; see Fig. 15. The finite-temperature and high-temperature limits are identical to those in the ferromagnetic case, but the low-temperature case differs. At $T = 0$ the spin system forms a perfect crystal of alternating spins and so one must make a single observation to know in which spatial-phase the crystal is. Then the entire structure is known exactly. Thus, the Markov order is $R = 1$. This situation is not a broken symmetry as in ferromagnetic low-temperature case. Even with a nonzero external field, an observation is still required to know in which causal state the system is.

Overall, now that we can directly determine intrinsic lengths in configurations, we see that the coupling range specified by a Hamiltonian need not be an intrinsic property of realized configurations. The simple extremes above make this easy to understand. At infinite temperature each system configuration is equally likely: the Hamiltonian range has no effect on which configurations are realized. At zero temperature only the ground states are expressed and these

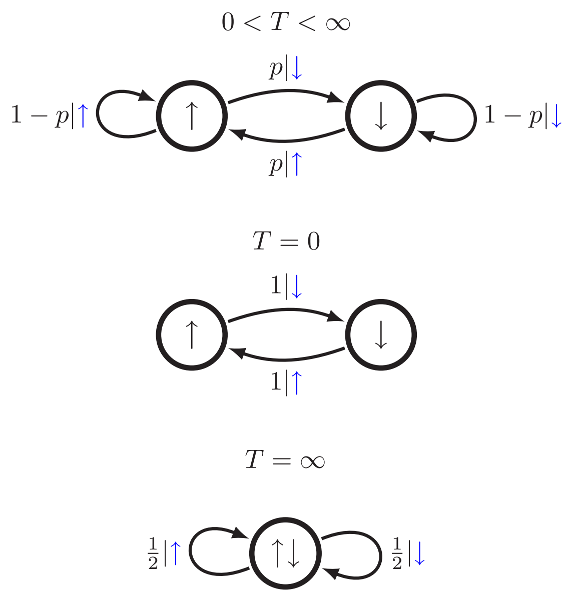

FIG. 15. (Color online) ϵ-Machines for a one-dimensional antiferromagnetic Ising model as a function of temperature T. Here $p = \frac{1}{2}(1 + \tanh\beta)$, the external field $B = 0$, and $J = k_B = 1$.

need not explore all the possible configurations allowed by the Hamiltonian. Both of these situations mask the coupling range specified by the Hamiltonian. Due to this, the Markov order R captures the effective coupling range and need not match that specified by the Hamiltonian.

In Ising spin chains, the cryptic order equals the Markov order. (This is due most directly to the fact that spin blocks are in one-to-one correspondence with the ϵ-machine causal states. In addition, one must add the caveat that the ϵ-machine be ergodic.) This equality need not be the case, however, even in simple physical systems. We note how restrictive Hamiltonian-specified dynamics are via two (again, 1D) examples of infinite Markov order, but finite cryptic order, that arise from finite specification. In the first class of systems, even though one starts with strictly local interactions—configurations with finite Markov order specified by a Hamiltonian with finite coupling range—a 1D system can anneal to one with effectively infinite-range interactions, as shown in Ref. [21]. (See the ϵ-machine in Fig. 2 there.) In this particular case, the annealed state is non-Markovian, exhibiting infinite-range structure and Markov order $R = \infty$. Notably, the annealed configurations for this example have finite cryptic order $k_\chi = 4$. For the second class of systems we just briefly note that these unusual length-scale properties are not restricted to classical systems. They also arise in quantum systems. See the analyses in Refs. [22] and [23].

Finally, since the results here emphasize properties intrinsic to realized configurations, let us turn the question around. Given a single typical instance from the ensemble of allowed configurations, how much can be inferred about the Hamiltonian? Though the topological techniques described above do not provide coupling amplitudes and the like, they do give the maximum range of effective interactions. What does one do, though, without a Hamiltonian or some other system specification? It turns out that a variety of methods exist for inferring hidden Markov models from a sample. And, since any hidden Markov model can be converted to an ϵ-machine [16], from there the Markov and cryptic orders can be directly computed. And so the above methods can be applied to a wide range of theoretically modeled or experimentally realized physical systems.

IX. CONCLUSION

We began by defining two different measures of memory in complex systems. The first, the Markov order R, is the length of time one must observe a system in order to make accurate predictions of its behavior. The second, the cryptic order k_χ, quantifies the ability to retrodict a system's internal dynamics. We showed that despite their statistical nature, these time scales are topological properties—properties of the synchronizing words of a process's ϵ-machine.

We demonstrated how to compute these length scales for hidden Markov models, most of which can be motivated in terms of the synchronization properties of the underlying process. Interestingly, we found that one of the most fundamental and important properties—the Markov order R—is computable using *only* the process's ϵ-machine (Proposition 2). When calculated with non-ϵ-machines, the algorithms yield related quantities, such as the synchronization order. For more details, see the appendices. In addition, the ϵ-machine provides an exact method for computing the cryptic order. From these results, we constructed very efficient algorithms for their calculation.

In the empirical setting, we now see that one should first infer the ϵ-machine and then, from it, calculate the Markov and cryptic orders. There are a number of methods of inferring an ϵ-machine from data (e.g., Ref. [24] and citations therein). In the theoretical setting, given some formal description of a process—such as a Hamiltonian or general hidden Markov model—one can analytically calculate a process's ϵ-machine. In any case, as soon as one has the ϵ-machine the preceding gives exact results.

To appreciate what is typical about these length scales, we surveyed the range of Markov and cryptic orders in the space of all structured binary processes represented by ϵ-machines with six states. The main result was rather surprising: Infinite Markov and cryptic orders dominate. Thus, the topological analysis leads one to conclude that synchronization, even to finite-state stochastic processes, can be generically difficult. However, from a probabilistic view it is exponentially fast [7,25]. A way to resolve this seeming contradiction is to conjecture that the topological properties are driven by sequences whose relative proportion vanishes with increasing length. The survey also revealed a variety of interesting ancillary properties that pose a number of open questions, presumably combinatoric and group theoretic in nature.

We closed analyzing the role these scales play in classical (and briefly quantum) spin systems, drawing out the physical interpretations. We emphasized, in particular, the difference between the interaction range specified by a Hamiltonian and the effective range of correlation in realized spin configurations. This led us to propose calculating the orders to put constraints on spin systems whose Hamiltonians are unknown.

Finally, appendices prove the key claims above, discuss other related measures of synchronization and length scales, and provide step-by-step details for each algorithm.

ACKNOWLEDGMENTS

We thank Ben Johnson, Paul Riechers, and Nick Travers for many invaluable discussions. This work was partially supported by ARO Grants No. W911NF-12-1-0234 and No. W911NF-12-1-0288 and the Defense Advanced Research Projects Agency (DARPA) Physical Intelligence project. The views, opinions, and findings contained here are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of ARO, DARPA, or the Department of Defense.

APPENDIX A: DEFINITIONS

Here we provide additional results on length scales and synchronization and prove a number of claims made in the main text. First, we lay out the definitions needed and then give several key results that follow. Building on these, we delineate the central algorithms and conclude with a discussion of companion length scales.

1. Minimal synchronizing words

For the synchronization problem, we consider an observer who begins with a correct model (a presentation) of a process. The observer, however, has no knowledge of the process's internal state. The challenge is to analyze how an observer's knowledge of the internal state changes as more and more measurements are observed.

At first glance, one might say that the observer's knowledge should increase with additional measurements, corresponding to a monotonically decreasing state uncertainty, but this is generically not true. In fact, it is possible for the observer's knowledge (measured in bits) to oscillate with each new measurement. The crux of the issue is that additional measurements are being used to inquire about the *current* state rather than the state at some fixed moment in time.

It is helpful to identify the set of words that take the observer from the condition of total ignorance to exactly knowing a process's state. First, we introduce what we mean by synchronization in terms of lack of state uncertainty. Second, we define the set of minimal synchronizing words.

Definition 1. A word w of length L is *synchronizing* if the Shannon entropy over the internal state, conditioned on w, is zero,

$$\text{Sync}(w) \Leftrightarrow H[\mathcal{S}_\ell | X_{0:\ell} = w] = 0, \tag{A1}$$

where $\text{Sync}(w)$ is a Boolean function.

Definition 2. A presentation's set of *minimal synchronizing words* is the set of synchronizing words that have no synchronizing prefix,

$$\mathcal{L}_{\text{sync}} \equiv \{w | \text{Sync}(w) \text{ and } \neg\text{Sync}(u) \text{ for all } u{:}w = uv\}.$$

Remark. $\mathcal{L}_{\text{sync}}$ is a prefix-free, regular language. If each word is associated with its probability of being observed, we obtain a prefix-free code encoding each path to synchrony—a word in $\mathcal{L}_{\text{sync}}$—with the associated probability of synchronizing via that path. These codes are generally nonoptimal in the familiar information-theoretic sense.

2. Synchronization order

According to Sec. A 1, one is synchronized to a process's presentation after seeing word w if there is complete certainty in the state. We now expand this view slightly to ask about synchronization over all words of a particular length. Equivalently, we examine synchronization to an ensemble of process realizations.

Definition 3. The *synchronization order* $k_{\mathbf{S}}$ [5] is the minimum length for which every allowed word is a synchronizing word:

$$k_{\mathbf{S}} \equiv \min\{\ell | H[\mathcal{S}_\ell | X_{0:\ell}] = 0\}. \tag{A2}$$

As for the Markov and cryptic orders, $k_{\mathbf{S}}$ is considered ∞ when the condition does not hold for any finite ℓ.

APPENDIX B: RESULTS

We now provide several results related to these length scales that shed light on their nature, introducing connections and simplifications that make their computation tractable.

Proposition 1. The synchronization order is as follows:

$$k_{\mathbf{S}} = \max\{R, k_\chi\}. \tag{B1}$$

Proof. First, note the following:

$$H[\mathcal{S}_\ell | X_{0:\ell}] = H[X_{0:\ell}, \mathcal{S}_\ell] - H[X_{0:\ell}]. \tag{B2}$$

Since the block-state entropy upper bounds the block entropy, the conditional entropy above can only reach its asymptotic value once both terms have individually reached their asymptotic behavior. The latter are controlled by k_χ and R, respectively. ■

This result reduces the apparent diversity of length scales, eventually allowing one to calculate the Markov order via the synchronization order, which itself is directly computable.

Proposition 2. For ϵ-machines,

$$R = k_{\mathbf{S}}. \tag{B3}$$

Proof. Applying the causal equivalence relation $\sim_\epsilon$ to Definition 1 we find

$$\Pr(X_{0:} | X_{:0}) = \Pr(X_{0:} | X_{-\ell:0}) \Rightarrow X_{:0} \sim_\epsilon X_{-R:0}. \tag{B4}$$

This further implies that the causal states $\boldsymbol{\mathcal{S}}$ are completely determined by $X_{-R:0}$,

$$H[\mathcal{S}_0 | X_{-R:0}] = 0. \tag{B5}$$

This statement is equivalent to the Markov criterion. ■

Remark. This provides an alternate proof that the cryptic order k_χ is bounded above by the Markov order R in an ϵ-machine via a simple shift in indices as follows:

$$H[\mathcal{S}_0 | X_{-R:0}] = 0, \tag{B6}$$

$$\Rightarrow H[\mathcal{S}_R | X_{0:R}] = 0, \tag{B7}$$

$$\Rightarrow H[\mathcal{S}_R | X_{0:}] = 0. \tag{B8}$$

This proposition gives indirect access to the Markov order via a particular presentation—the ϵ-machine. Since the Markov order is not defined as a property of a presentation it would generally be unobtainable, but due to unique properties of the ϵ-machine, it can be accessed through the synchronization order.

There is a subclass of ϵ-machines to which one synchronizes in finite time; these are the *exact* ϵ-machines of Ref. [7].

Proposition 3. Given an exact ϵ-machine with finite Markov order R, the subshift of finite type that underlies it has a "step" [11] equal to R.

Corollary 1. Given an exactly synchronizing ϵ-machine, the underlying sofic system is a subshift of finite type if and only if R is finite.

Remark. A process with infinite Markov order can have a presentation whose underlying sofic system is a subshift of finite type.

These results draw out a connection with length scales of sofic systems from symbolic dynamics [11]. Subshifts of finite type have a probability-agnostic length scale analog of the Markov order known as the "step." In the case of ϵ-machine presentations, they are in fact equal.

We will now prove that two of the lengths defined—the cryptic and synchronization orders—are topological. That is, they are properties of the presentation's graph topology and are independent of transition probabilities, so long as changes to the probabilities do not remove transitions and do not cause states to merge. Additionally, due to Proposition 2, the Markov order is topological. All three are topological since they depend only on the length at which a conditional entropy vanishes, not on how it vanishes.

Theorem 1. Synchronization order $k_{\mathbf{S}}$ is a topological property of a presentation.

Proof. Beginning from Definition 3, there is length $\ell = k_{\mathbf{S}}$ at which

$$H[\mathcal{S}_\ell | X_{0:\ell}] = \sum_{w \in \mathcal{A}^\ell} \Pr(w) H[\mathcal{S}_\ell | X_{0:\ell} = w] = 0.$$

Thus, $H[\mathcal{S}_\ell | X_{0:\ell} = w] = 0$ for all $w \in \mathcal{A}^\ell$, the set of length-ℓ words with positive probability. Since every word of length ℓ is synchronizing, ℓ is certainly greater than the synchronization order. As synchronizing words are synchronizing regardless of their probability of occurring, the synchronization order $k_{\mathbf{S}}$ is topological. ■

Corollary 2. Markov order R is a topological property of an ϵ-machine.

Proof. Since $k_{\mathbf{S}}$ is a topological property by Theorem 1 and since an ϵ-machine's $R = k_{\mathbf{S}}$ by Proposition 2, the Markov order is topological. ■

Theorem 2. Cryptic order k_χ is a topological property of a presentation.

Proof. Beginning from Definition 3, there is a length $\ell = k_\chi$ at which

$$\begin{aligned} 0 &= H[\mathcal{S}_\ell | X_{0:\ell}] \\ &\overset{(1)}{=} \sum_{x_{0:} \in \mathcal{A}^\infty} \Pr(x_{0:}) H[\mathcal{S}_\ell | X_{0:} = x_{0:}] \\ &\overset{(2)}{=} \sum_{w \in \mathcal{L}_{\text{sync}}} \Pr(w, \sigma_w) H[\mathcal{S}_\ell | X_{0:|w|} = w, \mathcal{S}_{|w|} = \sigma_w]. \end{aligned}$$

Here step (1) simply expands the conditional entropy. Step (2) is true provided that the sum is over minimal synchronizing words and σ_w is the state to which one synchronizes via w. This final sum is zero only if the sum vanishes term-by-term. Thus, given a word that synchronizes and the state to which it synchronizes, each term provides a *cryptic-order candidate*—the number of states that could not be retrodicted from that state and word. Finally, the longest such cryptic order candidate is the cryptic order for the presentation. ■

Restated, the cryptic order k_χ is topological as it depends only on the minimal synchronizing words, which are topological by definition.

APPENDIX C: ALGORITHMS

We are now ready to turn to computing the various synchronization length scales given a presentation. While all of the algorithms to follow have compute times that are exponential in the number of machine states, we find them to be very efficient in practice. This is particularly the case when compared to naïve algorithms to compute these properties. For example, computing synchronization, Markov, or cryptic orders by testing successively longer blocks of symbols is exponential in the length of the longest block tested. Worse, in the case of non-Markovian and ∞-cryptic processes the naïve algorithm will not halt. In addition, the naïve implementation of Theorem 2 given in the proof to compute the cryptic order has a compute time of $O(2^{2^N})$, whereas the one presented below is a simple exponential of N.

Unsurprisingly, given the results provided in Appendix B, we begin with the minimal synchronizing words as they are the underpinnings of the synchronization and cryptic orders. The algorithms make use of standard procedures. Most textbooks on algorithms provide the necessary background; see, for example, Ref. [18].

1. Minimal Synchronizing words

We construct a deterministic finite automaton (DFA) that recognizes $\mathcal{L}_{\text{sync}}$ of a given presentation $\mathcal{M} = (Q, E)$, where Q are the states and E are the edges. This is done as follows.

Algorithm 1.

(1) Begin with the recurrent presentation $\mathcal{M}$.

(2) Construct $\mathcal{M}$'s power automaton $2^{\mathcal{M}}$, producing a DFA $\mathcal{T} = 2^{\mathcal{M}}$.

(3) Set the node in $\mathcal{T}$ that corresponds to all $\mathcal{M}$'s states as $\mathcal{T}$'s start state.

(4) Remove all edges between singleton states of $\mathcal{T}$. (These are the edges from $\mathcal{M}$.)

(5) Set all singleton states of $\mathcal{T}$ as accepting states.

Now enumerate $\mathcal{L}_{\text{sync}}$ via an ordered breadth-first traversal of $\mathcal{T}$ and output each accepted word.

2. Synchronization order

Thanks to Eq. (1) we see that $k_{\mathbf{S}}$ is the shortest length ℓ that encompasses all of $\mathcal{L}_{\text{sync}}$. This is, trivially, the longest word in $\mathcal{L}_{\text{sync}}$. With this, computing the synchronization order reduces to the following.

Algorithm 2.

(1) If $\mathcal{L}_{\text{sync}}$ is infinite, return ∞.

(2) Determine the longest word w in $\mathcal{L}_{\text{sync}}$, then $k_{\mathbf{S}} = |w|$.

The test in the first step can be done simply by running a loop-detection algorithm on DFA $\mathcal{T}$. If there is a loop, then $\mathcal{L}_{\text{sync}}$ is infinite. The second step is quickly performed by using the Bellman-Ford or Floyd-Warshall algorithms.

3. Markov order

Due to Theorem 2, a process's Markov order can be computed by finding the synchronization order of the process's ϵ-machine. If one does not have the ϵ-machine for a process, but rather some other unifilar presentation, it is still possible in some cases to obtain the Markov order through the synchronization order. That is, the algorithms for $k_{\mathbf{S}}$ and k_χ provide probes into the presentation's length scales. It can be the case that R is accessible to those probes, if $k_\chi < k_{\mathbf{S}}$, but it is only guaranteed to be accessible in the case of ϵ-machines. Note that there exist techniques for constructing the ϵ-machine from any presentation [16].

4. Cryptic order

In the following algorithm $\mathcal{T}$ refers to the power automaton of the machine $\mathcal{M}$. $\mathcal{T}$'s states—p, q, and r—are elements of the power set of the states of $\mathcal{M}$. By the *predecessors* of a state q along edge $p \xrightarrow{x} q$ we refer to the set $p' = \{m | (m \xrightarrow{x} n) \in \mathcal{M} \text{ and } m \in p \text{ and } n \in q\}$. These are the states $m \in p$ that actually transition to a state $n \in q$ on symbol x. By *subset construction* below we refer to the standard NFA-to-DFA conversion algorithm [9].

Algorithm 3.

(1) Construct the power automaton $\mathcal{T} = 2^{\mathcal{M}}$ via subset construction.

(2) Push each edge $p \xrightarrow{x} q$ in $\mathcal{T}$ to a queue.

(3) While queue is not empty:

(a) Pop edge $p \xrightarrow{x} q$ in the queue.

(b) If edge is in processed list:

(i) Restart loop, popping the next edge from the queue.

(c) Find the predecessors p' of q along $p \xrightarrow{x} q$.

(d) If $p' \neq p$:

(i) Remove edge $p \xrightarrow{x} q$ from $\mathcal{T}$.

(e) If $|p'| > 1$:

(i) Perform subset construction on p' (implicitly, this adds the edge $p' \xrightarrow{x} q$ to $\mathcal{T}$).

(ii) Push each edge created in the prior step into the queue.

(iii) For each $r \xrightarrow{y} p$ in $\mathcal{T}$:

(A) Add edge $r \xrightarrow{y} p'$ to $\mathcal{T}$.

(B) Add edge $r \xrightarrow{y} p'$ to the queue.

(iv) Add $p \xrightarrow{x} q$ and $p' \xrightarrow{x} q$ to the processed list.

The result is an automaton $\mathcal{T}'$. The longest path in $\mathcal{T}'$ through transient states ending in a recurrent state is the cryptic order. Skipping previously processed edges is important since for some topologies the algorithm can enter a cycle where it will remove and then later add the same edge *ad infinitum*.

There are three simple additions to this algorithm that result in a sizable decrease in running time. The first is to store the edges to be processed in a priority queue, such that an edge $p \xrightarrow{x} q$ is popped before an edge $r \xrightarrow{y} s$ if $|q| < |s|$, or if $|q| = |s|$, then pop if $|p| < |r|$. The second optimization is to trim dangling states after each pass through the outer loop. A dangling state is a state p such that there is no path from p to the recurrent states. The last method for improving speed is to not add edges between recurrent states to the queue in step (2).

This algorithm for computing the cryptic order only holds for unifilar presentations.

APPENDIX D: OTHER NATURAL TIME SCALES

Paralleling the interpretation of the Markov and cryptic orders as the block lengths at which an associated information measure reaches its asymptotic behavior, this section briefly defines several new time scales associated with the multivariate information measures recently introduced in Ref. [6] to dissect the information in a single measurement.

The first order k_{I} is the length at which the multivariate mutual information $\mathrm{I}[X_0; X_1; \ldots; X_{N-1}]$ reaches its asymptotic behavior. Unfortunately, no bounds are known for this order.

The next collection of time scales—denoted k_{R}, k_{B}, k_{Q}, and k_{W}—are the lengths at which the ephemeral information r_μ, bound information b_μ, enigmatic information q_μ, and local exogenous information each reach their respective asymptotes [6]. Furthermore, these four orders are equal, due to the linear interdependence of their respective measures. It turns out that there are lower and upper bounds for these with respect to the Markov order, which can be easily explained. Consider Fig. 8 in Ref. [6]: By definition $H[X_{:0}]$ can be replaced with $H[X_{-R:0}]$ and, if the process is stationary, $H[X_{1:}]$ with $H[X_{1:R+1}]$. It is therefore reasonable that one requires at least R symbols and most $2R$ symbols to accurately dissect $H[X_0]$. In fact, numerical surveys that we have carried out agree with these limits.

Finally, a sequel analyzes the elusive information σ_μ, showing that the Markov order R equals the length k_σ at which the present measurement block $X_{0:\ell}$ renders the past and future conditionally independent.

While we have defined these orders and provided bounds, it remains to be seen if there exist efficient methods to calculate them, let alone topological interpretations for each.

[1] L. E. Reichl, *A Modern Course in Statistical Mechanics* (University of Texas Press, Austin, Texas, 1980).

[2] A. S. Weigend and N. A. Gershenfeld, editors. *Time Series Prediction: Forecasting the Future and Understanding the Past*, SFI Studies in the Sciences of Complexity (Addison-Wesley, Reading, MA, 1993).

[3] A. Raj and A. van Oudenaarden, Cell **135**, 216 (2008).

[4] J. P. Crutchfield and D. P. Feldman, Adv. Complex Sys. **04**, 251 (2001).

[5] J. P. Crutchfield, C. J. Ellison, R. G. James, and J. R. Mahoney, Chaos **20**, 037105 (2010).

[6] R. G. James, C. J. Ellison, and J. P. Crutchfield, Chaos **21**, 037109 (2011).
[7] N. Travers and J. P. Crutchfield, J. Stat. Phys. **145**, 1181 (2011).
[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. (Wiley-Interscience, New York, 2006).
[9] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed. (Prentice-Hall, New York, 2006).
[10] J. P. Crutchfield, Nat. Phys. **8**, 17 (2012).
[11] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding* (Cambridge University Press, Cambridge, 1999).
[12] Unifilar is known as "deterministic" in the finite automata literature. Here we avoid the latter term so confusion does not arise due to the stochastic nature of the models being used. It is referred to as "right-resolving" in symbolic dynamics [11].
[13] J. R. Mahoney, C. J. Ellison, and J. P. Crutchfield, J. Phys. A: Math. Theor. **42**, 362002 (2009).
[14] J. P. Crutchfield and D. P. Feldman, Chaos **13**, 25 (2003).
[15] C. R. Shalizi and J. P. Crutchfield, J. Stat. Phys. **104**, 817 (2001).
[16] C. J. Ellison, J. R. Mahoney, and J. P. Crutchfield, J. Stat. Phys. **136**, 1005 (2009).
[17] Robert B. Ash. *Information Theory* (Dover, London, 1990).
[18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, 3rd ed. (The MIT Press, Cambridge, MA, 2009).
[19] B. D. Johnson, J. P. Crutchfield, C. J. Ellison, and C. S. Mctague. arXiv:1011.0036.
[20] D. P. Feldman and J. P. Crutchfield. Santa Fe Institute Working Paper 98-04-026, 1998.
[21] D. P. Varn and J. P. Crutchfield, Phys. Lett. A **324**, 299 (2004).
[22] K. Wiesner and J. P. Crutchfield, Physica D **237**, 1173 (2008).
[23] K. Wiesner and J. P. Crutchfield, in *Unconventional Computation*, Lecture Notes in Computer Science Vol. 4618 (Springer, New York, 2007), pp. 214–225.
[24] C. C. Strelioff and J. P. Crutchfield, arXiv:1309.1392 [Phys. Rev. E (to be published)].
[25] N. Travers and J. P. Crutchfield, J. Stat. Phys. **145**, 1202 (2011).