

Inference of time-ordered multibody interactions

Unai Alvarez-Rodriguez ^{1,2} Luka V. Petrović ² and Ingo Scholtes ^{3,2}

¹*University of Deusto, 48007 Bilbao, Spain*

²*University of Zurich, CH-8006 Zürich, Switzerland*

³*Julius-Maximilians-Universität Würzburg, 97070 Würzburg, Germany*



(Received 11 July 2023; accepted 4 August 2023; published 27 September 2023)

We introduce time-ordered multibody interactions to describe complex systems manifesting temporal as well as multibody dependencies. First, we show how the dynamics of multivariate Markov chains can be decomposed in ensembles of time-ordered multibody interactions. Then, we present an algorithm to extract those interactions from data capturing the system-level dynamics of node states and a measure to characterize the complexity of interaction ensembles. Finally, we experimentally validate the robustness of our algorithm against statistical errors and its efficiency at inferring parsimonious interaction ensembles.

DOI: [10.1103/PhysRevE.108.034312](https://doi.org/10.1103/PhysRevE.108.034312)

I. INTRODUCTION

Earth's climate, traffic in large cities, or a baroque musical composition are all examples of complex systems. They are composed of multiple elements with different types of interactions and elusive dynamical laws [1–3]. Efforts to understand those laws have prompted considerable advances in network science during the last few decades. The journey started with models for complex networks with a single type of dyadic links [4–6], continued with multilayer and multiplex networks [7,8], and has most recently brought us different approaches to model higher-order interactions in complex systems [9–17]. Over time, we have learned to impose weaker assumptions upon our models and thus to describe more complex interactions.

Lately, the community has focused on the study of two types of interactions: multibody interactions and time-ordered interactions. The former are modeled with hypergraphs and the latter are modeled with higher-order Markov chains. Each of them violates a different assumption of standard network models: multibody interactions violate the assumption that system dynamics can be explained solely with dyadic interactions; time-ordered interactions violate the Markov assumption [18,19]. However, real systems may violate both assumptions, invalidating either modeling approach. This situation leads to multiple open questions. How can we analyze systems that exhibit interactions that are both time ordered and multibody? More specifically, how can we formalize such interactions and infer them from data? How can we use time-ordered multibody interactions to explain system dynamics?

The multibody facet of the problem has already been solved with the inference of multibody interactions from dynamical equations [20] and with information theory tools [21–23]. There are also preliminary efforts to unify the modeling of time-ordered and multibody interactions, e.g., in the study of synchronization [24], contagion [25], and consensus dynamics [26]. However, as discussed in [17], a general formalism is yet to be found.

In this manuscript, we propose a unified methodology for scenarios in which multibody and time-ordered dependencies coexist. First, we introduce parsimonious models of time-ordered multibody interactions. Then, we present an algorithm to decompose the system dynamics into an ensemble of interactions. Last, we introduce a measure to characterize the complexity of such ensembles of interactions. We show how the integration of time-ordered and multibody interactions enables a better description of real world systems than current modeling paradigms.

II. TIME-ORDERED MULTIBODY INTERACTIONS

A. Notation

Let S be a closed system of N nodes $n \in \mathcal{N}$, $\mathcal{N} = \{n\}_0^{N-1}$. Each node n is constrained to a finite set of states, called alphabet X_n with cardinality $|X_n|$. We denote the state of each node n at a given time t with $s_n(t)$, and the state of the whole system with $\mathbf{s}(t) = (s_n(t))_{n \in \mathcal{N}}$. Thus, the alphabet of the whole system is $X = \times_{n \in \mathcal{N}} X_n$, where \times denotes Cartesian product. We observe the system in an interval (t_0, t_{end}) and collect sequences of system states $\mathbf{D} = (\mathbf{s}(\tau))_{\tau=t_0}^{t_{\text{end}}}$ and element states $\mathbf{D}_n = (s_n(\tau))_{\tau=t_0}^{t_{\text{end}}}$.

We can model the dynamics of S as a higher-order Markov chain with memory m : the transition probability to a state $x \in X$ depends on the previous m states $\bar{x} \in |X|^m$. The transition probabilities $\pi(x|\bar{x})$ are encoded in matrix elements $T_{x\bar{x}}$ of transition matrices \mathbf{T} with dimensions $|X| \times |X|^m$. One can infer the transition matrix \mathbf{T} for different Markov orders m using the temporal data \mathbf{D} , and select the Markov order that provides the best predictability-parsimony balance [27–29]. See Fig. 1 for an example of our notation.

Higher-order Markov chains model the dynamics of S as a function of all nodes at all times and thus one cannot analyze node interdependencies in isolation. Therefore, we deviate from this modeling approach, and define time-ordered

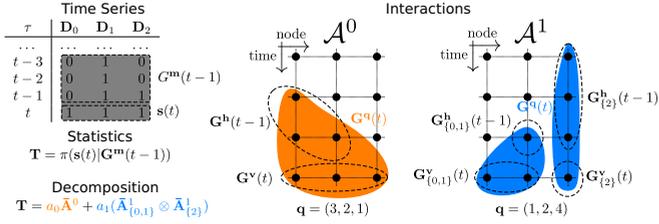


FIG. 1. We are presented the task to explain the dynamics of a system of $N = 3$ nodes from their temporal data \mathbf{D} . We first compute a higher-order Markov chain of memory $m = 3$ and collect the transition probabilities in a matrix \mathbf{T} , which predicts the joint node state $\mathbf{s}(t)$ as a function of the previous three states $\mathbf{G}^m(t-1)$. We then decompose \mathbf{T} as a convex sum of time-ordered multibody interactions \mathcal{A}^i . An interaction of type \mathbf{q} models the dynamics as a function of the group of system variables $\mathbf{G}^q(t)$, which contains q_n variables of each node n . $\mathbf{G}^q(t)$ is divided in $\mathbf{G}^v(t)$ and $\mathbf{G}^h(t-1)$ that account, respectively, for row and column variables of interaction transition matrices $\bar{\mathbf{A}}^i$. The interaction \mathcal{A}^1 is separated into two dynamically independent subsystem interactions: $\mathcal{A}^1_{\{0,1\}}$ and $\mathcal{A}^1_{\{2\}}$.

multibody interactions that depend on the last q_n states of every node n . We denote a group of q_n successive temporal states of node n with tuples $\mathbf{G}_n^{q_n}(t) := (s_n(\tau))_{\tau=t-q_n+1}^t$. If q_n is zero, $\mathbf{G}_n^{q_n}(t)$ denotes an empty tuple. We then construct a vector $\mathbf{q} = (q_n)_{n \in \mathcal{N}}$ and denote groups of variables for the whole system $\mathbf{G}^q(t) := (\mathbf{G}_n^{q_n}(t))_{n \in \mathcal{N}}$. States of $\mathbf{G}^q(t)$ take values in alphabet $X^q := \times_{n \in \mathcal{N}} X_n^{q_n}$. Before defining time-ordered multibody interactions, we emphasize that in our notation Markov chains with memory m model $\mathbf{s}(t) = \mathbf{G}^1(t)$ given the history $\mathbf{G}^m(t-1)$, where $\mathbf{1}$ and \mathbf{m} are vectors with all components 1 and m , respectively. The transition probabilities then depend on $o := m + 1$ system states, amounting to a total oN time variables $\mathbf{G}^o(t)$, where \mathbf{o} is a vector with all components o . See Fig. 1.

B. Interactions

The core of our formalism are time-ordered multibody interactions, which depend only on a subset of the $\mathbf{G}^o(t)$ variables of higher-order Markov chains. A time-ordered multibody interaction \mathcal{A} of type \mathbf{q} , $0 \leq q_n \leq o$, models the probability of $\mathbf{s}(t)$ given history $\mathbf{G}^m(t-1)$ as a function of variables $\mathbf{G}^q(t)$ and independent of other variables. We separate the variables $\mathbf{G}^q(t)$ in two groups: variables $\mathbf{G}^v(t)$ describing nodes at time t and variables $\mathbf{G}^h(t-1)$ describing their history. Here, the components of \mathbf{v} are $v_n = 0$ when $q_n = 0$, and $v_n = 1$ when $q_n > 0$; the vector \mathbf{h} contains history lengths $\mathbf{h} = \mathbf{q} - \mathbf{v}$. See Fig. 1 for an example of an interaction ensemble with two interactions in a system of three nodes, and corresponding $\mathbf{G}^v(t)$ and $\mathbf{G}^h(t-1)$. The variables $\mathbf{G}^v(t)$ are modeled with a multinomial distribution that depends on the history $\mathbf{G}^h(t-1)$. The state of $\mathbf{G}^v(t)$ takes values $y \in X^v$, and the state of $\mathbf{G}^h(t-1)$ takes values $\bar{y} \in X^h$. The transition probabilities from \bar{y} to y are encoded in the transition matrix $A_{y\bar{y}}$ of interaction \mathcal{A} . For nodes where $q_n = 0$, the probabilities of $s_n(t)$ do not depend on any variable, and thus they are uniform. In summary, an interaction \mathcal{A} models the

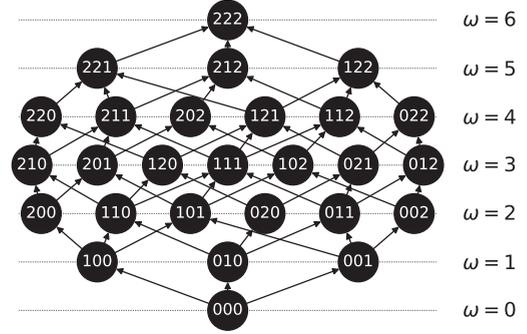


FIG. 2. Hasse diagram of time-ordered multibody interactions for a system of three nodes and memory equal to one. Interaction types are depicted as vertices \mathbf{q} , and arranged according to their interaction order ω . If there is a path from \mathbf{q} to \mathbf{q}' , then every interaction of type \mathbf{q} can be expressed as an interaction of type \mathbf{q}' .

dynamics as

$$\pi(\mathbf{s}(t)|\mathbf{G}^m(t-1), \mathcal{A}) = \frac{A_{y\bar{y}}}{\prod_{n \in \mathcal{N} | q_n=0} |X_n|}. \quad (1)$$

While the transition matrix \mathbf{T} has $|X| \cdot |X|^m = \prod_n |X_n|^o$ elements, the interaction \mathcal{A} of type \mathbf{q} has only $|X^v| \cdot |X^h| = \prod_{n \in \mathcal{N}} |X_n|^{q_n}$ parameters. Therefore, interactions produce a parsimonious model of the system dynamics. With this, we have formally introduced time-ordered multibody interactions, and we now discuss how we can use them to decompose the dynamics of S .

C. Interaction hierarchy

The aim of our decomposition is to explain the dynamics in terms of the simplest possible interactions. Therefore, we organize time-ordered multibody interactions in a hierarchy by their complexity. We define the interaction order ω of interaction type \mathbf{q} as $\omega = \sum_{n \in \mathcal{N}} q_n$, that counts the number of variables on which the interaction depends. Moreover, interactions are nested models: an interaction \mathcal{A} of type \mathbf{q} can be nested in (or represented with) an interaction \mathcal{A}' of type \mathbf{q}' if $\forall n : q_n \leq q'_n$. Intuitively, if variables of \mathcal{A} are a subset of variables of \mathcal{A}' , \mathcal{A} can be nested in \mathcal{A}' . The relation of nestedness over the set of interaction types naturally defines a partial order, which is commonly depicted with a Hasse diagram. In the Hasse diagram on Fig 2, two interaction types are connected as long as they are nested and separated by one interaction order ω . For example, an interaction of type $\mathbf{q} = (2, 1, 0)$ (leftmost type in fourth row of Fig. 2) models the dynamics of S as a function of $s_0(t), s_0(t-1)$, and $s_1(t)$. Thus, it can be nested in any interaction type that includes these variables. The interaction type $\mathbf{q} = \mathbf{o}$, which corresponds to a Markov chain on S , is particularly important for the decomposition because any other type of interaction can be nested in it. When we nest an interaction \mathcal{A} in type \mathbf{o} , we denote the resulting interaction with $\bar{\mathcal{A}}$ and its transition matrix with $\bar{\mathbf{A}}$.

D. Decomposition algorithm

The decomposition algorithm has two steps: in a first step we decompose the transition matrix \mathbf{T} into an ensemble of

time-ordered interactions; in a second step, we decompose the time-ordered interactions into subsystem interactions [30]. Formally, in the first part we find the interaction coefficients a_i and matrices \mathbf{A}^i (here, i is an upper index) such that the matrix \mathbf{T} can be expanded as a convex sum: $\mathbf{T} = \sum_i a_i \mathbf{A}^i$. The algorithm repeatedly extracts interactions from \mathbf{T} . Starting at $\omega = 0$, it explores all interactions at ω before proceeding with $\omega + 1$ and extracts the one with a highest coefficient. Let us assume we have already extracted i interactions from \mathbf{T} and that $\tilde{\mathbf{T}}$ is the remainder: $\tilde{\mathbf{T}} = \mathbf{T} - \sum_{j=0}^{i-1} a_j \mathbf{A}^j$. For the next interaction \mathcal{A}^i , the algorithm finds the transition matrix \mathbf{A}^i such that the corresponding coefficient a_i is maximal and $\tilde{\mathbf{T}} - a_i \mathbf{A}^i$ does not have negative elements. The first part finishes when either $\tilde{\mathbf{T}}$ is a matrix of zeros, or, equivalently, when the sum of the interaction coefficients is one. See Appendix A 5 for the details.

The second step of the algorithm identifies statistically independent subsystems and decomposes the interactions into subsystem interactions. Formally, subsystem interactions are interactions defined on a subset of nodes. Let $\mathcal{P}(\mathcal{N})$ be a partition of \mathcal{N} , and $\mathcal{B} \in \mathcal{P}(\mathcal{N})$ be a subset of nodes $\mathcal{B} \subset \mathcal{N}$. If x is a state of nodes \mathcal{N} and \bar{x} is the history of the state, we denote the corresponding state of the nodes \mathcal{B} with $z_{\mathcal{B}}$ and their corresponding history $\bar{z}_{\mathcal{B}}$. We say that interaction \mathcal{A} has a subsystem interaction partition $\mathcal{P}(\mathcal{N})$ if

$$\pi(x|\bar{x}, \mathcal{A}) = \prod_{\mathcal{B} \in \mathcal{P}(\mathcal{N})} \pi(z_{\mathcal{B}}|\bar{z}_{\mathcal{B}}, \mathcal{A}_{\mathcal{B}}). \quad (2)$$

For instance, nodes with $q_n = 0$ are obvious examples of independent subsystem interactions.

We identify subsystem interactions from an interaction \mathcal{A} by iteratively factorizing its transition matrix as a tensor product. First, we factor out each node n that is modeled with a uniform distribution $q_n = 0$. We thus obtain the partition with one-element subsets for nodes n with $q_n = 0$, and a single subset containing the remaining nodes. Then, we iteratively search for two-group partitions of the subsets with more than one element. We stop when we cannot find a valid partition of any subset. See Appendix A 5 for the details on the subsystem interaction factorization, and see the SM [31] for a selection of practical examples of the full protocol including the Glauber dynamics [32].

E. Complexity

Having discussed the formalism of time-ordered multibody interactions, and the algorithm to extract them from data capturing the system-level dynamics of node states, we now focus on the complexity of interaction ensembles. We note that the expansion of the transition matrix \mathbf{T} is not unique, as it can be reconstructed with different ensembles of interactions and interaction coefficients. Considering Occam's razor, here we are interested in an ensemble with the lowest possible complexity. We thus need a principled way of measuring the complexity for different combinations of interactions and subsystem interactions.

To quantify the complexity of an ensemble of interactions, we introduce the measure of reducibility. Let us denote the number of degrees of freedom of an interaction \mathcal{A} with $\lambda(\mathcal{A})$. When interactions have no subsystem interactions, $\lambda(\mathcal{A})$ is

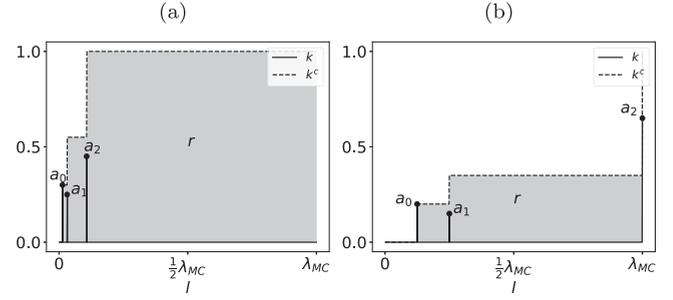


FIG. 3. Complexity measures $k(l)$, $k^c(l)$, and r for (a) high reducibility and (b) low reducibility systems. Here $\lambda(\mathcal{A})$ is the number of free parameters of an interaction, $k(l)$ weights how likely is to have an interaction with l parameters in an interaction ensemble, and $k^c(l)$ is the cumulative of $k(l)$. The reducibility r is the share of area under $k^c(l)$, and encodes the relative reduction in the average number of parameters per interaction of an ensemble. The ensemble in (a) has the same predictive power as \mathbf{T} even if its matrices contain 94% fewer parameters on average.

simply $(|X^v| - 1)|X^h|$ due to the normalization of stochastic matrices. If \mathcal{A} has subsystem interactions, $\lambda(\mathcal{A})$ is given by the sum of their parameters $\lambda(\mathcal{A}) = \sum_{\mathcal{B}} \lambda(\mathcal{A}_{\mathcal{B}})$. We call the ensemble reducible if simple interactions have a high contribution in the interaction ensemble. We call it irreducible if complex interactions have a high contribution in the interaction ensemble. Formally, reducibility r is defined as

$$r = \frac{\lambda_{\text{MC}} - \sum_i a_i \lambda(\mathcal{A}^i)}{\lambda_{\text{MC}}}, \quad (3)$$

where $\lambda_{\text{MC}} = (|X| - 1)|X|^m$ is the number of degrees of freedom of the Markov chain. Therefore, r accounts for the reduction in the average degrees of freedom when our interaction ensembles are used instead of Markov chains.

We showcase the power of this measure in the context of lossy compression. Let us denote with $k(l)$ the probability mass function of all interactions with l degrees of freedom: $k(l) = \sum_{\lambda(\mathcal{A}^i)=l} a_i$, and its cumulative distribution with $k^c(l)$. The value of $k^c(l)$ describes how well the interactions with less than l degrees of freedom represent the system dynamics. In Fig. 3 and Table I, we depict $k(l)$ and $k^c(l)$ of reducible and irreducible interaction ensembles. When the ensemble is reducible [panel (a) and Table I], it is dominated by the simple interactions, thus $k(l)$ peaks for small values of l , and $k^c(l)$ is high throughout the interval. When the ensemble is irreducible [panel (b)], $k(l)$ has peaks for large values of l ,

TABLE I. Interaction type \mathbf{q} , subsystem interaction partition \mathcal{B} , interaction order ω , matrix rows $|X^v|$; and columns $|X^h|$, number of parameters $\lambda(\mathcal{A})$, and coefficient a for the interaction ensemble of Fig. 3(a).

label	q	\mathcal{B}	ω	$ X^v $	$ X^h $	$\lambda(\mathcal{A})$	a
\mathcal{A}^0	210	{0, 1, 2}	3	4	2	6	0.3
\mathcal{A}^1	132	{0, 1}	4	4	4	12	0.25
		{2}	2	2	2	2	
\mathcal{A}^2	033	{0, 1, 2}	6	4	16	48	0.45

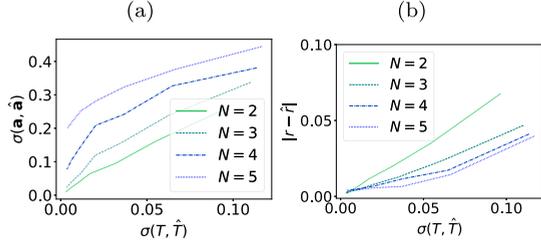


FIG. 4. We produce synthetic sequences and compute the statistical distance between the original transition matrices \mathbf{T} , (a) interactions \mathbf{a} and (b) reducibilities r , and the ones inferred from the data $\hat{\mathbf{T}}$, $\hat{\mathbf{a}}$, \hat{r} for a fixed alphabet $|X_n| = 2$ and order $o = 2$. Results reported here correspond to averages over 100 realizations.

and $k^c(l)$ is low throughout the interval. Therefore, when an ensemble is reducible, we can choose a small l where $k^c(l)$ is high. Thus, by only using the interactions with less than l degrees of freedom, we can compress a reducible ensemble with an approximate loss of $1 - k^c(l)$, as this accounts for the interactions that are above $\lambda = l$. Lastly, it is worth noting that the reducibility is the share of the area under $k^c(l)$.

III. VALIDATION

Finally, we present a series of experiments to validate different aspects of our contributions. In a first experiment, we establish a relation between the accuracy of our algorithm and the available amount of data. Our goal here is to understand how an error in \mathbf{T} propagates to errors in \mathbf{a} and r . We randomly generate \mathbf{T} by drawing its elements from a uniform distribution and obtain the ground truth values of interaction coefficients \mathbf{a} and reducibility r . Then, we construct random sequences out of \mathbf{T} , analyze them with our algorithm, and obtain estimators $\hat{\mathbf{T}}$, $\hat{\mathbf{a}}$, \hat{r} . We evaluate the accuracy of all three estimators in terms of the total variation distance σ between the ground truth and the estimated values.

The results are reported in Fig. 4, where we explore how the errors depend on the system size N for a system with all equal alphabets $|X_n| = 2$ and order $o = 2$ (see the SM [31] for a more general selection of parameters).

As noted above, given a dataset, there is not a unique interaction ensemble. This explains why in Fig. 4(a), inferred interactions deviate from the ground truth. However, and despite errors in inferring interactions, Fig. 4(b) reveals the capacity of the algorithm to explain the dynamics with an ensemble whose reducibility is close to the ground truth value. Therefore, the protocol is valid for obtaining simple and accurate interaction ensembles.

We perform a second experiment to evaluate the capacity of our algorithm to find high-reducibility decompositions. For this, we compare our protocol with an alternative decomposition algorithm that we call random algorithm (RA). In RA interactions are randomly selected from the interaction diagram. The experiment, Table II, shows that our algorithm consistently outperforms the random algorithm RA, and therefore validates our algorithm for inferring time-ordered multibody interactions for the first time.

TABLE II. Average values of reducibilities r_{RA} computed with a random algorithm and r computed with our method, over 1000 realizations. We decomposed the transition matrices of systems with parameters o , N , and $|X_n| = 2$.

(o, N)	(2,2)	(2,3)	(2,4)	(2,5)	(3,2)	(3,3)	(3,4)
r_{RA}	0.35	0.38	0.42	0.46	0.30	0.33	0.38
r	0.55	0.56	0.58	0.62	0.44	0.46	0.51

IV. CONCLUSION

In summary, we developed a framework for studying complex systems whose dynamics is ruled by time-ordered and multibody dependencies. The framework is based on interactions, which emerge from the expansion of time-evolution operators for multivariate higher-order Markov chains. We proposed a measure for the complexity of interaction ensembles and an algorithm to extract them from time series data. We believe these contributions to be a relevant asset for the field of complex systems as they address a currently latent problem by reconciling a data-oriented perspective with an analytical description of node interdependencies.

ACKNOWLEDGMENT

The authors acknowledge support from the Swiss National Science Foundation (SNSF) via Grant No. 176938. We also thank Vincenzo Perri for useful comments.

APPENDIX

1. Higher-order Markov chains

Higher-order Markov chains will be able to describe the dynamics of S as long as the following assumptions hold: (1) The same statistical behavior is expected at all points in time (causal stationarity). (2) No external variables can influence the dynamics of S (causal sufficiency) [3].

2. Nested interactions

An interaction \mathcal{A} of type \mathbf{q} is nested in an interaction \mathcal{A}' of type \mathbf{q}' by fixing the transition probabilities of \mathcal{A}' to those of \mathcal{A} :

$$\pi(\mathbf{G}^{\mathbf{v}'}(t)|\mathbf{G}^{\mathbf{h}'}(t-1)) = \frac{\pi(\mathbf{G}^{\mathbf{v}}(t)|\mathbf{G}^{\mathbf{h}}(t-1))}{\prod_{\substack{q'_n \neq 0 \\ q_n = 0}} |X_n|}. \quad (\text{A1})$$

3. Algorithm

We now explain the procedure to extract an interaction at interaction order ω . This procedure has been designed for nodes with all equal alphabets $X_n = X_{n'} \forall n, n' \in \mathcal{N}$. If nodes were different our algorithm would also yield a valid decomposition, but it should be modified for improved results.

At each round of the algorithm we extract the interaction \mathcal{A} with the highest coefficient a for a given interaction order ω . Therefore, we first explore all interactions \mathcal{A} of interaction order ω to obtain their interaction matrix \mathbf{A} . This matrix models transitions from histories $\bar{y} \in X^{\mathbf{h}}$ to states $y \in X^{\mathbf{v}}$.

Let us explain how to obtain an interaction matrix \mathbf{A} from $\tilde{\mathbf{T}}$: For every $y \in X^v$ and $\bar{y} \in X^h$, we create a set \mathcal{E} with matrix elements of $\tilde{\mathbf{T}}_{x\bar{x}}$. In order to decide which elements of $\tilde{\mathbf{T}}_{x\bar{x}}$ are in \mathcal{E} we define two functions $F_y(x)$ and $F_{\bar{y}}(\bar{x})$:

$$F_y(x) : X \rightarrow \{\text{True}, \text{False}\}, \quad F_{\bar{y}}(\bar{x}) : X^m \rightarrow \{\text{True}, \text{False}\}, \quad (\text{A2})$$

which characterize, respectively, the rows x and columns \bar{x} of $\tilde{\mathbf{T}}$ that should be included in \mathcal{E} .

$F_y(x)$ The function is computed as follows:

```

1: function  $F_{y,x}$ 
2:   for  $n \in \mathcal{N}$  do
3:     if  $y_n \neq \emptyset \wedge y_n \neq x_n$  then
4:       return False
5:     end if
6:   end for
7:   return True
8: end Function
    
```

If the component of node n of x and y coincide for all nodes in \mathcal{N} , with the exception of nodes not in y , the row has to be included in \mathcal{E} . If for any node the components of x and y are different, the function will output “False” before finishing the loop.

$F_{\bar{y}}(\bar{x})$ The function is computed as follows:

```

1: function  $F_{\bar{y},\bar{x}}$ 
2:   for  $n \in \mathcal{N}$  do
3:     for  $k \in \{0, \dots, m-1\}$  do
4:       if  $\bar{y}_{nk} \neq \emptyset \wedge \bar{y}_{nk} \neq \bar{x}_{nk}$  then
5:         return False
6:       end if
7:     end for
8:   end for
9:   return True
10: end Function
    
```

In this case, \bar{y} and \bar{x} are bidimensional tuples. A first dimension accounts for the node n , and the second dimension accounts for the memory. The idea is the same, we check coincidence for all nodes and for all memories between \bar{x} and \bar{y} , with the exception of nodes that are not in \bar{y} , or nodes whose memory is shorter than $m-1$.

\mathcal{E} contains those elements $\tilde{\mathbf{T}}_{x\bar{x}}$ that fulfill $F_y(x)$ and $F_{\bar{y}}(\bar{x})$. The minimum of such a set is the element $R_{y\bar{y}}$ of the auxiliary matrix \mathbf{R} :

$$R_{y\bar{y}} = \min \mathcal{E}. \quad (\text{A3})$$

This process is repeated for all y and \bar{y} to complete \mathbf{R} . We then obtain \mathbf{A} by normalizing \mathbf{R} columnwise, i.e.,

$$\mathbf{A}_{\bar{y}} = \frac{\mathbf{R}_{\bar{y}}}{\|\mathbf{R}_{\bar{y}}\|_1}, \quad (\text{A4})$$

where $\mathbf{A}_{\bar{y}}$ are the columns of \mathbf{A} , and $\mathbf{R}_{\bar{y}}$ are the columns of \mathbf{R} .

Once we have obtained all possible interactions at order ω , we represent them as interactions of type \mathbf{o} , and the transition matrices $\tilde{\mathbf{A}}$ have the same dimensions as $\tilde{\mathbf{T}}$. For each interaction we find the maximal value of a such that all elements of $\tilde{\mathbf{T}} - a\tilde{\mathbf{A}}$ are nonnegative: we compute minimal column sum of $\tilde{\mathbf{T}}$, and include the uniform distribution for the states of nodes

n for which $q_n = 0$. The coefficients read

$$a_i = \left(\min_{\bar{y}} \sum_y R_{y\bar{y}} \right) \prod_{q_n=0} |X_n|. \quad (\text{A5})$$

When all interactions at interaction order ω yield $a = 0$ we explore the next interaction order $\omega + 1$.

In the third step we select the interaction with the highest interaction coefficient a , and we extract it from $\tilde{\mathbf{T}}$: the new value, $\tilde{\mathbf{T}}'$, is given by $\tilde{\mathbf{T}}' = \tilde{\mathbf{T}} - a\tilde{\mathbf{A}}$. This sequence of steps is repeated until $\tilde{\mathbf{T}}'$ is a matrix of zeros, or equivalently until $\sum_i a_i = 1$. See the next subsection for an example.

4. Decomposition example

Consider the following example of $\tilde{\mathbf{T}}$ in a system with $N = 2$, $o = 2$, and $|X_n| = 2$. We here show an iteration of the decomposition algorithm step by step. The interaction to be extracted will be interaction \mathcal{A} with type $\mathbf{q} = (0, 2)$. Therefore we have $\mathbf{G}^o(t) = \mathbf{G}^{(2,2)}(t)$ and $\mathbf{G}^q(t) = \mathbf{G}^{(0,2)}(t)$ to characterize the complete system variables and the interaction variables, respectively. The auxiliary matrix \mathbf{R} is obtained from $\tilde{\mathbf{T}}$ as

$$\tilde{\mathbf{T}} = \frac{1}{100} \begin{pmatrix} 33 & 22 & 13 & 64 \\ 3 & 34 & 4 & 14 \\ 47 & 2 & 23 & 17 \\ 17 & 42 & 60 & 5 \end{pmatrix} \longrightarrow R = \frac{1}{100} \begin{pmatrix} 13 & 2 \\ 3 & 5 \end{pmatrix}. \quad (\text{A6})$$

Let us visualize how R_{00} has been computed. R_{00} accounts for $\pi(\mathbf{G}^{(0,1)}(t) = 0 | \mathbf{G}^{(0,1)}(t-1) = 0)$, and is thus associated with the elements of $\tilde{\mathbf{T}}$ that encode the same transition probability complemented with all the possible values for $\mathbf{G}^{(1,0)}(t)$ and $\mathbf{G}^{(1,0)}(t-1)$:

$$\begin{aligned} \pi(\mathbf{G}^{(1,1)}(t) = (0, 0) | \mathbf{G}^{(1,1)}(t-1) = (0, 0)) &= 33/100, \\ \pi(\mathbf{G}^{(1,1)}(t) = (0, 0) | \mathbf{G}^{(1,1)}(t-1) = (1, 0)) &= 13/100, \\ \pi(\mathbf{G}^{(1,1)}(t) = (1, 0) | \mathbf{G}^{(1,1)}(t-1) = (0, 0)) &= 47/100, \\ \pi(\mathbf{G}^{(1,1)}(t) = (1, 0) | \mathbf{G}^{(1,1)}(t-1) = (1, 0)) &= 23/100. \end{aligned}$$

The auxiliary matrix is constructed with the minimum of those elements such that when \mathbf{A} is removed from $\tilde{\mathbf{T}}$, no negative elements are created in the next iteration.

The second step yields $a = (2/100 + 5/100) 2$, where the first part comes from $R_{01} + R_{11}$, and the second part comes from the dimensions of $|X_0| = 2$,

$$\mathbf{R} = \frac{1}{100} \begin{pmatrix} 13 & 2 \\ 3 & 5 \end{pmatrix} \longrightarrow \mathbf{A} = \begin{pmatrix} \frac{13}{16} & \frac{2}{7} \\ \frac{3}{16} & \frac{5}{7} \end{pmatrix}. \quad (\text{A7})$$

In the third step we move from a $\mathbf{q} = (0, 2)$ representation of \mathbf{A} to a $\mathbf{q} = (2, 2)$ representation:

$$\begin{pmatrix} \frac{13}{16} & \frac{2}{7} \\ \frac{3}{16} & \frac{5}{7} \end{pmatrix} \longrightarrow \begin{pmatrix} \frac{13}{2 \cdot 16} & \frac{2}{2 \cdot 7} & \frac{13}{2 \cdot 16} & \frac{2}{2 \cdot 7} \\ \frac{3}{2 \cdot 16} & \frac{5}{2 \cdot 7} & \frac{3}{2 \cdot 16} & \frac{5}{2 \cdot 7} \\ \frac{13}{2 \cdot 16} & \frac{2}{2 \cdot 7} & \frac{13}{2 \cdot 16} & \frac{2}{2 \cdot 7} \\ \frac{3}{2 \cdot 16} & \frac{5}{2 \cdot 7} & \frac{3}{2 \cdot 16} & \frac{5}{2 \cdot 7} \end{pmatrix}. \quad (\text{A8})$$

At this point it becomes clear that the correction in the dimension at Eq. (A5) is necessary to normalize the $q_n = o$ form of \mathbf{A} to model nodes with $q_n = 0$.

5. Subsystem Interactions

We organize the transition probabilities of the model in a transition matrix \mathbf{W} ordered according to \mathcal{B} . The output matrices \mathbf{U} and \mathbf{V} , with dimensions $u_r \times u_c$ and $v_r \times v_c$ read

$$U_{ij} = \sum_{k=0}^{v_c-1} W_{iv_r+k, jv_c}, \quad V_{ij} = \sum_{k=0}^{u_c-1} W_{kv_r+i, j}. \quad (\text{A9})$$

If $\mathbf{W} = \mathbf{U} \otimes \mathbf{V}$ holds, \mathbf{U} and \mathbf{V} are valid subsystem interaction matrices. In that case, the procedure should be applied again on \mathbf{U} and \mathbf{V} until no more subsystems can be isolated. For an interaction \mathcal{A} of type \mathbf{q} subsystem, interactions at nodes \mathcal{B} have $\prod_{n \in \mathcal{B}} |X_n|^{q_n}$ parameters, and the decomposed interaction has only $\sum_{n \in \mathcal{B}} |X_n|^{q_n}$ degrees of freedom.

Let \mathbf{A} be the transition matrix of an interaction with $N = 2$, $|X_0| = 3$, and $|X_1| = 2$. Our goal here is to decompose \mathbf{A} as $\mathbf{A} = \mathbf{U} \otimes \mathbf{V}$ with $\mathbf{U} \in \mathbb{R}^{3 \times 3}$ and $\mathbf{V} \in \mathbb{R}^{2 \times 2}$. If we assumed $\mathbf{U} \otimes \mathbf{V}$, the following would hold:

$$\begin{pmatrix} u_{00} & u_{01} & u_{02} \\ u_{10} & u_{11} & u_{12} \\ u_{20} & u_{21} & u_{22} \end{pmatrix} \otimes \begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix} = \begin{pmatrix} u_{00}v_{00} & u_{00}v_{01} & u_{01}v_{00} & u_{01}v_{01} & u_{02}v_{00} & u_{02}v_{01} \\ u_{00}v_{10} & u_{00}v_{11} & u_{01}v_{10} & u_{01}v_{11} & u_{02}v_{10} & u_{02}v_{11} \\ u_{10}v_{00} & u_{10}v_{01} & u_{11}v_{00} & u_{11}v_{01} & u_{12}v_{00} & u_{12}v_{01} \\ u_{10}v_{10} & u_{10}v_{11} & u_{11}v_{10} & u_{11}v_{11} & u_{12}v_{10} & u_{12}v_{11} \\ u_{20}v_{00} & u_{20}v_{01} & u_{21}v_{00} & u_{21}v_{01} & u_{22}v_{00} & u_{22}v_{01} \\ u_{20}v_{10} & u_{20}v_{11} & u_{21}v_{10} & u_{21}v_{11} & u_{22}v_{10} & u_{22}v_{11} \end{pmatrix}. \quad (\text{A10})$$

It is possible to invert the tensor operation by using the property that both \mathbf{U} and \mathbf{V} are stochastic, and therefore sums over columns of \mathbf{V} or \mathbf{U} contained in \mathbf{A} are canceled out and can be used to isolate the remaining variable. As an example one may obtain $U_{00} = A_{00} + A_{10}$ and also $U_{00} = A_{10} + A_{11}$. The same is true for \mathbf{V} as $V_{00} = A_{00} + A_{20} + A_{40}$, $V_{00} = A_{02} + A_{22} + A_{42}$, and $V_{00} = A_{04} + A_{24} + A_{44}$. Since these equations are redundant, one may pick one at random and discard the rest. The expression at Eq. (A9) always the first one. This

technique will always output a \mathbf{U}, \mathbf{V} pair even if the tensor factorization does not exist. Therefore one should always check whether $\mathbf{A} = \mathbf{U} \otimes \mathbf{V}$ holds. For the purpose of finding subsystem interactions, we recursively apply this procedure on different combinations of interaction partitions until no more subsystem interactions can be found.

6. Complexity

In this section we prove that r is the fraction of the area behind $k^c(l)$. From the definition of $k(l) = \sum_{\lambda(\mathcal{A}^i)=l} a_i$ we could obtain the cumulative as $k^c(l) = \sum_{\lambda(\mathcal{A}^i) \leq l} a_i$. Instead, we are going to express the k as a function of a continuous variable $\phi \in [0, \lambda_{\text{MC}}]$ as $k = \sum a_i \delta(\phi - \lambda(\mathcal{A}^i))$. Then the cumulative $k^c(\phi)$ is

$$\begin{aligned} k^c(\phi) &= \int_0^\phi k(\phi') d\phi' \\ &= \int_0^\phi \sum a_i \delta(\phi' - \lambda(\mathcal{A}^i)) d\phi' \\ &= \sum a_i \int_0^\phi \delta(\phi' - \lambda(\mathcal{A}^i)) d\phi' \\ &= \sum a_i \theta(\phi - \lambda(\mathcal{A}^i)). \end{aligned}$$

Now the area behind $k^c(\phi)$ is

$$\begin{aligned} \int_0^{\lambda_{\text{MC}}} k^c(\phi) d\phi &= \int_0^{\lambda_{\text{MC}}} \sum a_i \theta(\phi - \lambda(\mathcal{A}^i)) d\phi \\ &= \sum a_i \int_0^{\lambda_{\text{MC}}} \theta(\phi - \lambda(\mathcal{A}^i)) d\phi \\ &= \sum a_i \int_{-\lambda(\mathcal{A}^i)}^{\lambda_{\text{MC}} - \lambda(\mathcal{A}^i)} \theta(\phi') d\phi' \\ &= \sum a_i (\lambda_{\text{MC}} - \lambda(\mathcal{A}^i)) \\ &= \lambda_{\text{MC}} - \sum a_i \lambda(\mathcal{A}^i). \end{aligned}$$

The total area is simply λ_{MC} , therefore the share is obtained by dividing the expression above with λ_{MC} , which yields exactly r .

-
- [1] C. W. J. Granger, *Econometrica* **37**, 424 (1969).
[2] J. Pearl, *Causality: Models, Reasoning, and Inference* (Cambridge University Press, 2000).
[3] J. Runge, *Chaos* **28**, 075310 (2018).
[4] R. Albert and A.-L. Barabási, *Rev. Mod. Phys.* **74**, 47 (2002).
[5] V. Latora, V. Nicosia, and G. Russo, *Complex Networks: Principles, Methods and Applications* (Cambridge University Press, 2017).
[6] M. Newman, *Networks: An Introduction* (Oxford University Press, 2018).
[7] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, *J. Complex Netw.* **2**, 203 (2014).
[8] S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, *Phys. Rep.* **544**, 1 (2014).
[9] C. Berge, *Hypergraphs: Combinatorics of Finite Sets* (Elsevier, 1984).
[10] A. Hatcher, *Algebraic Topology* (Cambridge University Press, 2002).
[11] P. Holme, *Eur. Phys. J. B* **88**, 234 (2015).
[12] A. R. Benson, D. F. Gleich, and J. Leskovec, *Science* **353**, 163 (2016).
[13] R. Lambiotte, M. Rosvall, and I. Scholtes, *Nat. Phys.* **15**, 313 (2019).
[14] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri, *Phys. Rep.* **874**, 1 (2020).
[15] L. Torres, A. S. Blevins, D. S. Bassett, and T. Eliassi-Rad, *SIAM Rev.* **63**, 435 (2021).
[16] C. Bick, E. Gross, H. A. Harrington, and M. T. Schaub, *SIAM Review* **65**, 686 (2023).

- [17] F. Battiston *et al.*, *Nat. Phys.* **17**, 1093 (2021).
- [18] A. A. Markov, *Sci. Context* **19**, 591 (2006).
- [19] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley, 2006).
- [20] J. Casadiego, M. Nitzan, S. Hallerberg, and M. Timme, *Nat. Commun.* **8**, 2192 (2017).
- [21] P. L. Williams and R. D. Beer, [arXiv:1004.2515](https://arxiv.org/abs/1004.2515).
- [22] F. E. Rosas, P. A. M. Mediano, M. Gastpar, and H. J. Jensen, *Phys. Rev. E* **100**, 032305 (2019).
- [23] F. E. Rosas, P. A. M. Mediano, H. J. Jensen, A. K. Seth, A. B. Barrett, R. L. Carhart-Harris, and D. Bor, *PLoS Comput. Biol.* **16**, e1008289 (2020).
- [24] Y. Zhang, V. Latora, and A. E. Motter, *Commun. Phys.* **4**, 195 (2021).
- [25] S. Chowdhary, A. Kumar, G. Cencetti, I. Iacopini, and F. Battiston, *J. Phys. Complex.* **2**, 035019 (2021).
- [26] L. Neuhäuser, R. Lambiotte, and M. T. Schaub, *Phys. Rev. E* **104**, 064305 (2021).
- [27] H. Akaike, *IEEE Trans. Autom. Control* **19**, 716 (1974).
- [28] G. Schwarz, *Ann. Statist.* **6**, 461 (1978).
- [29] I. Scholtes, *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2017), pp. 1037–1046.
- [30] Code available at <https://github.com/unaiapro/inin>.
- [31] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevE.108.034312> for practical examples of the formalism.
- [32] R. J. Glauber, *J. Math. Phys.* **4**, 294 (1963).