scientific reports

Check for updates

OPEN

Learning to predict synchronization of coupled oscillators on randomly generated graphs

Hardeep Bassi[1,6], Richard P. Yim[2,6], Joshua Vendrow[3], Rohith Koduluka[3], Cherlin Zhu[4] & Hanbaek Lyu[5]✉

Suppose we are given a system of coupled oscillators on an unknown graph along with the trajectory of the system during some period. Can we predict whether the system will eventually synchronize? Even with a known underlying graph structure, this is an important yet analytically intractable question in general. In this work, we take an alternative approach to the synchronization prediction problem by viewing it as a classification problem based on the fact that any given system will eventually synchronize or converge to a non-synchronizing limit cycle. By only using some basic statistics of the underlying graphs such as edge density and diameter, our method can achieve perfect accuracy when there is a significant difference in the topology of the underlying graphs between the synchronizing and the non-synchronizing examples. However, in the problem setting where these graph statistics cannot distinguish the two classes very well (e.g., when the graphs are generated from the same random graph model), we find that pairing a few iterations of the initial dynamics along with the graph statistics as the input to our classification algorithms can lead to significant improvement in accuracy; far exceeding what is known by the classical oscillator theory. More surprisingly, we find that in almost all such settings, dropping out the basic graph statistics and training our algorithms with only initial dynamics achieves nearly the same accuracy. We demonstrate our method on three models of continuous and discrete coupled oscillators—the Kuramoto model, Firefly Cellular Automata, and Greenberg-Hastings model. Finally, we also propose an "ensemble prediction" algorithm that successfully scales our method to large graphs by training on dynamics observed from multiple random subgraphs.

Many important phenomena that we would like to understand—formation of public opinion, trending topics on social networks, movement of stock markets, development of cancer cells, the outbreak of epidemics, and collective computation in distributed systems – are closely related to predicting large-scale behaviors in networks of locally interacting dynamic agents. Perhaps the most widely studied and mathematically intractable of such collective behavior is the *synchronization* of coupled oscillators (e.g., blinking fireflies, circadian pacemakers, BZ chemical oscillators), and has been an important subject of research in mathematics and various areas of science for decades[1,2]. Moreover, it is closely related to the *clock synchronization* problem, which is essential in establishing shared notions of time in distributed systems and has enjoyed fruitful applications in many areas including wildfire monitoring, electric power networks, robotic vehicle networks, large-scale information fusion, and wireless sensor networks[3–5].

For a system of deterministic coupled oscillators (e.g., the Kuramoto model[6]), the entire forward dynamics (i.e., the evolution of phase configurations) is analytically determined by (1) the initial phase configuration and (2) the graph structure (see Fig. 1). In this paper, we are concerned with the fundamental problem of *predicting whether a given system of coupled oscillators will eventually synchronize*, using some information on the underlying graph or on the initial dynamics (that is, early-stage of the forward dynamics). More specifically, we consider the following three types of synchronization prediction problems (see Fig. 4):

Q1 Given the initial dynamics and graph structure, can we predict whether the system will eventually synchronize?

[1]Department of Applied Mathematics, University of California, Merced, CA 95343, USA. [2]Department of Mathematics, University of California, Davis, CA 95616, USA. [3]Department of Mathematics, University of California, Los Angeles, CA 90095, USA. [4]Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218, USA. [5]Department of Mathematics, University of Wisconsin, Madison, WI 53706, USA. [6]These authors contributed equally: Hardeep Bassi and Richard P. Yim. ✉email: hlyu@math.wisc.edu

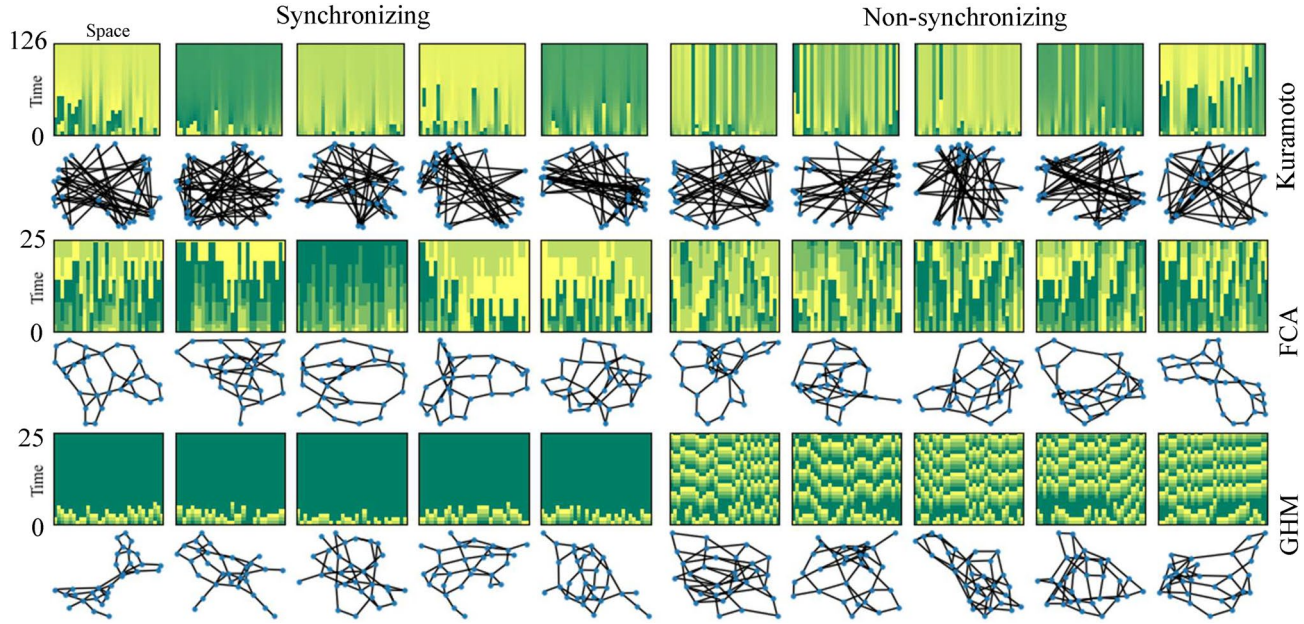

Figure 1. Sample points in the 30-node dynamics dataset for synchronization prediction. The heat maps show phase dynamics on graphs beneath them, where colors represent phases and time is measured by iterations from bottom to top (e.g. $t = 0$–$t = 25$). Each example is labeled as 'synchronizing' if it synchronizes at iteration 1758 for the Kuramoto model (70 for FCA and GHM) and 'non-synchronizing' otherwise. Synchronizing examples have mostly uniform colors in the top row. For training, only a portion of dynamics is used so that the algorithms rarely see a fully synchronized example (see Fig. 4).

Q2 Given the initial dynamics and not knowing the graph structure, can we predict whether the system will eventually synchronize?

Q3 Given the initial dynamics partially observed on a subset of nodes and possibly not knowing the graph structure, can we predict whether the whole system will eventually synchronize?

Analytical characterization of synchronization would lead to a perfect algorithm for the synchronization prediction problems above. However, while a number of sufficient conditions on graph topology[7–10] (e.g., complete graphs or trees), model parameters (e.g., large coupling strength) or on initial configuration (e.g., phase concentration into open half-circle) for synchronization are known, obtaining an analytic or asymptotic solution to the prediction question, in general, appears to be out of reach, especially when these sufficient conditions for synchronization are not satisfied. Namely, we are interested in predicting the synchronization of coupled oscillators where the underlying graphs are non-isomorphic and the initial configuration is not confined within an open half-circle in the cyclic phase space. Since the global behavior of coupled oscillators is built on non-linear local interactions, as the number of nodes increase and the topology of the underlying graphs become more diverse, the behavior of the system becomes rapidly intractable. To provide a sense of the complexity of the problem, note that there are more than 10^9 non-isomorphic connected simple graphs with 11 nodes[11].

However, the lack of a general analytical solution does not necessarily preclude the possibility of successful prediction of synchronization. In this work, we propose a radically different approach to this problem that we call *Learning To Predict Synchronization* (L2PSync), where we view the synchronization prediction problem as a binary classification problem for two classes of 'synchronizing' and 'non-synchronizing', based on the fact that any given deterministic coupled oscillator system eventually synchronizes or converges to a non-synchronizing limit cycle. In this work, we consider three models of continuous and discrete coupled oscillators—the Kuramoto model (KM)[2], Firefly Cellular Automata (FCA)[10], and Greenberg-Hastings model (GHM)[12].

Utilizing a few basic statistics of the underlying graphs, our method can achieve perfect accuracy when there is a significant difference in the topology of the underlying graphs between the synchronizing and the non-synchronizing examples (see Figs. 2, 3). We find that when these graph statistics cannot separate the two classes of synchronizing and non-synchronizing very well (e.g., when the graphs are generated from the same random graph model, see Tables 2, 3), pairing a few iterations of phase configurations in the initial dynamics along with the graph statistics as the input to the classification algorithms can lead to significant improvement in accuracy. Our methods far surpass what is known by the half-circle concentration principle in classical oscillator theory (see our "Methods" section). We also find that in almost all such settings, dropping out the basic graph statistics and training our algorithms with only initial dynamics achieves nearly the same accuracy as with the graph statistics. Furthermore, we find that our methods are robust under using incomplete initial dynamics only observed on a few small subgraphs of large underlying graphs.

Problem statement. A graph $G = (V, E)$ consists of sets V of nodes and E of edges. Let Ω denote the *phase space* of each node, which may be taken to be the circle $\mathbb{R}/2\pi\mathbb{Z}$ for continuous-state oscillators or the color

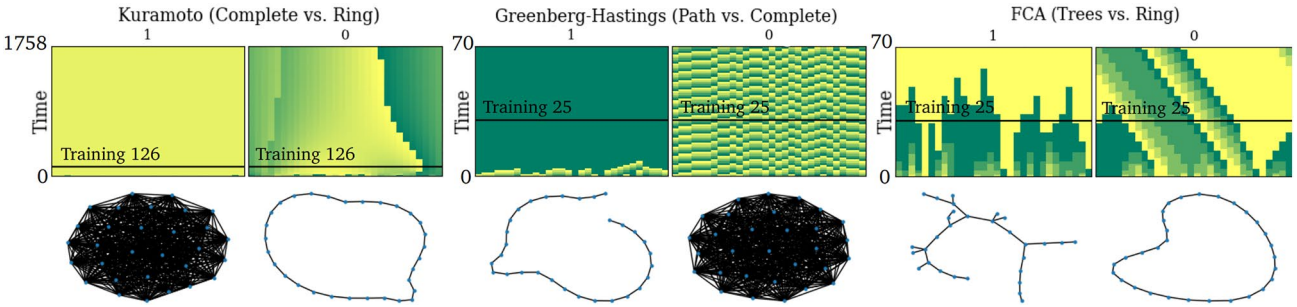

Figure 2. Examples of synchronizing and non-synchronizing dynamics of Kuramoto, Greenberg-Hastings, and FCA oscillators on 30-node graphs with two topologies (complete vs. ring, path vs. complete, and tree vs. ring, respectively). The heat maps show phase dynamics on graphs beneath them, where colors represent phases and time is measured by iterations from bottom-to-top (e.g. $t = 0$–$t = 70$). Synchronizing examples have uniform color in the top row. The horizontal bar indicates the 'training iteration', which is the maximum number of iterations in the initial dynamics fed into the classification algorithm for prediction.

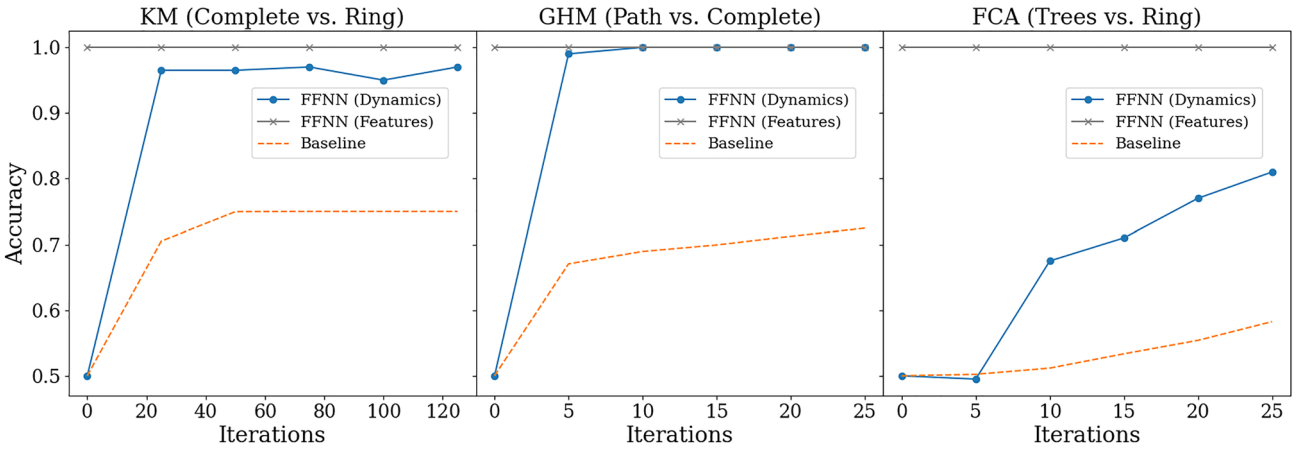

Figure 3. Binary classification accuracies of synchronizing vs. non-synchronizing dynamics when the underlying graphs in either of these two classes share the same topology. For the Kuramoto model, we compare complete graphs (synchronizing) to rings (non-synchronizing), for the GHM model we compare rings (synchronizing) vs complete graphs (non-synchronizing), and for FCA we compare trees with maximum degree 4 (synchronizing) to rings (non-synchronizing). FFNN (Features) uses FFNN as the binary classification algorithm with five basic graph features—the number of edges, min/max degree, diameter, and the number of nodes—as the input, which achieves perfect classification in all cases. FFNN (Dynamics) uses FFNN as the binary classification algorithm with a varying number of iterations of initial dynamics as input (specified as the horizontal axis), without any information about the underlying graph. The baseline predictor uses the concentration principle (see the main text for more details). Notice that FFNN (Dynamics) far surpasses the baseline.

wheel $\mathbb{Z}/\kappa\mathbb{Z}$, $\kappa \in \mathbb{N}$ for discrete-state oscillators. We call a map $X : V \to \Omega$ a *phase configuration*, and say it is *synchronized* if it takes a constant value across nodes (i.e., $X(v) = Const.$ for all $v \in V$). A *coupling* is a function $\mathscr{F}$ that maps each pair (G, X_0) of graph and initial configuration $X_0 : V \to \Omega$ deterministically to a *trajectory* $(X_t)_{t\geq 0}$ of phase configurations $X_t : V \to \Omega$. For instance, $\mathscr{F}$ could be the time evolution rule for the KM, FCA, or GHM. Throughout the paper, $\mathbf{1}(\cdot)$ denotes the indicator function. The main problem we investigate in this work is stated below:

- (Synchronization Prediction) *Fix parameters* $n \in \mathbb{N}, T \gg r > 0$, *and coupling* $\mathscr{F}$. *Predict the following indicator function* $\mathbf{1}(X_T \text{is synchronized})$ *given the initial trajectory* $(X_t)_{0\leq t\leq r}$ *and optionally also with statistics of graph* G.

We remark that as T tends to infinity, the indicator in the problem statement converges to the following indicator function $\mathbf{1}(X_t \text{is eventually synchronized})$ which aligns more directly with the initial questions Q1 and Q2 than the indicator in the above problem statement, but determining whether a given system will never synchronize in amounts to finding a non-synchronizing periodic orbit, which is computationally infeasible in general. See Fig. 4 for an illustration of the synchronization prediction problem.

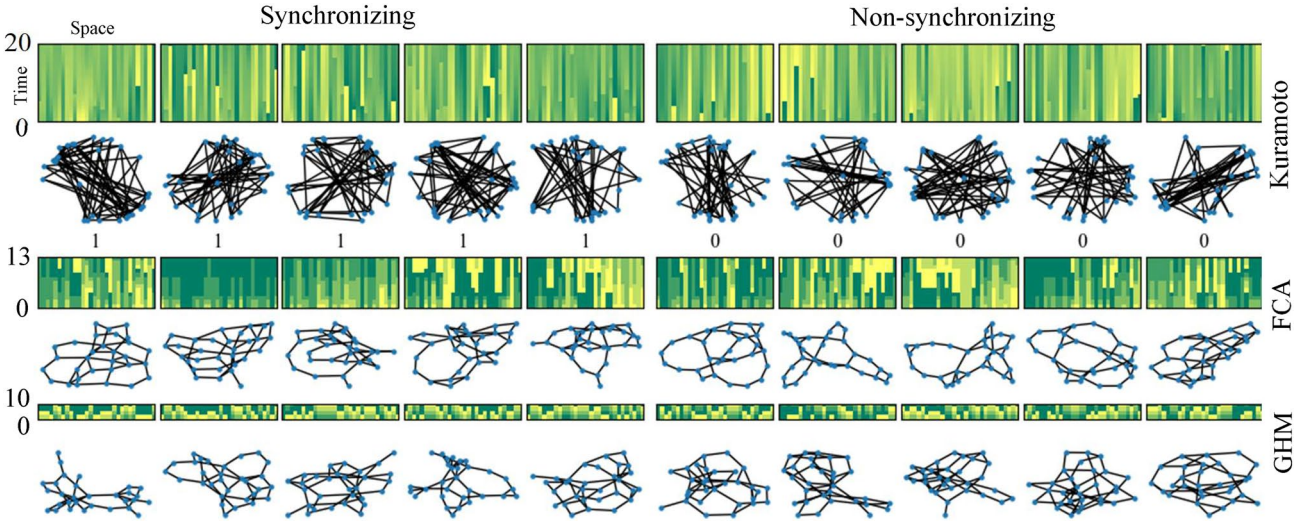

Figure 4. Sample points in the 30-node training data set for synchronization prediction. The full dataset consists of 40 K synchronizing and 40 K non-synchronizing 30-node connected non-isomorphic graphs and dynamics on them for each of the three models the KM, FCA, and GHM (see Table 2).

References	# nodes	# graphs	# configs.	model	ML	Goal
Fan et al.[13]	2-4	1	1	Lorenz, KM	FFNN	Phase transition
Guth et al.[14]	N/A	N/A	1	1D and 2D PDE	SO	Extreme events
Chowdhury et al.[15]	200	1	1	Mean-field logistic map	LSTM	Extreme events
Thiem et al.[17]	1500-8000	1	2000	KM	FFNN	Parameter estimation
Biccari et al.[19]	10-1000	1	1	KM	GD, RBM	Parameter estimation
Itabashi et al.[21]	128-256	1	100	KM	TDA	Synchronization/ Non-synchronization/ Chimera state
This work	15-600	**2K-200K**	1	KM, FCA, GHM	RF, GB, FFNN, LRCN	Synchronization/ Non-synchronization

Table 1. Comparison of settings in related works on learning coupled oscillator dynamics using machine learning methods. Recent works in the table focus on learning features of dynamics on fixed graphs. In contrast, we aim to classify the long-term dynamics of a given system on a diverse set of underlying graphs. The column for '# configs.' refers to the number of distinct initial phase configurations considered for each graph in training.

Related works. There are a number of recent works incorporating machine learning methods to investigate problems on coupled oscillators or other related dynamical systems, which we briefly survey in this section and summarize in Table 1.

In Fan et al.[13], the authors are concerned with identifying the critical coupling strength at which a system of coupled oscillators undergo a phase transition into synchronization, where the underlying graph consists of 2-4 nodes with a fixed topology (e.g., a triangle or a star with three leaves). Guth et al.[14] use binary classification methods with surrogate optimization (SO) in order to learn optimal parameters and predictors of extreme events. Their work is primarily concerned with learning whether or not intermittent extreme events will occur in various 1D or 2D partial differential equation models. Similarly, Chowdhury et al.[15] utilize a long-short term memory (LSTM)[16] network to predict whether or not an extreme event will occur on globally coupled mean-field logistic maps on complete graphs. Thiem et al.[17] use Feed-forward neural networks (FFNN)[18] to learn coarse-grained dynamics of Kuramoto oscillators and recover the classical order parameter. Biccari et al.[19] use gradient descent (GD) and the random batch method (RBM) to learn control parameters to enhance the synchronization of Kuramoto oscillators. Slightly less related work is Hefny et al.[20], where the authors use hidden Markov models, LASSO regression, and spectral algorithms for learning lower-dimensional state representations of dynamical systems and apply their method to a knowledge tracing model for a dataset of students' responses to a survey.

On the other hand, Itabashi et al.[21] consider classifying coupled Kuramoto oscillators according to their future dynamics from certain features derived (using topological data analysis (TDA)) from their early-stage dynamics (or 'initial dynamics' in our terminology). As in the other references above, the underlying graph is fixed in each classification task (a complete graph with weighted edges, which may have two or four communities). But unlike in the references above, the initial configuration, instead of the model parameter, is varied to generate different examples on the same underlying graph. The authors observed that some long-term dynamical

properties (e.g., multi-cluster synchronization) can be predicted by the derived features of the initial dynamics. This point is consistent with one of our findings that the first few iterations of the initial dynamics may contain crucial information in predicting the long-term behavior of coupled oscillator systems.

While sharing high-level ideas and approaches with the aforementioned works, our work has multiple distinguishing characteristics in the problem setting and approaches. First, in all of the aforementioned works, a dynamical system on a fixed underlying graph is considered. But in our setting, there are as many as 200 K non-isomorphic underlying graphs and we seek to predict whether a given system of oscillators on highly varied or even unknown graphs will eventually synchronize or not. Furthermore, we also consider the case when machine learning algorithms are trained on partial observation (e.g., initial dynamics restricted on some subgraphs). Second, only in our work, are discrete models of coupled oscillators (e.g., FCA and GHM) considered, whereas only models with continuous phase space (e.g., the KM) are considered in the literature. Third, only in our work, is the classical concentration principle (a.k.a. the 'half-circle concentration', see our "Methods" section) in the oscillator theory used as a rigorous benchmark to evaluate the efficacy of employed machine learning methods. Finally, we employ various binary classification algorithms such as Random Forest (RF)[22] Gradient Boosting (GB)[23], Feed-forward Neural Networks (FFNN)[18], and our own adaptation of a Long-term Recurrent Convolutional Networks (LRCN)[24].

We remark that there are a number of cases where rigorous results are available for the question of predicting the long-term behavior of coupled oscillators on a graph G and initial configuration X_0. For instance, the $\kappa = 3$ instances of GHM and another related model called Cyclic Cellular Automata (CCA)[25] have been completely solved[26]. Namely, given the pair (G, X_0), the trajectory X_t synchronizes eventually if and only if the discrete vector field on the edges of G induced from X_0 is conservative (see[26] for details). Additionally, the behavior of FCA on finite trees is also well-known: given a finite tree T and $\kappa \in \{3, 4, 5, 6\}$, every κ-color initial configuration on T synchronizes eventually under κ-color FCA if and only if the maximum degree of T is less than κ; for $\kappa \geq 7$, this phenomenon does not always hold[10,27]. This theoretical result on FCA was used in the experiment in Figs. 2 and 3. Furthermore, there is a number of works on the clustering behavior of these models on the infinite one-dimensional lattice, $\mathbb{Z}$ (FCA[10,28], CCA[28–31] and GHM[28,32]).

Methods

The pipeline of our approach is as follows. Namely, (1) fix a model for coupled oscillators; (2) generate a *dynamics dataset* of a large number of non-isomorphic graphs with an even split of synchronizing and non-synchronizing dynamics; (3) train a selected binary classification algorithm on the dynamics dataset to classify each example (initial dynamics with or without underlying features of the graph) into one of two classes, 'synchronizing' or 'non-synchronizing'; (4) validate the accuracy of the trained algorithms on fresh examples by comparing the predicted behavior of the true long-term dynamics. We use the following classification algorithms: Random Forest (RF)[22] Gradient Boosting (GB)[23], Feed-forward Neural Networks (FFNN)[18], and our own adaptation of Long-term Recurrent Convolutional Networks (LRCN)[24] which we call the *GraphLRCN* (further information such as implementation details and hyperparameters can be found in the SI).

As a baseline for our approach, we use a variant of the well-known "concentration principle" in the literature on coupled oscillators. Namely, regardless of the details of graph structure and model, synchronization is guaranteed if the phases of the oscillators are concentrated in a small arc of the phase space *at any given time* (see the next subsection). This principle is applied at each configuration up to the training iteration used to train the binary classifiers.

For question Q3 on synchronization prediction on the initial dynamics partially observed on subgraphs, as well as reducing the computational cost of our methods for answering Q1 and Q2, we propose an "ensemble prediction" algorithm (Algorithm 1) that scales up our method to large graphs by training on dynamics observed from multiple random subgraphs. Namely, suppose we are to predict the dynamics of some connected N-node graphs, where only the initial dynamics are observed on a few small connected subgraphs of $n \ll N$ nodes. We first train a binary classification algorithm on the dynamics observed from those subgraphs and then aggregate the predictions from each subgraph (e.g., using a majority vote) to get a prediction for the full dynamics on N nodes.

The concentration principle for synchronization and baseline predictor. In the literature on coupled oscillators, there is a fundamental observation that concentration (e.g., into an open half-circle) of the initial phase of the oscillators leads to synchronization for a wide variety of models on arbitrary connected graphs (see, e.g.,[33], Lem 5.5). This is stated in the following bullet point for the KM and FCA and we call it the "concentration principle". This principle has been used pervasively in the literature of clock synchronization[34–37] and also in multi-agent consensus problems[38–40].

- (Concentration Principle) Let G be an arbitrary connected graph. For the Kuramoto model (see Eq. (1) in SI) with identical intrinsic frequency and for FCA (see Eq. (3) in SI), given dynamics on G synchronize if all phases at any given time are confined in an open half-circle in the phase space Ω. Furthermore, if all states used in the time-t configuration X_t are confined in an open half-circle for any $t \geq 1$, then the trajectory on G eventually synchronizes.

The 'open half-circle' refers to any arc of length $< \pi$ for the continuous phase space $\Omega = \mathbb{R}/2\pi\mathbb{Z}$ and any interval of $< \kappa/2$ consecutive integers (mod κ) for the discrete phase space $\Omega = \mathbb{Z}/\kappa\mathbb{Z}$. This is a standard fact known to the literature and it follows from the fact that the couplings in the statement monotonically contract given any initial phase configuration under the half-circle condition toward synchronization. It is not hard to see the

above half-circle concentration does not hold for GHM. Accordingly, for GHM, we say a phase configuration X_t is *concentrated* if X_t is synchronized.

We now introduce the following baseline synchronization predictor: given $(X_t)_{0\le t\le r}$ and $T > r$,

- (Baseline predictor) Predict synchronization of X_T if X_t is concentrated for any $1 \le t \le r$. Otherwise, flip a fair coin.

Notice that the baseline predictor never predicts synchronization incorrectly if X_r is concentrated. For non-concentrated cases, the baseline does not assume any knowledge and gives a completely uninformed prediction. Quantitatively, suppose we have a dataset where α proportion of entire samples are synchronizing (in all our datastes, $\alpha = 0.5$). Suppose we apply the baseline predictor where we use the first r iterations of dynamics for each sample. Let $x = x(r)$ denote the proportion of synchronizing samples that concentrate by iteration r among all among all synchronizing samples. Then the baseline predictor's accuracy is given by $x\alpha + (1 - \alpha + (1 - x)\alpha)/2 = 0.5 + x\alpha/2$, where the term $x\alpha/2$ can be regarded as the gain obtained by using the concentration principle layered onto the uninformed decision.

An illustrative example: the Kuramoto model on complete vs. ring; GHM on path vs. complete; FCA on tree vs. ring. We first give a simple example to illustrate our machine learning approach to the synchronization prediction problem. More specifically, we create datasets of an equal number of synchronizing and non-synchronizing examples, where there is a significant difference in the topology of the underlying graphs between the synchronizing and the non-synchronizing examples. In such a setting, one can expect that knowing the basic graph features—the number of edges, min/max degree, diameter, and the number of nodes—will be enough to distinguish between synchronizing and non-synchronizing examples.

For each of the coupled oscillator models of the KM, FCA, and GHM, we create a dataset that consists of 1 K synchronizing examples and 1 K non-synchronizing examples on 30 node graphs, where all 1 K examples in each of the two classes share the same underlying graph topology. For the KM, the synchronizing and the non-synchronizing examples are on a complete graph and on a ring, respectively. For GHM (with $\kappa = 5$), the synchronizing and the non-synchronizing examples are on a path and on a complete graph, respectively. Lastly for FCA (with $\kappa = 5$), the synchronizing and the non-synchronizing examples are on a tree with a maximum degree at most four and on a ring, respectively. Our choice of these graph topologies reflects rigorously established results in the literature. Namely, it is well-known that coupled Kuramoto oscillators (with identical intrinsic frequencies) on a complete graph will always synchronize, and one can easily generate a non-synchronizing initial configuration for Kuramoto oscillators on a ring[1,2]. For GHM, in the SI, we prove that a κ-color (arbitrary $\kappa \ge 3$) GHM on a path of length n will always synchronize by iteration $n + \kappa$, regardless of the initial configuration. Lastly for FCA, it is known that FCA with $\kappa = 5$ color dynamics always synchronize on trees with maximum degree at most four[10]. The results of our experiments for these three datasets are shown in Fig. 3.

For the three datasets described above, we achieve perfect classification accuracy with FFNN of distinguishing synchronizing vs non-synchronizing by using the following five basic statistics of the underlying graphs—the number of edges, min/max degree, diameter, and the number of nodes—as expected by the theoretical results we mentioned above. Additionally, we hide the graph statistics completely and train FFNN only using the initial dynamics up to a variable training iteration. As we feed in more initial dynamics for training, FFNN quickly improves in prediction accuracy, far more rapidly than the baseline does. This indicates that FFNN may be quickly learning some latent distinguishing features from the initial dynamics that are more effective than the half-circle concentration employed by the baseline predictor.

Generating the dynamics datasets. In the example we considered in Figs. 2 and 3, there was a clear distinction between the topology of the underlying graphs that were synchronizing and the graphs that were non-synchronizing, and basic graph statistics can yield perfect classification accuracy through FFNN as the choice of the binary classifier. In this section, we consider datasets for which it is much more difficult to classify based only on the same graph statistics. The way we do so is by generating a large number of underlying graphs from the *same* random graph model with the *same* parameters. In this way, even though individual graphs realized from the random graph model may have different topologies, graph statistics such as edge density or maximum degree are concentrated around their expected values. This is in contrast to the set of underlying graphs in the three datasets we considered in Figs. 2 and 3, where half of them are isomorphic to each other and another half are also isomorphic to each other. For the datasets we generate in this way, which we will describe in more detail shortly after, classifying only with the basic graph statistics achieves an accuracy of 60–70% (in comparison to 100% as before).

We generate a total of twelve datasets described in Tables 2 and 3 as follows. Data points in each dataset consist of three statistics computed for a pair (G, X_0) of an underlying graph, $G = (V, E)$, and initial configuration, $X_0 : V \to \Omega$: (1) first r iterations of dynamics $(X_t)_{0\le t\le r}$ (using either the KM, FCA, or GHM), (optional) (2) features of G and X_0, and (3) the label that indicates whether X_T is concentrated or not. We optionally include the following six features: *number of edges*, *min degree*, *max degree*, *diameter*, *number of nodes*, and *quartiles of initial phases in* X_0. We say a data point is 'synchronizing' if the label is 1, and 'non-synchronizing,' label 0, otherwise. Every dataset we generate contains an equal number of synchronizing and non-synchronizing examples, and the underlying graphs are all connected and non-isomorphic (as opposed to the datasets in Figs. 2 and 3).

To generate a single n-node graph, we use an instance of the Newman-Watts-Strogatz (NWS) model[41], which originally has three parameters n (number of nodes), p (shortcut edge probability) and k (initial degree of nodes; We use the implementation in `networkx` python package, see[42]), with an added integer parameter M (number

Datasets	$\mathtt{KM}_{15}$	$\mathtt{KM}_{30}$	$\mathtt{FCA}_{15}$	$\mathtt{FCA}_{30}$	$\mathtt{GHM}_{15}$	$\mathtt{GHM}_{30}$
# nodes	15	30	15	30	15	30
avg of # edges	29.65	57.49	23.91	47.45	22.88	48.18
std of # edges	3.42	5.67	2.34	4.15	2.35	4.11
Avg diameter	4.32	7.00	4.32	6.01	4.30	6.12
std of diameter	0.68	1.29	0.67	0.95	0.66	0.98
r (training iter)	126	126	25	25	25	25
T (prediction iter)	1758	1758	70	70	70	70
# Sync.	100K	40K	100K	40K	100K	40K
# Nonsync.	100K	40K	100K	40K	100K	40K

Table 2. Dynamics datasets were generated for three models with two node counts. In each dataset, all graphs are connected and non-isomorphic. # Sync. denotes the number of examples in the dataset such that the phase configuration X_T at iteration T is concentrated.

Datasets	$\mathtt{FCA}_{600}$	$\mathtt{FCA}'_{600}$	$\mathtt{FCA}''_{600}$	$\mathtt{KM}_{600}$	$\mathtt{KM}'_{600}$	$\mathtt{KM}''_{600}$
# nodes	600	600	300-600	600	600	300-600
std of # nodes	0	0	86.60	0	0	86.60
avg of # edges	2985.53	4749.24	2799.49	1051.08	1109.85	757.89
std of # edges	37.85	2371.72	1461.08	19.66	79.47	56.81
r	50	50	50	400	400	400
T	600	600	600	1758	1758	1758
# Sync.	1K	1K	1K	1K	1K	1K
# Nonsync.	1K	1K	1K	1K	1K	1K

Table 3. Dynamics datasets generated for FCA and Kuramoto on 600 nodes ($\mathtt{FCA}_{600}$, $\mathtt{KM}_{600}$, $\mathtt{FCA}'_{600}$, $\mathtt{KM}'_{600}$) and on 300–600 nodes ($\mathtt{FCA}''_{600}$, $\mathtt{KM}''_{600}$). In each dataset, all graphs are connected and non-isomorphic. # Sync. denotes the number of examples in the dataset such that the phase configuration X_T at iteration T is concentrated. Here r and T refers to the training and the prediction iterations defined in Table 2.

of calls for adding shortcut edges). Namely, we start from a cycle of n nodes, where each node is connected to its k nearest neighbors. Then we attempt to add a new edge between each initial non-edge (u, v) with probability $p/(n-k-3)$ independently M times. The number of new edges added in this process follows the binomial distribution with $\binom{n}{2} - \frac{nk}{2}$ trials with success probability $\left(1-\left(1-\frac{p}{n-k-1}\right)\right)^M \approx pm/(n-k-1)$. This easily yields that the expected number of edges in our random graph model is $\frac{nk}{2} + \frac{n^2pM}{2(n-k-1)} + O(k)$.

The NWS model of random graphs is known to exhibit the 'small world property', which is to have relatively small mean path lengths in contrast to having a low local clustering coefficient. This is a widely observed property of many real-world complex networks[41]—as opposed to the commonly studied Erdös-Réyni graphs. It is known that many models of coupled oscillators have a hard time synchronizing when the underlying graph is a ring, as the discrepancy between oscillators tends to form traveling waves circulating on the ring[43]. On the other hand, it is observed that coupled oscillator systems on dense graphs are relatively easier to synchronize. For instance, Kassabov, Strogatz, and Townsend recently showed that Kuramoto oscillators with an identical natural frequency on a connected graph where each node is connected to at least 3/4 of all nodes are globally synchronizing for almost all initial configurations[44]. Since we intend to generate both synchronizing and non-synchronizing examples to form a balanced dataset, it is natural for us to use a random graph model that sits somewhere between rings and dense graphs. In this sense, NWS is a natural choice for a random graph model for our purpose. Using other models such as Erdös–Réyni, for example, for generating the balanced dataset of synchronizing and non-synchronizing examples as in Tables 2 and 3, is computationally very demanding.

In Table 2, we make note of the average graph edge counts and standard deviations of the random graphs that were used to simulate our models. These characteristics, average edge count and standard deviation of edges counts, correspond to the clustering of edges in the graph, which intuitively affects the propagation of information or states in our cellular automata.

Also in Table 2, we give a summary of the six datasets on the three models for two node counts $n = 15, 30$, each with 200 K and 80 K examples, respectively, which we refer to as $\mathtt{KM}_n, \mathtt{FCA}_n, \mathtt{GHM}_n$ for $n = 15, 30$. Underlying graphs are sampled from the NWS model with parameters $n \in \{15, 30\}$, $M = 1$, and $p = 0.85$ for the KM and $p = 0.65$ for FCA and GHM. In all cases, we generated about 400 K examples and subsampled 200 K and 80 K examples for $n = 15, 30$, respectively, so that there are an equal number of synchronizing and non-synchronizing examples, with all underlying graphs as non-isomorphic. The limits for both sets were chosen by memory constraints imposed by the algorithms used. To give a glance at the datasets, we provide visual representations. In Fig. 4, we show five synchronizing and non-synchronizing examples in $\mathtt{KM}_{30}$, $\mathtt{FCA}_{30}$, and $\mathtt{GHM}_{30}$.

We also generated six dynamics datasets with a larger number of nodes on FCA and Kuramoto dynamics, as described in Table 3. The fixed node datasets $\texttt{FCA}_{600}$, $\texttt{FCA}'_{600}$, $\texttt{KM}_{600}$, and $\texttt{KM}'_{600}$ each consist of 1K synchronizing and non-synchronizing examples of FCA and Kuramoto dynamics on non-isomorphic graphs of 600 nodes. The underlying graphs for the fixed node FCA datasets are generated by the NWS model with parameters $n = 600$, $p = 0.6$ and $N = 5$ for $\texttt{FCA}_{600}$ and $p \sim \text{Normal}(\mu_X, 0.04)$, $\mu_X \sim \text{Uniform}(0.32, 0.62)$, for each $N \sim \text{Uniform}(\{1, 2, \ldots, 20\})$ calls for $\texttt{FCA}'_{600}$. Similarly, to generate the fixed node Kuramoto datasets, $\texttt{KM}_{600}$ and $\texttt{KM}'_{600}$, we used the NWS model with parameters $n = 600$, $p = 0.15$ and $N = 5$ for $\texttt{KM}_{600}$ and $p \sim \text{Normal}(\mu_X, 0.04)$, $\mu_X \sim \text{Uniform}(0.32, 0.62)$, for each $N \sim \text{Uniform}(\{1, 2, \ldots, 20\})$ calls for $\texttt{KM}'_{600}$. Consequently, the number of edges in the graphs from $\texttt{KM}_{600}$ and $\texttt{FCA}_{600}$ are sharply concentrated around its mean whereas $\texttt{KM}'_{600}$ and $\texttt{FCA}'_{600}$ have much greater overall variance in the number of edges (see Table 3). For the varied node datasets, $\texttt{FCA}''_{600}$ and $\texttt{KM}''_{600}$, we kept p distributed in the same way as done for $\texttt{KM}'_{600}$ $\texttt{FCA}'_{600}$, but for each $N \sim \text{Uniform}(\{1, 2, \ldots, 20\})$ calls of adding shortcut edges, but additionally varied the number of nodes as $n \sim \text{Uniform}(\{300, 301, \ldots, 600\})$. In this case, both the number of nodes and edges have relatively greater variation compared to the other datasets.

We omit the GHM from this experiment because the dynamics are extremely prone to non-synchronization as a network has more cycles[45]. Hence, for these large graphs, almost all GHM dynamics will be non-synchronizing. For instance, for the same set of networks we used to generate the $\texttt{FCA}_{600}$ dataset (consisting of 40K networks of 600 nodes), none of the GHM dynamics synchronized. Hence, there is no meaningful classification problem to be discussed since the presence of synchronizing graphs that follow this model is extremely sparse.

Scaling up by learning from subgraphs. In this subsection, we discuss a way to extend our method for the dynamics prediction problem simultaneously in two directions; (1) larger graphs and (2) a variable number of nodes. The idea is to train our dynamics predictor on subsampled dynamics of large graphs (specified as induced subgraphs and induced dynamics), and to combine the local classifiers to make a global prediction. In the algorithm below, $f(X_T) := \mathbf{1}(X_T \text{is concentrated})$, and if X_t is a phase configuration on $G = (V, E)$ and $G_i = (V_i, E_i)$ is a subgraph of G, then $X_t|_{G_i}$ denotes the restriction $v \mapsto X_t(v)$ for all $v \in V_i$.

Algorithm 1 Ensemble Prediction of Synchronization

Input: Dynamics dataset on graphs with $\geq N$ nodes; Test point $(G', (X'_t)_{0 \leq t \leq r})$;
Parameters: $n_0 \leq N$ (size of subgraphs), $k_{\text{train}}, k_{\text{test}}$ (# of subgraphs), θ (prediction threshold)
Subsample Dynamics:
  For each data point $(G, (X_t)_{0 \leq t \leq r}, f(X_T))$:
    Sample n_0-node connected subgraphs $G_1, \ldots, G_{k_{\text{train}}}$ of G;
    Form restricted triples $(G_i, (X_t|_{G_i})_{0 \leq t \leq r}, f(X_T))$
Train Dynamics Predictor:
  Train a binary classifier on the restricted triples;
Ensemble Prediction:
  Sample n_0-node connected subgraphs $G'_1, \ldots, G'_{k_{\text{test}}}$ of G';
  $\hat{f} :=$ mean of predictions of $f(X_T)$ on subdynamics on G'_i's
Output: $\mathbf{1}(\hat{f} > \theta)$

Results

Regardless of the three models of coupled oscillators and selected binary classification algorithm, we find that our method used to address Q1 and Q2 on average shows at least a 30% improvement in prediction performance compared to this concentration–prediction baseline for dynamics on 30 node graphs. In other words, our results indicate that the concentration principle applied at each configuration is too conservative in predicting synchronization, and there might be a generalized concentration principle that uses the whole initial dynamics, which our machine learning methods seem to learn implicitly.

Using Algorithm 1 for Q3, we achieve an accuracy score of over 85% for predicting the commonly studied Kuramoto model on 600 node graphs by only using four 30-node subgraphs where the corresponding baseline gets 55% accuracy. In particular, we observe the baseline with locally observed initial dynamics tends to misclassify non-synchronizing examples as synchronizing, as locally observed dynamics can concentrate in a short time scale while the global dynamics do not.

Synchronization prediction accuracy for 15–30 node graphs. We apply the four binary classification algorithms for the six 15–30 node datasets described in Table 2 in order to learn to predict synchronization. Each experiment uses initial dynamics up to a variable number of training iterations r that is significantly less than the test iteration T, and the goal is to predict whether each example in the dataset is synchronized at an unseen time T. We also experiment with and without the additional graph information described in Table 2 in order to investigate the main questions Q1 and Q2, respectively. We plot prediction accuracy using four classification algorithms (RF, GB, FFNN, LRCN) and the baseline predictor versus the training iteration r, with and without the graph features. The problem of synchronization prediction becomes easier as we increase the training iteration r, as indicated by the baseline prediction accuracy in Fig. 5. For instance, for $\texttt{KM}_{30}$, there is no trivi-

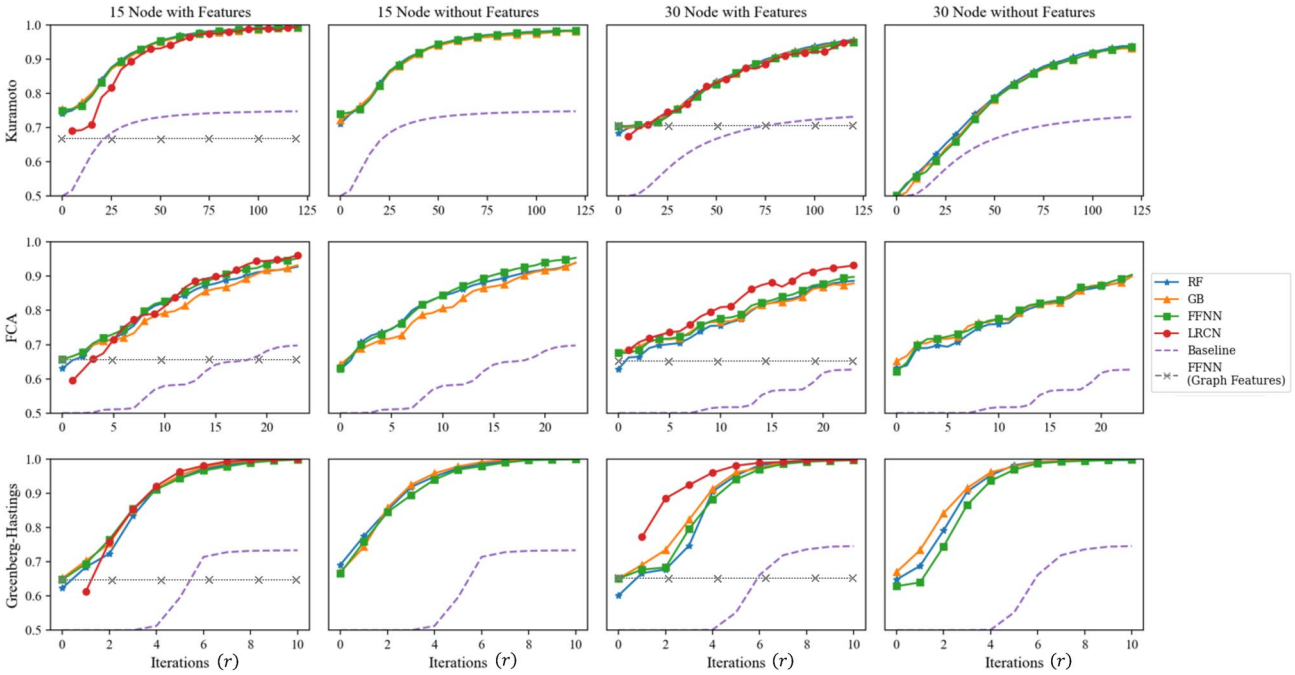

Figure 5. Synchronization prediction accuracies of four machine learning algorithms for the KM, FCA, and GHM coupled oscillators synchronization. For each of the six datasets in Table 2, we used 5-fold cross-validation with 80/20 split of test/train. Accuracy is measured by the ratio of the number of correctly classified examples to the total number of examples. Algorithms for the second and the fourth columns are trained only with dynamics up to various iterations r indicated on the horizontal axes, whereas the other two columns also use additional graph features. $r = 0$ in the "with Features" columns indicates the input consisting of initial phase configuration X_0 and its *quartiles* paired with the five graph statistics of *number of edges*, *min degree*, *max degree*, *diameter*, *number of nodes*. The gray dashed line represents training FFNN only on the five graph statistics. Hence its accuracy is constant with respect to the varying number of training iterations.

ally synchronizing example at iteration 0, and but about 10% of the synchronizing examples will have become phase-concentrated by iteration $r = 25$.

Now we discuss the results in Fig. 5. As we intended before, for the six datasets in Table 2, classifying only with the basic graph statistics achieves an accuracy of 60–70% in comparison to 100% as in the experiments in Figs. 2 and 3. This is shown by the gray horizontal lines in Fig. 5, which indicates the classification accuracy of FFNN trained only with the same five basic graph statistics used before. This indicates that we may need to use more information on the data points in order to obtain improved classification accuracy. One way to proceed is to use more graph statistics such as clustering coefficients[46], modularity[47], assortativity[48], eigenvalues of the graph Laplacian[49], etc.

Instead, aiming at investigating Question Q1, we proceed by additionally using dynamics information, meaning that we include the initial dynamics, $(X_t)_{0\le t\le r}$, up to a varying number of training iteration r. These results are shown in the first and the third columns in Fig. 5. At $r = 0$, the input consists of five graph statistics we considered before—*number of edges*, *min degree*, *max degree*, *diameter*, *number of nodes*—as well the initial configuration X_0 with its quartiles. In all cases, all four binary classifiers trained with initial dynamics significantly outperform the concentration principle baseline. The classifiers RF, GB, and FFNN show similar performance in all cases. On the other hand, the GraphLRCN in some cases outperforms the other classifiers, especially so with GHM on 30 nodes. For instance, when $r = 20$, 10 and 4 for $\mathtt{KM}_{30}$, $\mathtt{FCA}_{30}$ and $\mathtt{GHM}_{30}$, respectively, GraphLRCN achieves a prediction accuracy of 73% (baseline 55%, 1.25% concentrates), 84% (baseline 52%, 1% concentrates) and 96% (baseline 50%, 0% concentrates), respectively.

Now that we have seen that initial dynamics can be used along with the basic graph features to gain a significant improvement in classification performance, we take a step further and see how accurate we can be by only using the initial dynamics, in relation to Question Q2. That is, we drop all graph-related features from the input, and train the binary classifiers only with the initial dynamics up to varying iteration r. The results are reported in the second and the fourth columns in Fig. 5. Since now the classifiers are not given any kind of graph information at all, one might expect a significant drop in the performance. However, surprisingly, except for the dataset $\mathtt{KM}_{30}$, the classification accuracies are almost unchanged after dropping graph features altogether from training.

In addition, note that for $\mathtt{FCA}_{15}$ and $\mathtt{FCA}_{30}$, training all four classifiers on only training iterations of $r = 3$ and $r = 5$, respectively, without any graph features, produces a prediction accuracy of at least 70%. In this case, the baseline achieves only 50%, meaning that no synchronizing example is phase-concentrated by a given amount of training iterations. Similarly, for $\mathtt{KM}_{15}$, training all four classifiers on only training iterations of $r = 5$, without any graph features, produces a prediction accuracy of about 77%. In this case, the baseline achieves only 52%,

so only 10% of all synchronizing examples are phase-concentrated by iteration r (see the formula for baseline accuracy in the section on baseline predictor). This indicates that there may be some evidentiary condition for synchronization on the initial dynamics, which is *different* from half-circle concentration. A further investigation is due to exactly pin-point what such an evidentiary condition for synchronization might be.

We give multiple additional remarks on the experiments reported in Fig. 5. First, we see that the $\mathtt{KM}_{30}$ dataset is adversely affected without the use of the graph features, and beginning at $r = 0$, we only achieve 50% accuracy initially. This is further discussed in "Discussion" section along with a systematic analysis of the statistical significance of the features to the classification accuracy in more detail.

Second, the GraphLRCN binary classifier is offset with respect to the other algorithms as it is fed dynamics information encoded into the adjacency matrix of the underlying graph (see Eq. (5) in SI), so it cannot be trained only with the initial dynamics information. Oftentimes, GraphLRCN begins below the classical algorithms but is able to outperform them in intermediate iterations, but by the final training iteration, there is only a negligible difference in the performance over different classifiers.

Third, one may be concerned that the few selected graph features we use for training may not give enough information about the given classification tasks, and whether including further graph features may change the results significantly. We remark that GraphLRCN uses the entire graph adjacency matrix as part of the input (see the SI), so technically it uses every possible graph feature during training. In Fig. 5, it does show somewhat improved performance on 30 node graphs with the FCA and the GHM dynamics, but virtually identical results for the Kuramoto dynamics. On the other hand, its performance is diminished for 15 node graphs. This may be due to the fact that the training data for 15 node graphs is not rich enough to properly train GraphLRCN, which is a significantly more complex classification model combining convolutional and recurrent neural networks than the other classification models we use.

Fourth, in order to investigate the impact of the actual amount of graphs that were used in the datasets with respect to our experiments, we perform an additional experiment on the 30 node case, in which we subsample 10 K and 40 K balanced datasets from our 80 K sets to see if we get similar performance to Fig. 5, where the full datasets of 80 K examples are used. As seen in Fig. S3 in the SI, we can see that most methods show a significantly better accuracy than the baseline, with the exception of the GB and FFNN methods for $\mathtt{KM}_{30}$ without graph features. We see however that the accuracies between different methods become almost identical as we increase the number of training iterations. We note that with only a subsampled portion of the datasets being used, we see larger differences in accuracy between the methods themselves, which contrasts with Fig. 5, where all methods were rather saturated and displayed very similar accuracy across all iterations.

Synchronization prediction accuracy for 300–600 node graphs. In this section, we investigate Question Q3, which is to predict synchronization only based on local information observed on select subgraphs. Note that this is a more difficult task than Q2, since not only may we not have information about the underlying graph but we also may not have observed the entire phase configuration. For example, the dynamics may appear to be synchronized at a local scale (e.g., on 30-node connected subgraphs), but there are still large-scale waves being propagated and the global dynamics are not synchronized. Nevertheless, we can use the ensemble prediction method (Algorithm 1) to combine decisions based on each subgraph to predict the synchronization of the full graph.

In Figs. 6 and 7, we report the synchronization prediction accuracy of the ensemble predictor (Algorithm 1) on datasets $\mathtt{FCA}_{600}$, $\mathtt{FCA}'_{600}$, $\mathtt{FCA}''_{600}$, and $\mathtt{KM}_{600}$, $\mathtt{KM}'_{600}$, $\mathtt{KM}''_{600}$, respectively, described in Table 3. We used Algorithm 1 with $n_0 = 30$ (amount of nodes in the subgraphs) and $k \in \{1, 2, 4, 8\}$ (# of subgraphs). The binary classification algorithm we used is FFNN. We chose FFNN because as seen in Fig. 5, there is no significant difference in accuracies between all methods used in the 30 node case. Furthermore, we can not use the GraphLRCN model, as this method relies on knowing the dynamics and adjacency matrices of the underlying subgraphs. As the baseline, we use a slight modification of the baseline predictor from the 15–30 node classification task. Namely, we combine all phase values observed on all k subgraphs and predict synchronization if they satisfy the concentration principle otherwise we flip a fair coin. Note that the concentration of phases observed on subgraphs does *not* imply synchronization of the full system, as the full phase configuration may not be concentrated.

So far we have been using the metric of accuracy for our synchronization prediction task, which is defined as the ratio of the correctly predicted synchronizing and non-synchronizing examples to the total number of examples in the dataset. There are multiple ways that this metric could be low, for example, if an algorithm is overly conservative and misses lots of synchronizing examples or in the opposite case it may incorrectly classify lots of non-synchronizing examples. In order to provide better insights on these aspects, we also report the performance of our method (Algorithm 1) and the baseline in terms of precision and recall. Here, *precision* is formally defined as the proportion of positive classifications that are correct, or $\frac{TP}{TP+FP}$, where TP is "true positive" and FP is "false positive". In our problem, this corresponds to what proportion of predictions in synchronization was truly correct. *Recall* is formally defined as the proportion of accurately identifying true labels in the data, or $\frac{TP}{TP+FN}$, where FN is "false negative". Again with respect to our prediction problem, measuring recall corresponds to how well a given algorithm, ensemble, or baseline, correctly identifies synchronization behavior when presented with synchronizing data.

Our accuracy results for both the ensemble method and baseline are presented in Figs. 6 and 7 for Kuramoto and FCA models, respectively. In these figures, the first columns represents the results for the datasets $\{\mathtt{KM}, \mathtt{FCA}\}_{600}$ with fixed number of nodes (600) and relatively smaller variation of edge counts (std $\approx$ 20, 38, respectively); the second columns for the datasets $\{\mathtt{KM}, \mathtt{FCA}\}'_{600}$ with fixed number of nodes (600) and relatively larger variation of edge counts (std $\approx$ 80, 2371, respectively); and the third for the datasets $\{\mathtt{KM}, \mathtt{FCA}\}''_{600}$ with varied number of nodes (std $\approx$ 86) and relatively larger variation of edge counts (std $\approx$ 1461, 56, respectively).

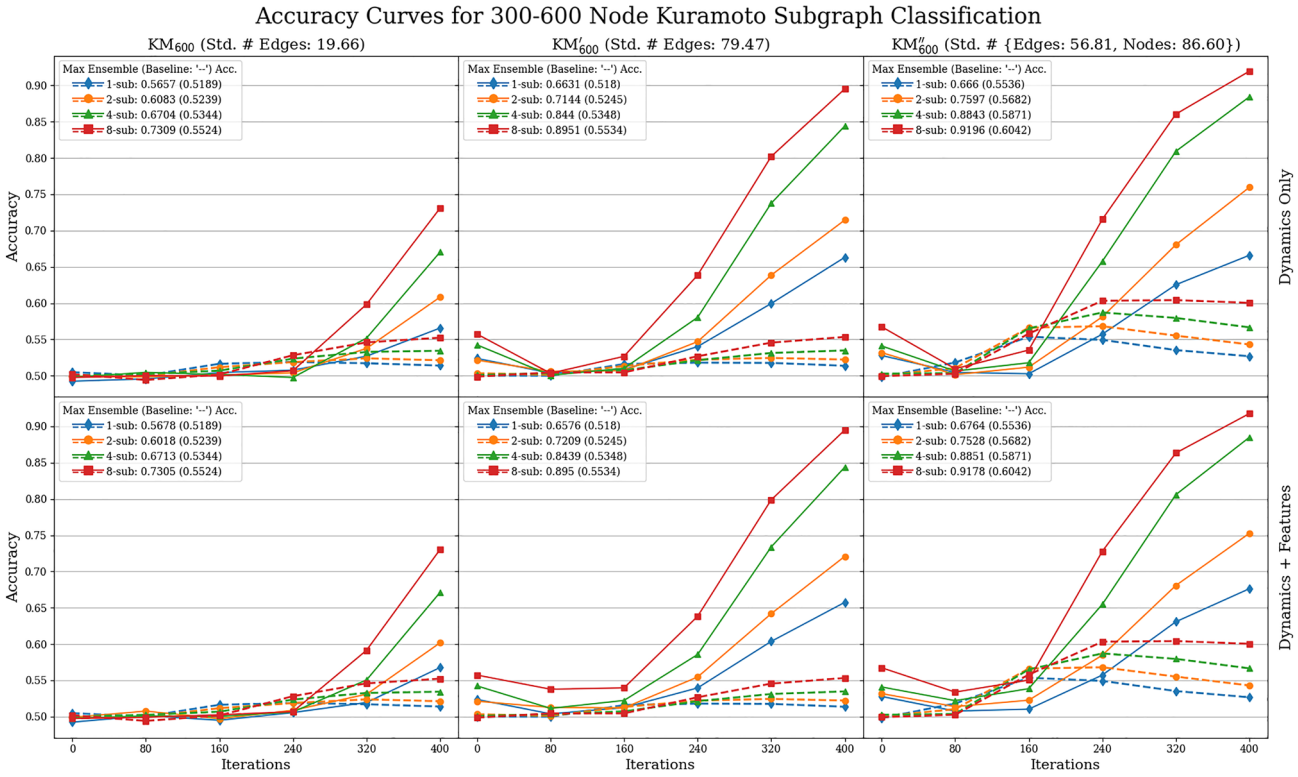

Figure 6. Accuracy curves for predicting synchronization of the Kuramoto model on 600-node graphs from dynamics observed from $k \in \{1, 2, 4, 8\}$ subgraphs of 30 nodes. All plots observe the performance of both the ensemble machine learning (solid) and baseline (dashed) accuracies over increasing amounts of training iterations $r \in \{0, 80, 240, 320, 400\}$. The first row shows results using only dynamics whereas the second row includes both the dynamics and graph features. Maximum accuracies for using k subgraphs are given by 'k-sub: Acc. (Baseline Acc.)'

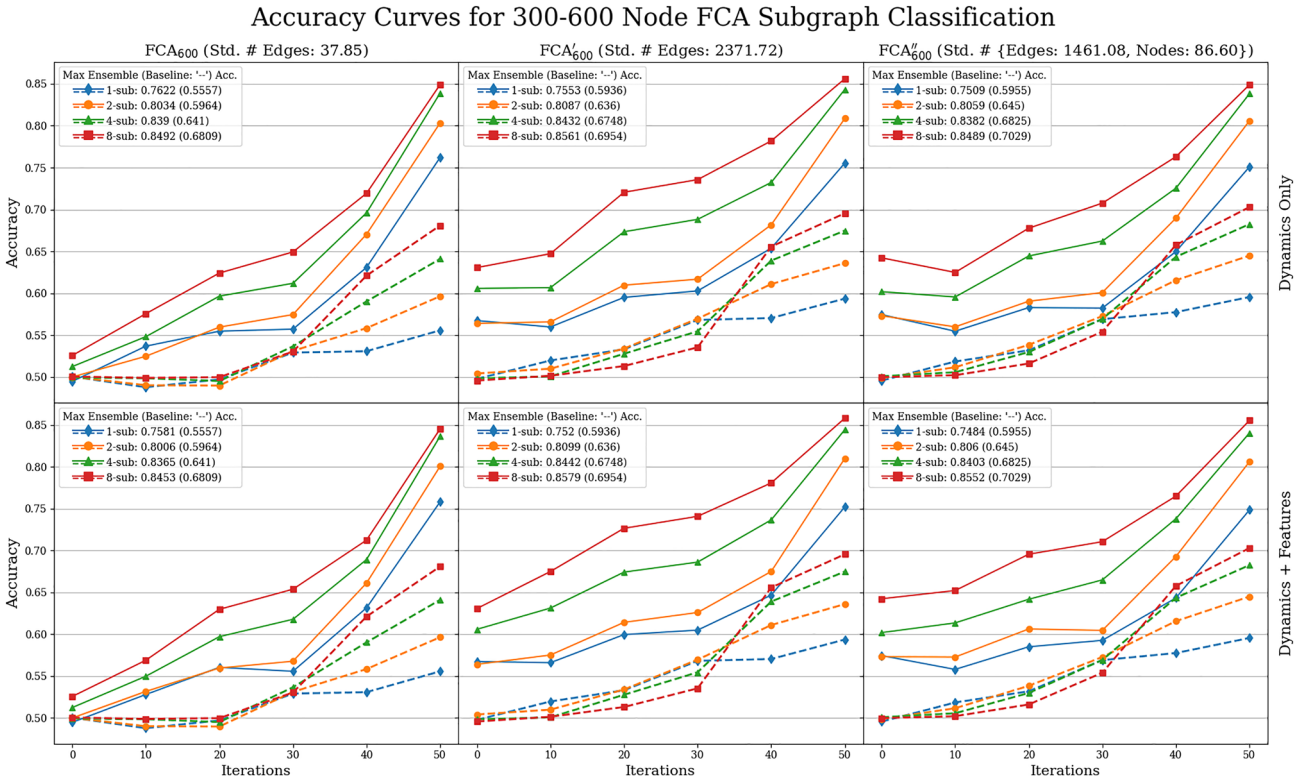

Figure 7. Accuracy curves for predicting synchronization of 5-color FCA on 600-node graphs from dynamics observed from $k \in \{1, 2, 4, 8\}$ subgraphs of 30 nodes. See the caption of Fig. 6 for details.

The first rows of both figures represent prediction accuracies using exclusively dynamics data, and the second row utilizes both dynamics and graph features. In the SI, Figs. S4 and S5 show the recall and precision curves of the ensemble and baseline methods with the same row and column orientation as the accuracy figures representing different graph datasets and subsets of features. For the Kuramoto data in Fig. 6, we applied the ensemble and baseline algorithms cumulatively up to training iterations $r = 0, 80, 160, 240, 320$ and 400; and for the FCA data in Fig. 7, we applied the ensemble and baseline algorithms cumulatively up to training iterations $r = 0, 10, 20, 30, 40$ and 50 (Note that using $r = 0$ means fitting the prediction algorithms at the initial coloring). We additionally remark that these curves were averaged over 30 train-test splits with 80% training and 20% testing.

Across all datasets considered, we see that our ensemble method consistently outperforms the baseline method in accuracy at training iterations $r = 400$ for Kuramoto and $r = 50$ for FCA. For example, across all Kuramoto datasets and feature subsets by the last iteration, $r = 400$, the ensemble method on a single subgraph outperforms the baseline algorithm on eight subgraphs; the best that the baseline algorithm does is 60.42% accuracy compared to the 91.96% for the ensemble method's best accuracy. For FCA, the baseline accuracy is 70.29% compared to the best ensemble method score of 85.79%, both on eight subgraphs. Considering the recall and the precision plots in Figs. S4 and S5 in the SI gives a more detailed explanation. Namely, the ensemble method significantly outperforms the baseline in the recall by at least 35%, whereas it performs relatively worse in precision than the baseline by at most 10% for both Kuramoto and FCA with $k = 8$ subgraphs except for the datasets $\mathtt{KM}_{600}$. This means the baseline is 'too conservative' in the sense that it misses correctly classifying a large number of synchronizing examples. From this, we deduce that a large number of synchronizing examples exhibit phase concentration over subgraphs much later in the dynamics—making early detection through phase concentration difficult. On the contrary, the high recall scores of the ensemble methods indicate that our method can still detect most of the synchronizing examples by only using local information observed on the selected subgraphs. To elaborate, the baseline only determines phase concentration at a single point in time, whereas the ensemble method is able to learn the whole variation of dynamics up to iteration r. Furthermore, between training our data on dynamics only, versus dynamics *and* graph features, across accuracy, recall, and precision curves, the inclusion of graph features hardly improves the maximum score of these values.

Finally, for both the Kuramoto model and FCA, we observe that high variation of the node and edge counts boosts all performance metrics of accuracy, recall, and precision. For example, for the recall curves we see that for varied edge count, compared to fixed edge count, the recall is considerably higher than fixed edge count by iteration $r = 400$, comparing a recall rate of 0.8755 and 0.9759 for 8 subgraphs. Furthermore, the performance gain in introducing a larger variation of node and/or edge counts is significantly larger for the Kuramoto model than for FCA. We speculate that having a larger variation of node and edge counts within the dataset presumably implies a better separation between the synchronizing and the non-synchronizing examples in the space of initial dynamics. We remark that the performance gain here is not due to a better separation between the classes in terms of the graph features, as we can see by comparing the first and the second rows of all figures (Figs. 6, 7 and also Figs. S4 and S5 in the SI).

Discussion

In Fig. 5, observe that not using the additional graph features as input (column 4) decreases the prediction accuracy from 70 to 50% for the case of $\mathtt{KM}_{30}$ at initial training iteration $r = 0$, but there is no such significant difference for the discrete models $\mathtt{FCA}_{30}$ and $\mathtt{GHM}_{30}$. In fact, in all experiments we perform in this work (those reported in Figs. 5, 6, 7 and Figs. S4 and S5 in the SI), this is only instance that we see that including the graph features during training affects the prediction accuracy.

In order to explain why this is the case, we compare the statistical significance of the all features (including the initial coloring) we use for the prediction task. We do so by computing the Gini indexes of all features by repeating the prediction experiment over 300 train/test splits of the datasets $\mathtt{KM}_{30}$, $\mathtt{FCA}_{30}$, $\mathtt{GHM}_{30}$ using GB for the choice of binary classifier. The analysis using GB will be representative of all other binary classifiers since there are negligible differences in their performance in Fig. 5.

As mentioned before, the procedure works by fitting random subsets of features and iteratively growing decision trees; it also records what is known as the Gini index[50] over each consecutive partition through multiple training iterations. As decision trees split the feature space, the Gini index measures total variance across class labels—in our case, synchronizing v.s. non-synchronizing—over each partition. Allowing the supposition that synchronization can be modeled through our graph features data using the gradient boosting method, observing the mean decrease in the Gini index across decision trees allows us to infer feature importance for synchronization over different models and graphs (see Fig. 8). See, Hartman et al.[50] for more discussions on GB and computing the Gini index.

Interestingly, the discrete models of FCA and GHM place significantly greater importance on color statistics such as the initial quartile colorings, and less importance on graph features such as diameter and minimum and maximum degree. Note that the initial color statistics can be directly computed from the initial dynamics even at training iteration $r = 0$, so this explains that there is no significant performance difference in prediction accuracy for the discrete models in Fig. 5. On the contrary, the Kuramoto model puts greater importance on graph features such as diameter and number of edges rather than the initial color statistics, and such graph features are not available from the information given by the initial dynamics at training iteration $r = 0$. From this, we can see why our algorithms show low prediction accuracy in the case of $\mathtt{KM}_{30}$. In addition, we note that there is only a negligible difference in prediction accuracy for $\mathtt{KM}_{15}$ in Fig. 5. We speculate that this is because for $\mathtt{KM}_{15}$, as we see in Table 2, having only 15 nodes does not allow to have a significant variation in the diameter and the edge counts of the graphs in the dataset.

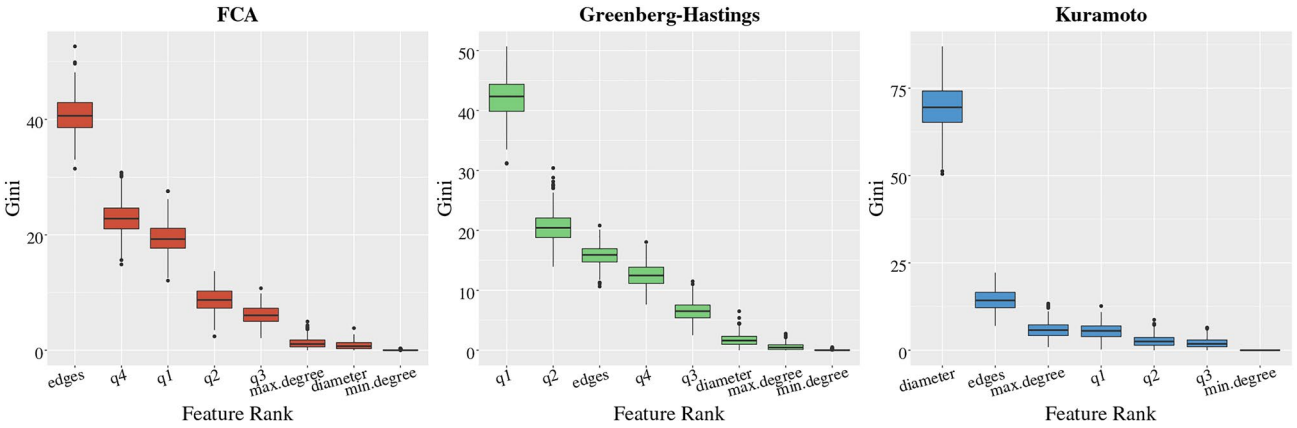

Figure 8. Boxplots of Gini index values sampled from gradient boosting procedure. Color information appears to be very important according to the distribution of Gini index values in both the FCA and GH models, discrete cellular automata models, and diameter overwhelmingly appears to have the greatest importance for Kuramoto.

Lastly, we remark that the synchronization behavior of the Kuramoto model on complete graphs is well-understood in the literature on dynamical systems[1,2,6]. In our experiments, we observed that graphs with small diameters tend to be synchronized, and this aligns with the literature since graphs with the same number of nodes but with smaller diameters are closer to complete graphs.

Conclusion

Predicting whether a given system of coupled oscillators with an underlying arbitrary graph structure will synchronize is a relevant yet analytically intractable problem in a variety of fields. In this work, we offered an alternative approach to this problem by viewing this problem as a binary classification task, where each data point consisting of initial dynamics and/or statistics of underlying graphs needs to be classified into two classes of 'synchronizing' and 'non-synchronizing' dynamics, depending on whether a given system eventually synchronizes or converges to a non-synchronizing limit cycle. We generated large datasets with non-isomorphic underlying graphs, where classification only using basic graph statistics is challenging. In this setting, we found that pairing a few iterations of the initial dynamics along with the graph statistics as the input to the classification algorithms can lead to significant improvement in accuracy; far exceeding what is known by the half-circle concentration principle in classical oscillator theory. More surprisingly, we found that in almost all such settings, dropping out the basic graph statistics and training our algorithms with only initial dynamics achieves nearly the same accuracy. Finally, we have also shown that our methods are scale well to large underlying graphs by using incomplete initial dynamics only observed on a few small subgraphs .

Drawing conclusions from our machine learning approaches to the synchronization prediction problem, we pose the following hypotheses:

- The entropy of the dynamics of coupled oscillators may decay rapidly in the initial period to the point that the uncertainty of the future behavior from an unknown graph structure becomes not significant.
- The concentration principle applied at any given time is too conservative in predicting synchronization, and there might be a generalized concentration principle that uses the whole initial dynamics, which our machine learning methods seem to learn implicitly.

Given that our machine learning approach is able to achieve high prediction accuracy, we suspect that there may be some analytically tractable characterizations on graphs paired with corresponding initial dynamics signaling eventual synchronization or not, which we are yet to establish rigorously. As mentioned at the end of "Related works" section, previously known characterizing conditions include the initial vector field on the edges induced by the initial color differential for the 3-color GHM and CCA[26], as well as the number of available states being strictly less than the maximum degree of underlying trees for FCA[10,27]. Designing similar target features into datasets and training binary classification algorithms could guide the further analytic discovery of such conditions for the coupled oscillator models considered in this work.

Furthermore, even though we have focused on predicting only two classes of the long-term behavior of complex dynamical systems as only synchronizing and non-synchronizing dynamics, our method can readily be extended to predicting an arbitrary number of classes of long-term behaviors. For instance, one can consider the κ-state voter model on graphs, where the interest would be the final dominating color. In such circumstances, one can train κ-state classification machine learning algorithms on datasets of non-isomorphic graphs. We can also consider extending our method to be able to predict synchronization on a network based on parameter control. For instance, we can train many different trajectories on a singular graph using different intrinsic frequencies in the Kuramoto model, and learn to predict what range of values of intrinsic frequencies promote synchronization.

Finally, a more ambitious task beyond long-term dynamic behavior quantified by a single metric is the potential extension of our methods to full time-series and graph state regression. In other words, if each node in the graph represents an individual in an arbitrary social network, can we predict the sentiment level for a given topic at any given time t for every single individual in that particular social network? One can again generate large overarching social networks and run many simulations of sentiment dynamics with many possible edge configurations between individuals (for example, measured by the number of mutual friends or likes/shares of posts on social media). The ultimate goal would be a framework for learning to predict, with precision, entire trajectories of complex dynamical systems.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Code availability

The codes for the main algorithm used during the current study are available in the repository https://github.com/richpaulyim/L2PSync.

Received: 1 March 2022; Accepted: 22 August 2022
Published online: 05 September 2022

References

1. Strogatz, S. H. From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators. *Phys. D* **143**, 1–20 (2000).
2. Acebrón, J. A., Bonilla, L. L., Vicente, C. J. P., Ritort, F. & Spigler, R. The Kuramoto model: A simple paradigm for synchronization phenomena. *Rev. Mod. Phys.* **77**, 137 (2005).
3. Dorfler, F. & Bullo, F. Synchronization and transient stability in power networks and nonuniform Kuramoto oscillators. *SIAM J. Control. Optim.* **50**, 1616–1642 (2012).
4. Nair, S. & Leonard, N. E. Stable synchronization of rigid body networks. *Netw. Heterogeneous Media* **2**, 597 (2007).
5. Pagliari, R. & Scaglione, A. Scalable network synchronization with pulse-coupled oscillators. *IEEE Trans. Mob. Comput.* **10**, 392–405 (2010).
6. Kuramoto, Y. *Chemical Oscillations, Waves, and Turbulence* (Courier Corporation, 2003).
7. Eom, Y.-H., Boccaletti, S. & Caldarelli, G. Concurrent enhancement of percolation and synchronization in adaptive networks. *Sci. Rep.* **6**, 1–7 (2016).
8. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U. Complex networks: Structure and dynamics. *Phys. Rep.* **424**, 175–308 (2006).
9. Chowdhury, S. N., Ghosh, D. & Hens, C. Effect of repulsive links on frustration in attractively coupled networks. *Phys. Rev. E* **101**, 022310 (2020).
10. Lyu, H. Synchronization of finite-state pulse-coupled oscillators. *Phys. D* **303**, 28–38 (2015).
11. McKay, B. Graphs. http://users.cecs.anu.edu.au/~bdm/data/graphs.html.
12. Greenberg, J. M. & Hastings, S. P. Spatial patterns for discrete models of diffusion in excitable media. *SIAM J. Appl. Math.* **34**, 515–523 (1978).
13. Fan, H., Kong, L.-W., Lai, Y.-C. & Wang, X. Anticipating synchronization with machine learning. *Phys. Rev. Res.* **3**, 023237 (2021).
14. Guth, S. & Sapsis, T. P. Machine learning predictors of extreme events occurring in complex dynamical systems. *Entropy* **21**, 925 (2019).
15. Chowdhury, S. N., Ray, A., Mishra, A. & Ghosh, D. Extreme events in globally coupled chaotic maps. *J. Phys. Complex.* **2**, 035021 (2021).
16. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
17. Thiem, T. N., Kooshkbaghi, M., Bertalan, T., Laing, C. R. & Kevrekidis, I. G. Emergent spaces for coupled oscillators. *Front. Comput. Neurosci.* **14**, 36 (2020).
18. Bishop, C. M. & Nasrabadi, N. M. *Pattern Recognition and Machine Learning* Vol. 4 (Springer, 2006).
19. Biccari, U. & Zuazua, E. A stochastic approach to the synchronization of coupled oscillators. *Front. Energy Res.* **8**, 115 (2020).
20. Hefny, A., Downey, C. & Gordon, G. J. Supervised learning for dynamical system learning. *Adv. Neural Inf. Process. Syst.* **28**, 25 (2015).
21. Itabashi, K., Tran, Q. H. & Hasegawa, Y. Evaluating the phase dynamics of coupled oscillators via time-variant topological features. *Phys. Rev. E* **103**, 032207 (2021).
22. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).
23. Friedman, J. H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **38**, 367–378 (2002).
24. Donahue, J. *et al.* Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2625–2634 (2015).
25. Fisch, R. Cyclic cellular automata and related processes. *Phys. D* **45**, 19–25 (1990).
26. Gravner, J., Lyu, H. & Sivakoff, D. Limiting behavior of 3-color excitable media on arbitrary graphs. *Ann. Appl. Probab.* **28**, 3324–3357 (2018).
27. Lyu, H. Phase transition in firefly cellular automata on finite trees. arXiv:1610.00837 (arXiv preprint) (2016).
28. Lyu, H. & Sivakoff, D. Persistence of sums of correlated increments and clustering in cellular automata. *Stoch. Processes Appl.* **129**, 1132–1152 (2019).
29. Fisch, R. The one-dimensional cyclic cellular automaton: A system with deterministic dynamics that emulates an interacting particle system with stochastic dynamics. *J. Theor. Probab.* **3**, 311–338 (1990).
30. Fisch, R., Gravner, J. & Griffeath, D. Cyclic cellular automata in two dimensions. In *Spatial Stochastic Processes* 171–185 (Springer, 1991).
31. Fisch, R. Clustering in the one-dimensional three-color cyclic cellular automaton. *Ann. Probab.* **20**, 1528–1548 (1992).
32. Durrett, R. & Steif, J. E. Some rigorous results for the Greenberg–Hastings model. *J. Theor. Probab.* **4**, 669–690 (1991).
33. Lyu, H. Global synchronization of pulse-coupled oscillators on trees. *SIAM J. Appl. Dyn. Syst.* **17**, 1521–1559 (2018).
34. Nishimura, J. & Friedman, E. J. Robust convergence in pulse-coupled oscillators with delays. *Phys. Rev. Lett.* **106**, 194101 (2011).
35. Klinglmayr, J., Kirst, C., Bettstetter, C. & Timme, M. Guaranteeing global synchronization in networks with stochastic interactions. *New J. Phys.* **14**, 073031 (2012).
36. Proskurnikov, A. V. & Cao, M. Synchronization of pulse-coupled oscillators and clocks under minimal connectivity assumptions. *IEEE Trans. Autom. Control* **62**, 5873–5879 (2016).

37. Nunez, F., Wang, Y. & Doyle, F. J. Synchronization of pulse-coupled oscillators on (strongly) connected graphs. *IEEE Trans. Autom. Control* **60**, 1710–1715 (2014).
38. Moreau, L. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. Autom. Control* **50**, 169–182 (2005).
39. Papachristodoulou, A., Jadbabaie, A. & Münz, U. Effects of delay in multi-agent consensus and oscillator synchronization. *IEEE Trans. Autom. Control* **55**, 1471–1477 (2010).
40. Chazelle, B. The total s-energy of a multiagent system. *SIAM J. Control. Optim.* **49**, 1680–1706 (2011).
41. Newman, M. E., Watts, D. J. & Strogatz, S. H. Random graph models of social networks. *Proc. Natl. Acad. Sci.* **99**, 2566–2572 (2002).
42. Hagberg, A., Swart, P. & Chult, S. D. Exploring network structure, dynamics, and function using networkx. Tech. Rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008).
43. Lee, S. & Lister, R. Experiments in the dynamics of phase coupled oscillators when applied to graph colouring. In *Conferences in Research and Practice in Information Technology Series* (2008).
44. Kassabov, M., Strogatz, S. H. & Townsend, A. Sufficiently dense Kuramoto networks are globally synchronizing. *Chaos Interdiscip. J. Nonlinear Sci.* **31**, 073135 (2021).
45. Durrett, R. & Griffeath, D. Asymptotic behavior of excitable cellular automata. *Exp. Math.* **2**, 183–208 (1993).
46. Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature* **393**, 440–442 (1998).
47. Newman, M. E. Spectral methods for community detection and graph partitioning. *Phys. Rev. E* **88**, 042822 (2013).
48. Allen-Perkins, A., Pastor, J. M. & Estrada, E. Two-walks degree assortativity in graphs and networks. *Appl. Math. Comput.* **311**, 262–271 (2017).
49. Zhang, X.-D. The laplacian eigenvalues of graphs: A survey. arXiv:1111.2897 (arXiv preprint) (2011).
50. Hartmann, C., Varshney, P., Mehrotra, K. & Gerberich, C. Application of information theory to the construction of efficient decision trees. *IEEE Trans. Inf. Theory* **28**, 565–577 (1982).

Acknowledgements

This work is supported by NSF Grant DMS-2010035. We are also grateful for partial support from and the Department of Mathematics and the Physical Sciences Division at UCLA. We also thank Andrea Bertozzi and Deanna Needell for support and helpful discussions. JV is partially supported by Deanna Needell's grant NSF BIGDATA DMS #1740325.

Author contributions

H.B. and R.P.Y. ran simulations and experiments. H.L. led the experiment design. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-18953-8.

Correspondence and requests for materials should be addressed to H.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.