Inferring missing edges in a graph from observed collective patterns

Selim Haj Ali and Marc-Thorsten Hütt*
Department of Life Sciences and Chemistry, Jacobs University Bremen, D-28759 Bremen, Germany

(Received 10 February 2022; accepted 26 May 2022; published 27 June 2022)

Many real-life networks are incomplete. Dynamical observations can allow estimating missing edges. Such procedures, often summarized under the term 'network inference', typically evaluate the statistical correlations among pairs of nodes to determine connectivity. Here, we offer an alternative approach: completing an incomplete network by observing its *collective* behavior. We illustrate this approach for the case of patterns emerging in reaction-diffusion systems on graphs, where collective behaviors can be associated with eigenvectors of the network's Laplacian matrix. Our method combines a partial spectral decomposition of the network's Laplacian matrix with eigenvalue assignment by matching the patterns to the eigenvectors of the incomplete graph. We show that knowledge of a few collective patterns can allow the prediction of missing edges and that this result holds across a range of network architectures. We present a numerical case study using activator-inhibitor dynamics and we illustrate that the main requirement for the observed patterns is that they are not confined to subsets of nodes, but involve the whole network.

DOI: 10.1103/PhysRevE.105.064610

I. INTRODUCTION

At the core of network science [1] is the representation of large amounts of data in the form of mathematical graphs. Data about the binding of proteins yield protein-interaction networks [2,3], contact data and surveys of behavioral patterns yield social networks [4,5], compilations of transcription factor binding sites can be summarized as transcriptional regulatory networks [6], connectivities of brain areas can be estimated from measured asymmetries in local water diffusion via diffusion tensor imaging (DTI) leading to connectome representations of biological neural networks [7,8].

It is by now widely accepted that most network representations of biological, social, and technical networks are incomplete [6,9–15]. The most common type of incompleteness is to be equipped with complete (or near-complete) knowledge of the nodes of the network (e.g., having an inventory of all genes of an organism or all social actors in a community), but only with incomplete information on their connections (e.g., the regulatory processes among genes or the acts of communication or interactions among social actors). A large body of algorithms predicts missing edges either evaluating the plausibility of certain topological features (like the degree of a node or its betweenness centrality) or similarity of the network to a particular class of generic network topologies (e.g., hierarchical modular networks) or via the embedding of the network in a geometric (e.g., three-dimensional) space exploiting spatial proximity for prediction purposes (see also Ghasemian *et al.* [15] for details of these categories and a clear summary and comparative analysis of these algorithms). Examples of such edge prediction algorithms are [10,16–20].

Often, dynamical information of nodes in a network is analyzed in one of two ways: (i) A set such dynamical 'snapshots' is converted into pairwise relationships among nodes, thus leading to a network representation of this dynamical information, which is then thought of as an approximation of the (e.g., regulatory) interaction structure underlying the system at hand and which is used to determine missing edges [14,21–24]. This is the core of the rich topic of *network inference* [25,26]. (ii) The dynamical information is represented in a network-like fashion (often called *functional connectivity* and then compared to a given (structural) network. This is the core idea of comparisons of structural and functional connectivity (SC/FC relationships), as frequently employed in computational neuroscience [27] and other disciplines [28].

Beyond the research described above, the general challenge posed by incomplete networks has led to a range of other investigations, such as the question of accurate community detection or centrality in incomplete or erroneous networks [29,30] or the reconstruction of topological and dynamical information when only incomplete dynamical information is available, as in compressive sensing approaches [31]. Other recent methods leverage observed heterogeneous patterns of activity of nonlinear random walks [32,33] and spiking neuron models [34,35] to infer on the underlying network topology (see also [36] for details on patterns on graphs based on random walks). In [14], a formalism is presented for estimating network structure from noisy and erroneous data in cases where additional information is provided. Newman [14] discusses how to leverage this information, in order to maximise the agreement between network structure and data, using a variant of Bayesian statistics.

Here, we draw attention to an underrated source of information – collective patterns – about missing edges in a network and more generally we construct a framework for a pattern-based network inference. Our focus is on a special case associated for example with Turing patterns on graphs [37–39], where (near the Turing instability) these collective patterns are proportional to eigenvectors of the Laplacian matrix. By simulating a reaction-diffusion system on a graph, we show how this approach can be operationalized in practice.

*Correspondending author: m.huett@jacobs-university.de

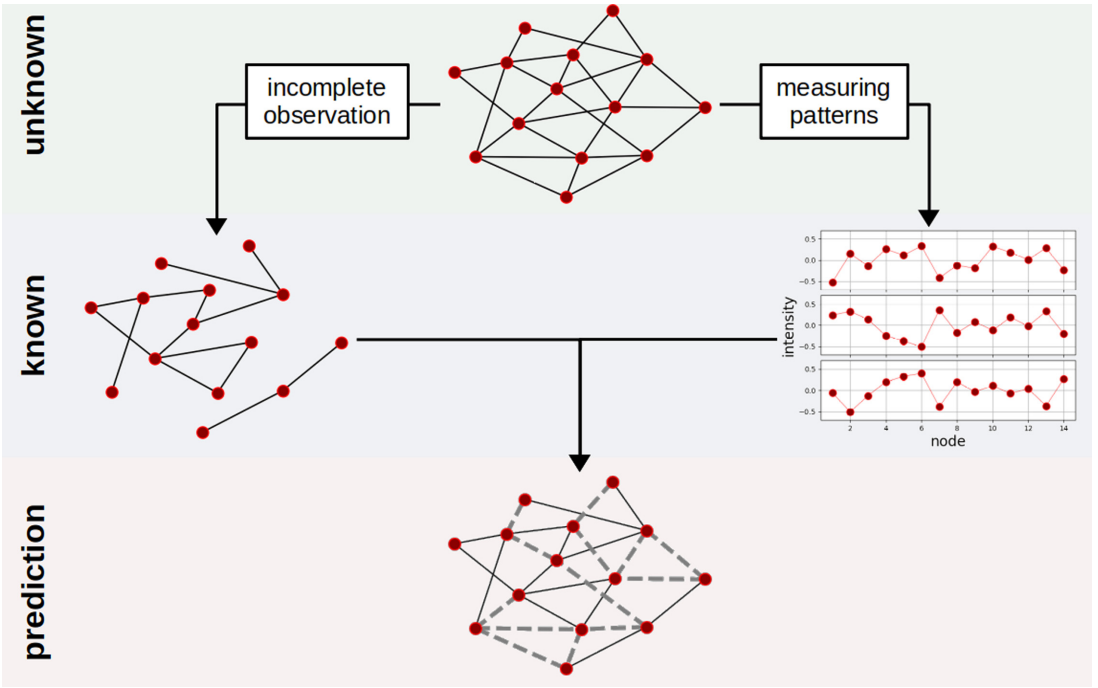

FIG. 1. Schematic illustration of the pattern-based network inference method. Collective (reaction-diffusion) patterns emerge as a dynamical process on the original graph (unknown) and bear the imprints of its (Laplacian) eigenvectors. Measuring the former and using them as a proxy for the latter, and adding to it the partial knowledge of the graph edges (known), we are able to infer the full network topology (prediction).

Pattern forming capabilities of complex networks unites two fields of fundamental relevance to statistical physics, namely the theory of complex networks and the theory of self-organization and spatiotemporal pattern formation. Prominent examples of this combination of fields are Turing patterns on graphs [37–39] and self-organized waves in excitable networks [40–42].

Only few investigations consider the perspective of dynamical systems to enhance edge prediction of networks, for example leveraging recorded statistics of time invariants [43], response patterns [44,45], or entire dynamical trajectories [46].

In the following, we will first present the technicalities involved in our method, including the eigenvalue assignment procedure and the formalism undertaken for the simulation of Turing patterns. We show how we can reliably infer the missing edges of a partially unknown graph from the incomplete graph together with the knowledge of a few eigenvectors of the corresponding original graph. In a subsequent step, we reconstruct the original graph using Turing patterns instead of the eigenvectors of the latent network, Fig. 1 summarizes our approach. Further, we assess the role of network architecture, the importance of eigenvector ranks, measure and explain the performance of our method as a function of the amount of missing information. Later, we finish by discussing its strengths, identify its limitations and propose future directions to explore.

II. METHODS

A. Notation

Let $G = (\mathcal{N}, \mathcal{E})$ be an undirected unweighted graph with $N = |\mathcal{N}|$ nodes and $E = |\mathcal{E}|$ edges with no self-loops or multiple edges. For $i = 1 \ldots N$, the degree of the node i is denoted as k_i. The adjacency matrix of G is an $N \times N$ symmetric matrix denoted as $A = (a_{ij})$ such that $i, j = 1 \ldots N$, where $a_{ij} = 1$ if j is adjacent to i and $a_{ij} = 0$ otherwise. Hence, $k_i = \sum_{j=1}^{N} a_{ij}$. The Laplacian matrix of G is denoted as $L = (l_{ij})$ and is defined as $l_{ij} = a_{ij} - k_i \delta_{ij}$. Via an eigenvector equation $\sum_{j=1}^{N} l_{ij}\phi_{j_\alpha} = \Lambda_\alpha \phi_{i_\alpha}$, we can obtain Λ_α, the real negative Laplacian eigenvalue associated to the eigenvector $\vec{\phi}_\alpha = (\phi_{1\alpha}, \ldots, \phi_{N\alpha})$. The indices α are sorted in increasing order of the eigenvalues such that $\Lambda_1 \leqslant \Lambda_2 \leqslant \ldots \leqslant \Lambda_N = 0$. For $\alpha, \beta = 1 \ldots N$, all $\vec{\phi}_\alpha$ and $\vec{\phi}_\beta$ are orthonormal such that $\sum_{i=1}^{N} \phi_{i_\alpha}\phi_{i_\beta} = \delta_{\alpha\beta}$.

After deleting pE edges from G with $0 < p < 1$, we are left with a graph denoted as $G^{(D)} = (\mathcal{N}, \mathcal{E}')$ with an adjacency matrix $A^{(D)}$ and a Laplacian matrix $L^{(D)}$ which has for eigenvalues $\Lambda_\alpha^{(D)}$ and eigenvectors $\vec{\phi}_\alpha^{(D)}$ for $\alpha = 1 \ldots N$.

Note that, while some eigen*vectors* of the original graph are known (in terms of observed collective patterns), the eigen*values* of the original graph are *not* known. In fact, a core component of our approach is to estimate these quantities from the eigenvalues and eigenvectors of the incomplete graphs (via eigenvector matching; see Sec. II C). Table I summarizes our starting point.

B. Spectral decomposition

G is reconstructible [47,48] and its Laplacian L has a spectral decomposition such that

$$L = \sum_{\alpha=1}^{N} \Lambda_\alpha \vec{\phi}_\alpha \vec{\phi}_\alpha^{T}. \tag{1}$$

TABLE I. Information availability in our investigation.

Quantity	Symbol	Known/unknown
original graph	$G = (\mathcal{N}, \mathcal{E})$	partially known (via $G^{(D)}$)
incomplete graph	$G^{(D)} = (\mathcal{N}, \mathcal{E}')$	known
eigenvectors of G	$\vec{\phi}_\alpha$	partially known
eigenvalues of G	Λ_α	unknown
eigenvectors of $G^{(D)}$	$\vec{\phi}^{(D)}_\alpha$	known
eigenvalues of $G^{(D)}$	$\Lambda^{(D)}_\alpha$	known

Since L is a negative semidefinite matrix, the coefficient Λ_α associated to $\vec{\phi}_\alpha$ in the eigen-decomposition always keeps the same negative sign for $\alpha = 1 \ldots N$.

C. Matching coefficients

In order to select coefficients of the truncated spectral decomposition, we need to find the eigenvalue among the $\Lambda^{(D)}_\alpha$, which is likely to be most similar to the (unknown) eigenvalue Λ_α corresponding to the eigenvector $\vec{\phi}_\alpha$ of the original network (which is given as an observed pattern). This is done by computing the (Pearson) correlation coefficient between the eigenvector $\vec{\phi}_\alpha$ and all N eigenvectors $\vec{\phi}^{(D)}_\beta$. The index β^*, which gives the maximal correlation with the given eigenvector $\vec{\phi}_\alpha$, is then allowing us to select $\Lambda^{(D)}_{\beta^*}$ as the missing coefficient of $\vec{\phi}_\alpha$ in the truncated spectral decomposition.

For $\alpha = 1 \ldots N$, we estimate the Pearson correlation coefficients [49]

$$C\left(\vec{\phi}_\alpha, \vec{\phi}^{(D)}_\beta\right) = \frac{N \sum_i \phi_{i_\alpha} \phi^{(D)}_{i_\beta} - \sum_i \phi_{i_\alpha} \sum_i \phi^{(D)}_{i_\beta}}{\sqrt{N \sum_i (\phi_{i_\alpha})^2 - \left(\sum_i \phi_{i_\alpha}\right)^2} \sqrt{N \sum_i \left(\phi^{(D)}_{i_\beta}\right)^2 - \left(\sum_i \phi^{(D)}_{i_\beta}\right)^2}}$$

between $\vec{\phi}_\alpha$ and all N $\vec{\phi}^{(D)}_\beta$ for $\beta = 1 \ldots N$ and select the maximally correlated $\vec{\phi}_{\beta^*}$. After matching each α with the corresponding $\beta^* = \text{argmax}_\beta |C(\vec{\phi}_\alpha, \vec{\phi}^{(D)}_\beta)|$, we write the inferred Laplacian $L^{(I)} = (l^{(I)}_{ij})$ according to

$$L^{(I)} = \Theta\left(\sum_{\alpha=1}^{N} \Lambda^{(D)}_{\beta^*} \vec{\phi}_\alpha \vec{\phi}^T_\alpha, \tau\right), \tag{2}$$

where Θ is the Heaviside function applied element-wise such that $\Theta(l_{ij}, \tau) = \{^{1 \quad \text{if } l_{ij} \geqslant \tau}_{0 \quad \text{if } l_{ij} < \tau}$ with τ serving as binarization threshold.

After estimating $L^{(I)}$ and its corresponding adjacency matrix $A^{(I)} = (a^{(I)}_{ij}) = l^{(I)}_{ij}(1 - \delta_{ij})$, we fully reconstruct the graph denoted as $G^{(R)}$ by writing its adjacency matrix denoted as $A^{(R)}$ as a combination of $A^{(I)}$ and $A^{(D)}$. $A^{(R)} = (a^{(R)}_{ij})$ such that

$$a^{(R)}_{ij} = \begin{cases} 1 & \text{if } a^{(I)}_{ij} = 1 \ \vee \ a^{(D)}_{ij} = 1 \\ 0 & \text{if } a^{(I)}_{ij} = 0 \ \wedge \ a^{(D)}_{ij} = 0 \end{cases}.$$

D. Pairwise correlations from eigenvectors

For comparison of pattern view with standard inference methods based on pairs of nodes, we use the following correlation-based classifier: We extract pairwise correlations from the eigenvectors, or interchangeably from measured Turing patterns, and use them to reconstruct the original graph. Having $\{\vec{\phi}_\alpha\}$, a set of k eigenvectors, as a starting point, for every pair of nodes $i, j = 1 \ldots N$, we compute the Pearson correlation $C(\vec{\varphi}_i, \vec{\varphi}_j)$ such that $\vec{\varphi}_i = (\phi_{i_1}, \phi_{i_2}, \ldots, \phi_{i_k})$ a vector made of the ith component of every eigenvector available.

Shuffling n_S times every vector $\vec{\varphi}_i$ and $\vec{\varphi}_j$ into $\vec{\varphi}^{(s_i)}_i$ and $\vec{\varphi}^{(s_j)}_j$ with $s_i, s_j = 1 \ldots n_S$, we compute a list of mutual correlation coefficients of length $n_S \times n_S$ and z-score the corresponding $C(\vec{\varphi}_i, \vec{\varphi}_j)$ according to it. We construct a matrix $A^{(P)} = (a^{(P)}_{ij})$, where $a^{(P)}_{ij}$ is the aforementioned z-transformed pairwise correlation coefficient between i and j. After applying a binarization threshold and including the information from $A^{(D)}$, we obtain an adjacency matrix for $G^{(R)}$.

E. Turing patterns

Turing patterns emerge from reaction-diffusion systems [50] described on a graph $G = (\mathcal{N}, \mathcal{E})$ by a set of $2N$ equations modeling the populations of activators and inhibitors [51,52] such that in every node $i = 1 \ldots N$ [37], the following dynamical equations hold:

$$\frac{\partial}{\partial t} u_i(t) = f(u_i, v_i) + \epsilon \sum_{j=1}^{N} L_{ij} u_j, \tag{3}$$

$$\frac{\partial}{\partial t} v_i(t) = g(u_i, v_i) + \sigma\epsilon \sum_{j=1}^{N} L_{ij} v_j. \tag{4}$$

The diffusive mobility of the activator and the inhibitor species are given by $D_{\text{act}} = \epsilon$ and $D_{\text{inh}} = \sigma\epsilon$, respectively. Hence, σ is the ratio of the inhibitor and activator diffusion coefficients, $\sigma = D_{\text{inh}}/D_{\text{act}}$. The functions $f(u, v)$ and $g(u, v)$ represent the local interaction terms of both species. $f(\bar{u}, \bar{v}) = 0$ and $g(\bar{u}, \bar{v}) = 0$ correspond to the uniform stationary state $(u, v) = (\bar{u}, \bar{v})$. In the case of the Gierer-Meinhardt model [52], the reaction terms are formulated with two kinetic parameters a and b as $f(u, v) = a - bu + u^2/v$ and $g(u, v) = u^2 - v$, and for which $(\bar{u}, \bar{v}) = (\frac{a+1}{b}, (\frac{a+1}{b})^2)$.

Then, following the same formalism described by Nakao and Mikhailov [37], we obtain the growth rates λ_α as

$$\lambda_\alpha = \frac{1}{2}\left\{ -b + \frac{2b}{a+1} + \epsilon\Lambda_\alpha(1+\sigma) + \sqrt{\frac{8b}{a+1} + \left[1 - b + \frac{2b}{a+1} + \epsilon\Lambda_\alpha(1-\sigma)\right]^2} \right\}.$$

To excite a narrow range of the graph eigenvectors around the mode α corresponding to $\vec{\phi}_\alpha$, we follow the procedure from Hütt *et al.* [38] and tune all four parameters, a, b, ϵ, and σ

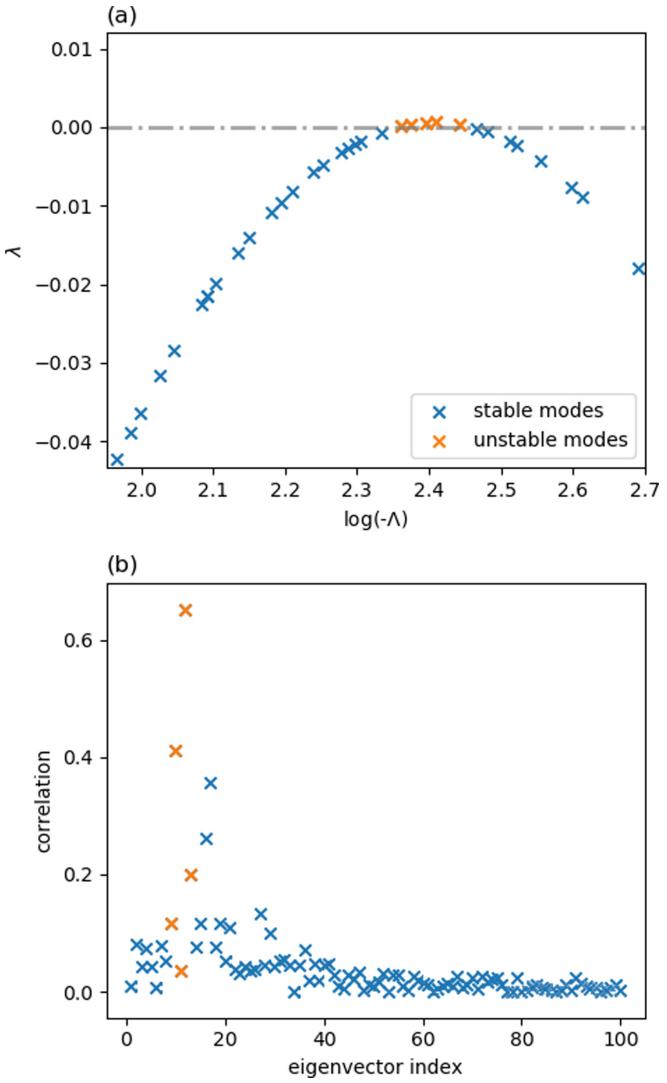

FIG. 2. (a) Dispersion relation for Gierer-Meinhardt (reaction-diffusion) dynamics on an Erdös-Rényi graph of $N = 100$ nodes and $E = 300$ edges, using the same parameters as in Fig. 5. The dispersion relation indicates the relation between the growth rates λ (representing dynamics) and the graph eigenvalues Λ (representing network topology). In this case, Turing patterns are driven by the five unstable modes $\alpha = \{9, \ldots, 13\}$ verifying $\lambda_\alpha > 0$. (b) Distribution of the correlations between the graph eigenvectors and a single Turing pattern for the same graph and parameters as in (a), as a function of the corresponding eigenvector index.

so that λ_α is located on top of the dispersion curve given by λ_α as a function of $\log(-\Lambda)$ (see Fig. 2(a)). This is done by first solving the system of equations $\{\lambda_\alpha = 0, \frac{\partial\lambda_\alpha}{\partial\Lambda_\alpha} = 0\}$ and then, incrementing σ by $\Delta\sigma$ to translate the curve upward and be in the unstable regime. Note that the term $\partial\lambda_\alpha/\partial\Lambda_\alpha$ is meant in the following way: The mathematical expression yields a curve $\lambda(\Lambda)$ and the actual eigenvalues Λ_α constitute a discrete sampling of this curve (see [38] for a detailed discussion).

Projected on the $\{\vec{\phi}_\alpha\}$ basis for $\alpha = 1 \ldots N$, $u_i(t)$ time evolution is proportional to e^{λ_α}. Thus, only the unstable terms α_c with $\lambda_{\alpha_c} > 0$ do not vanish and contribute to the stationary state, that is, the Turing pattern $\vec{u} = (u_1, \ldots, u_N) \propto \vec{\phi}_\alpha$ for all $\lambda_\alpha > 0$. The correlations between $\vec{u}$ (i.e., the concentration of activators per node) representing the patterns, and the eigenvectors $\vec{\phi}_\alpha$ are shown in Fig. 2(b).

TABLE II. ROC confusion matrix adapted to the network inference problem.

if $a_{ij}^{(D)} = 0$	$a_{ij}^{(O)} = 1$	$a_{ij}^{(O)} = 0$	
$a_{ij}^{(I)} = 1$	True Positive	False Positive	True Positive Rate: $TPR = \frac{TP}{TP+FN}$
$a_{ij}^{(I)} = 0$	False Negative	True Negative	False Positive Rate: $FNR = \frac{FN}{FN+TN}$

III. RESULTS

As schematically illustrated in Fig. 1, augmenting an incomplete knowledge of the network structure by the measurement of a few Turing patterns, we can construct a partial spectral decomposition of the graph's Laplacian matrix and, in this way, infer its missing edges. Detailed characterization and explanation of this procedure, including the coefficients matching process are given in Sec. III C.

The graph reconstruction problem can be represented as a set of binary classifications, every pair of nodes can either be linked by an edge (returns 1) or not (returns 0). A common method to evaluate the performance of a classifier is by plotting the receiver operating characteristic (ROC) curve which describes the variation of the true positive rate (TPR) as a function of the false positive rate (FPR) over a range of decision thresholds. Then, by computing the area under the ROC curve (denoted as AUC [53]), we obtain the probability for a classifier to rank a randomly chosen positive case higher than a randomly chosen negative one. The quality of our inference is assessed by comparing the original adjacency matrix $A = (a_{ij}^{(O)})$ to the reconstructed adjacency matrix $A^{(R)} = (a_{ij}^{(R)})$, the edges already present in $A^{(D)} = (a_{ij}^{(D)})$ being excluded from the evaluation, according to Table II.

In the case, where an edge exists between i and j, we are faced with two possibilities, when going from the original network (where $a_{ij} = 1$) to the deleted network: (i) The edge has not been deleted and we can recover it given that we observe $a_{ij}^{(D)} = 1$ (even if we infer $a_{ij}^{(I)} = 0$). (ii) The edge has been deleted ($a_{ij}^{(D)} = 0$) and we can recover it, in case we infer that $a_{ij}^{(I)} = 1$. Hence, the logical statement $(a_{ij}^{(D)} = 1 \vee a_{ij}^{(I)} = 1)$ (i.e., a logical OR) covers the two cases for which we can conclude that $a_{ij}^{(R)} = 1$ is the correct assumption.

We consider an Erdös-Rényi graph G of $N = 100$ nodes and $E = 300$ edges, and attempt reconstructing it after random deletion of 50% of all its edges based on the information contained in different sets of its eigenvectors. The resulting ROC curves are given in Fig. 3. The role of the eigenvector rank is later explored in Sec. III C.

A. Inferring from eigenvectors

The outlined method stands out by its resilience to topological damage as observed in Fig. 4 depicting, for three Erdös-Rényi graphs G_i of $N = 100$ nodes and $E_i = 250, 300, 350$ edges, the evolution of the AUC – that is, the

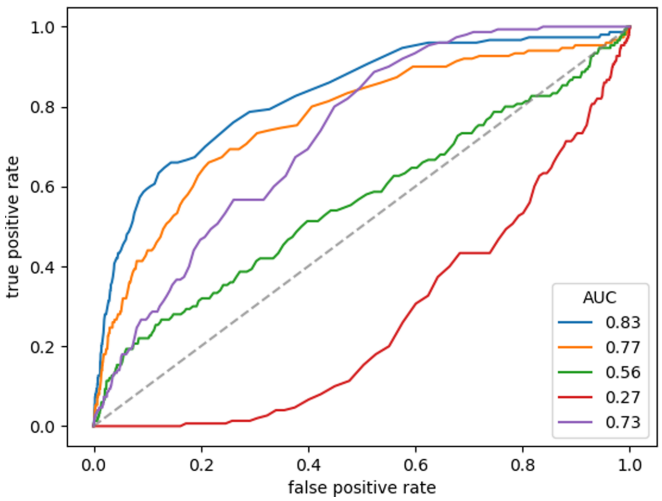

FIG. 3. ROC curves of the reconstruction of an Erdös-Rényi graph G of $N = 100$ nodes and $E = 300$ edges. We infer the original graph structure after random deletion of 50% of all network edges, using: a set of five nonadjacent eigenvectors from the intervals $\vec{\phi}_3 \dots \vec{\phi}_{17}$ (blue curve with $AUC_b = 0.81$), $\vec{\phi}_8 \dots \vec{\phi}_{22}$ (orange curve with $AUC_o = 0.76$), $\vec{\phi}_{31} \dots \vec{\phi}_{45}$ (green curve with $AUC_g = 0.56$), $\vec{\phi}_{84} \dots \vec{\phi}_{98}$ (red curve with $AUC_r = 0.27$) or the latter set after swapping *edges* and *nonedges* of the inferred graph adjacency matrix (purple curve with $AUC_p = 0.73$). The grey line indicates random predictions, curves covering an area superior (respectively, inferior) to it, meaning $AUC > 0.5$ (respectively, $AUC < 0.5$) correspond to better (respectively, worse) than random predictions.

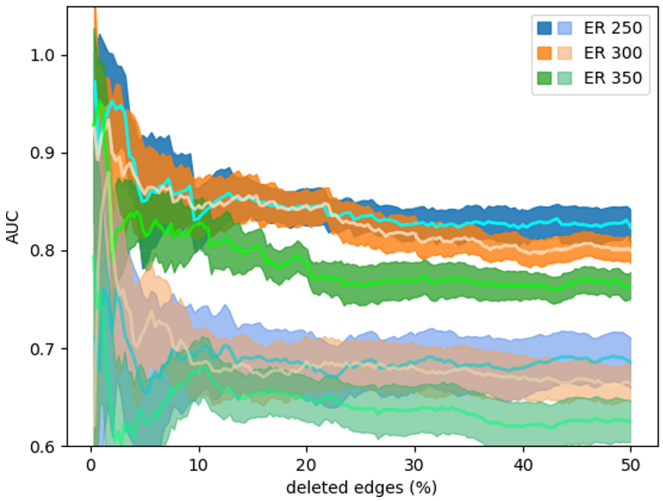

FIG. 4. AUC estimation of the reconstruction of three Erdös-Rényi graphs G_i of $N = 100$ nodes and $E_i = 250, 300, 350$ edges as a function of the percentage of deleted edges. In each case and for a $n_R = 100$ iterations each, we use a set of five eigenvectors randomly picked from the interval $\vec{\phi}_3 \dots \vec{\phi}_{17}$. We infer the original graph structures after random deletion of a single edge and up to 50% of all network edges. For each graph, the resulting AUCs are depicted as bands (standard deviation) of which average is indicated by a line within each band. These values are compared with those drawn from pairwise correlations, depicted in corresponding pale colors.

accuracy of our predictions – as a function of the percentage of deleted edges. As a proof of principle, in each of these cases, the AUC is estimated starting from $n_R = 100$ different selections of a set of five eigenvectors randomly picked, in this case, from the interval $\vec{\phi}_3 \dots \vec{\phi}_{17}$.

For comparison, we use a naive classifier where the eigenvectors are not considered as 'patterns', but rather as 'data', from which we compute (Pearson) correlations among all pairs of nodes, in order to predict missing edges. The results are shown in Fig. 4 as well (pale colors). This numerical experiment with pairwise correlations as a classifier following traditional network inference paradigm highlights the significant advantage of considering eigenvectors as collective modes of the network (brighter colors). Details about this process are given in Sec. II D.

We observe that performance tends to decrease as the connectivity of the graph increases, as in the previous case, going from $E_1 = 250$ edges to $E_3 = 350$ edges. This is intuitively clear: The same number of five eigenvectors is provided in both cases but is leveraged for different amounts of information to cover the entire graph structure and compensate for the increased number of missing edges.

We observe initial fluctuations of the AUC as a function of the percentage of deleted edges. This is due to the significant and statistically inevitable fluctuations of the number of true positives (TP) when only a small number of edges are missing from G (see Table II).

We conduct the same numerical experiments on regular random graphs and scale-free graphs of different connectivities. The results are presented in Appendix B.

B. Inferring from collective patterns

Were all graph eigenvectors and eigenvalues empirically available, graph reconstruction would ensue directly according to Eq. (1) (see Sec. II). In practice, this is not the case. Only collective patterns approximating eigenvectors will be given, often only a small number of distinct patterns will be available, and eigenvalues are unknown. These points will be addressed in the following.

We resort to the use of measurable collective patterns focusing on Turing patterns (see Sec. II E for a detailed description). We consider a Gierer-Meinhardt [52] system of reaction-diffusion equations while keeping in mind that the outlined method is model-independent. Using the method from [38] for parameter selection to active a specific region of the spectrum, we excite a narrow range of the graph eigenvectors around the mode $\alpha = 10$ (see Fig. 2). The AUC as a function of the percentage of deleted edges is given in Fig. 5. As in the previous case, the reconstruction process remains robust with regards to edge deletion, although, understandably, our method overall works better with eigenvectors due to their mutual orthogonality, they tend to collectively carry more information about the latent structure than patterns do. This can be largely compensated by the measurements of multiple patterns. Using the same previously considered network configuration and parameters, and after random deletion of 50% of all the network edges, we estimate the AUC using an increasing number of Turing patterns. The results are presented in Fig. 6.

So far, our analysis has focused on predictability in general, by computing the AUC averaged over multiple runs. In practical applications, one would rather be confronted with a single

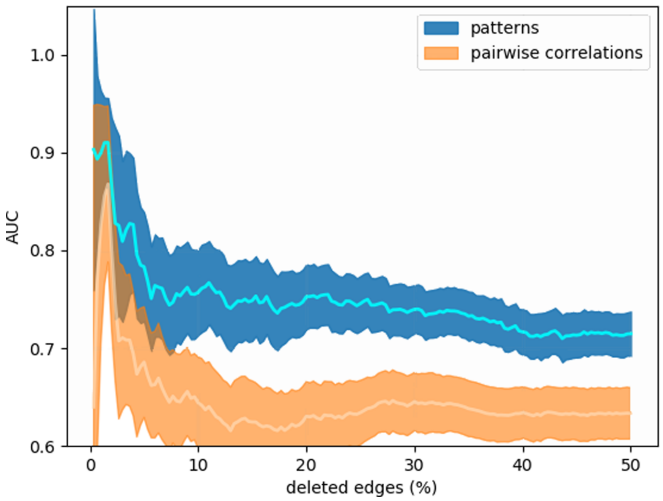

FIG. 5. AUC estimation of the reconstruction of an Erdös-Rényi G of $N = 100$ nodes and $E = 300$ edges, as a function of the percentage of deleted edges. We run a Gierer-Meinhardt model with kinetic parameters obtained via the selection scheme from [38] (here, $a = 0.077$ and $b = 0.759$, and diffusion parameters $\epsilon = 0.025$ and $\sigma = 10.02$) in order to excite a narrow range of the graph eigenvectors around the mode $\alpha = 10$. We obtain five Turing patterns and use them to infer the original graph structure after random deletion of a single edge and up to 50% of all the network edges. This process is repeated for a $n_R = 100$ iterations. The resulting AUCs are depicted as a band (standard deviation) of which average is indicated by a line within it (blue). These values are compared with those drawn from pairwise correlations (orange).

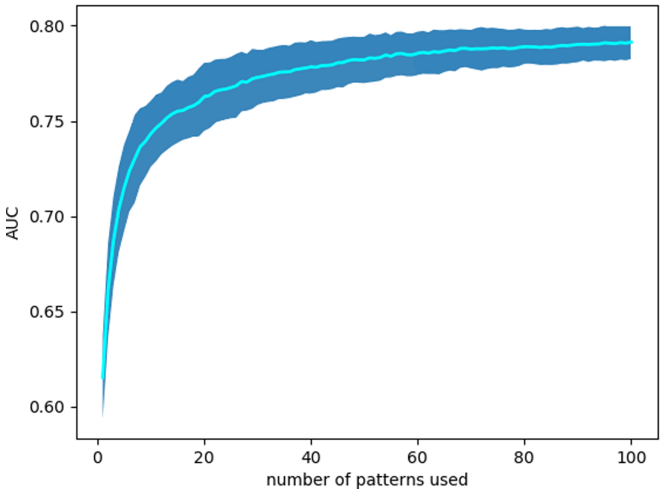

FIG. 6. AUC estimation of the reconstruction of an Erdös-Rényi graph G of $N = 100$ nodes and $E = 250$ edges, as a function of the number of patterns used. Parameters for the Gierer-Meinhardt model are the same in Fig. 5. We obtain $n_p = 1 \ldots 100$ Turing patterns and use them to infer the original graph structure after random deletion of 50% of all the network edges. This process is repeated for a $n_R = 100$ iterations. The resulting AUCs are depicted as a band (standard deviation) of which average is indicated by a lighter color.

incidence of a network, together with a few measured patterns. In Appendix A, we provide two examples of such a practical application, demonstrating that, indeed, our approach leads to a good prediction of missing links, outperforming substantially a simple heuristic based on pairwise correlations.

C. Eigenspace characterization of prediction quality

1. Eigenvector rank importance and network architecture

Not all eigenvectors contribute similarly to the reconstruction process. Across all network architectures, those associated with the largest eigenvalues lead to more accurate predictions, represented in a higher AUC downstream. This is outlined by Fig. 7 where the AUC is estimated from five eigenvectors randomly picked from a moving interval of 15 eigenvectors, of which index ranges from 1 to $N - 14$. This process is iterated for three different network topologies, for $n_n = 100$ networks each.

The trend is consistent in all three cases: There is an increasing linear relation between the eigenvectors' indices and the resulting AUC. In other words, the eigenvectors associated with the largest eigenvalues – those corresponding to the first indices – lead to more accurate predictions. Additionally, we observe increased performances going from a scale-free to an Erdös-Rényi to a regular random graph. We hypothesize that this is due to the differences in eigenvectors localization: Widely distributed eigenvectors generally provide more information about the overall graph structure than localized eigenvectors. This results in the method – statistically – performing better for graphs with more widely distributed eigenvectors, such as regular random graphs [54].

Less intuitive, however, is that the eigenvectors from the other end of the spectrum provide precisely opposite information, suggesting that placing the edges *not* in the slots derived

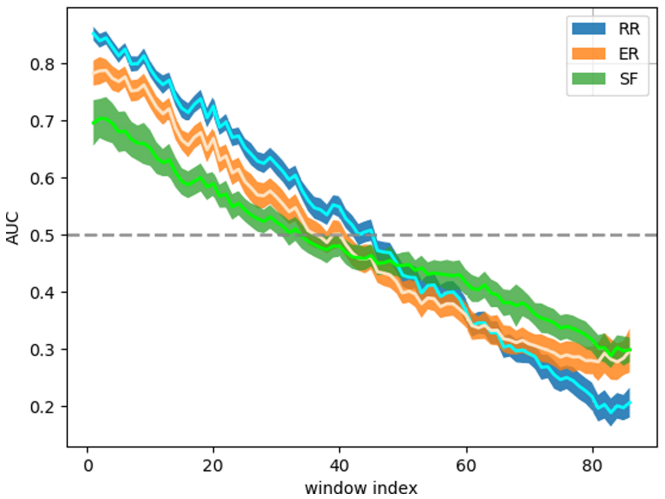

FIG. 7. AUC estimation of the reconstruction of three different network architectures: A regular random graph G_R of $N_R = 100$ nodes, degree $d = 6$, and $E_R = 300$ edges, an Erdös-Rényi graph of $N_E = 100$ nodes and $E_E = 300$ edges, and a scale-free graph of $N_S = 100$ nodes, attachment parameter $m = 3$ and $E_S = 291$ edges. In all three cases, the method is applied after random deletion of 50% of all network edges and taking as input a moving interval of five eigenvectors. The resulting AUCs are depicted as bands (standard deviation) of which average is indicated by a line within each band. The grey line $AUC = 0.5$ indicates random predictions.

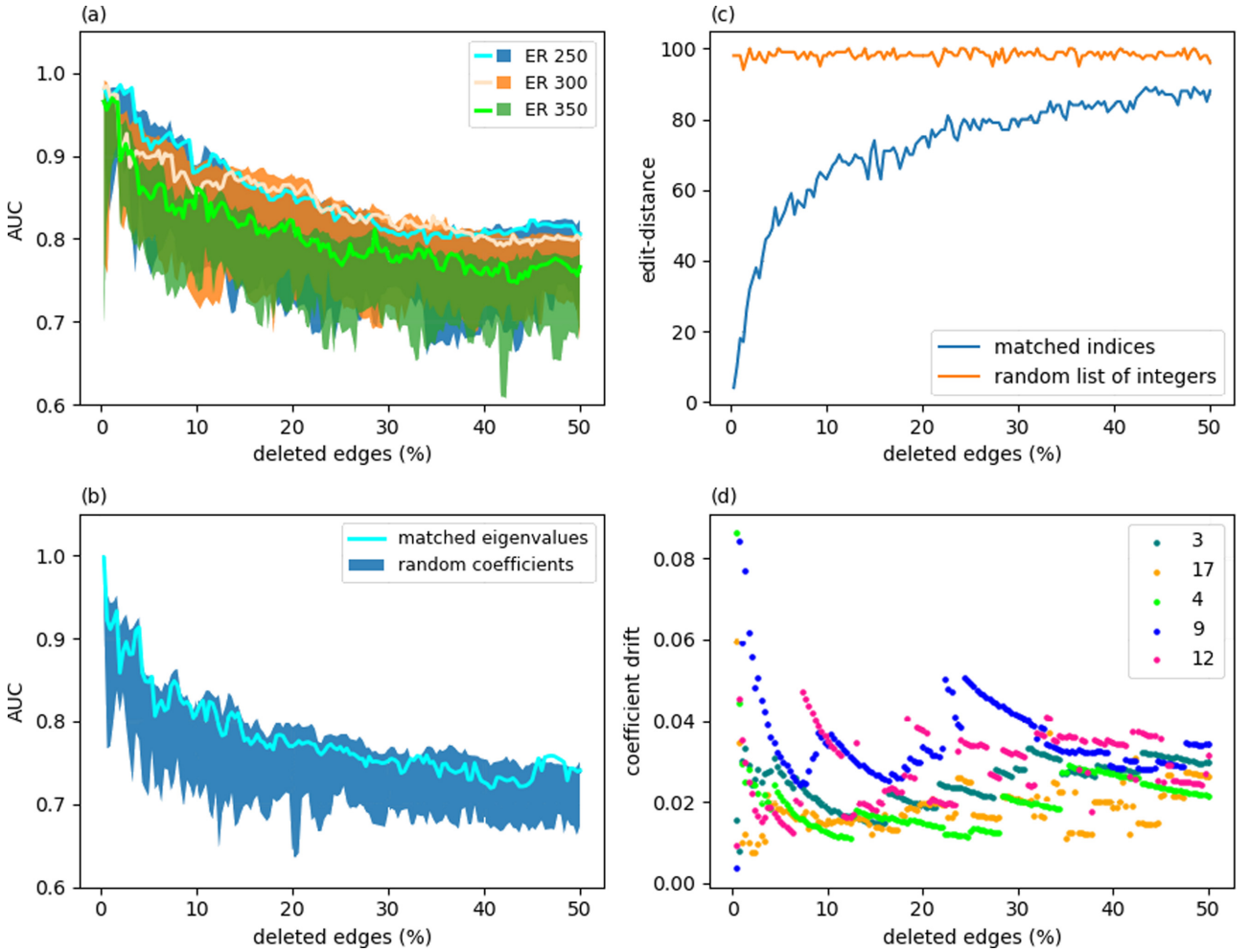

FIG. 8. Coefficient matching panel: (a) AUC estimation of the reconstruction of three Erdös-Rényi graphs G_i of $N = 100$ nodes and degrees $d_i = 4, 5, 6$ as a function of the percentage of deleted edges. In each case, we use a set of five eigenvectors randomly picked from the interval $\vec{\phi}_3 \ldots \vec{\phi}_{17}$. We infer the original graph structure after random deletion of a single edge and up to 50% of all network edges. Each case (blue, orange and green lines) is then compared with a $n_r = 100$ benchmark trials where we do not match eigenvalues but use random numbers between -1 and 0 as coefficients (blue, orange and green bands). (b) AUC estimation of the reconstruction of an Erdös-Rényi graph G of $N = 100$ nodes and $E = 250$ edges, as a function of the percentage of deleted edges. Parameters for the Gierer-Meinhardt model are the same as in Figs. 5 and 6. We obtain five Turing patterns and use them to infer the original graph structure after random deletion of a single edge and up to 50% of all the network edges (line). Benchmark comparison is performed as in (a). (c) Levenshtein edit-distance [55] between the sequence of matched eigenvectors indices $\{\alpha\}$ and the ideal sequence of $\{1...N\}$ as a function of the percentage of deleted edges. We consider the same regular random graph G as in (a) and (b). The Levenshtein edit-distance between two sequences measures the minimal number of single-character edits needed to transform one into another. Thus, its maximum possible value is equal to the length of the considered sequence which is, in this case, $N = 100$. As a benchmark test assessing the precision of index matching, the Levenshtein edit-distance relative to the matched indices (blue) is compared, at each step of the deletion process, with the same distance relative to a random sequence of integers between 1 and $N = 100$ (orange). (d) Coefficient drift as a function of the percentage of deleted edges for the same Erdös-Rényi graph G as in (a) and (b).

from these patterns is a better strategy for missing-edge detection (see red and purple curves in Fig. 3). We believe that this is due to the residual (but opposite) information about large-scale patterns contained in the localized patterns coming from the mutual orthogonality of eigenvalues.

2. *Matching coefficients in the spectral decomposition*

Inferring the original graph Laplacian involves associating the measured eigenvectors $\vec{\phi}_\alpha$ (or patterns) to the right coefficients, the exact values of which would be the Laplacian eigenvalues Λ_α. Having access, in practice, only to the incomplete graph Laplacian eigenvalues $\Lambda_\alpha^{(D)}$, we match the latter with the former according to a process described in Sec. II C.

To assess the added performance of this matching process, we compare, at each step of the deletion process, the AUC derived using matched coefficients to the one using N random real numbers uniformly distributed between -1 and 0. Repeating this process $n_r = 100$ times, we obtain Fig. 8(a). Conducting the same experiment using Turing patterns instead of eigenvectors, we get Fig. 8(b).

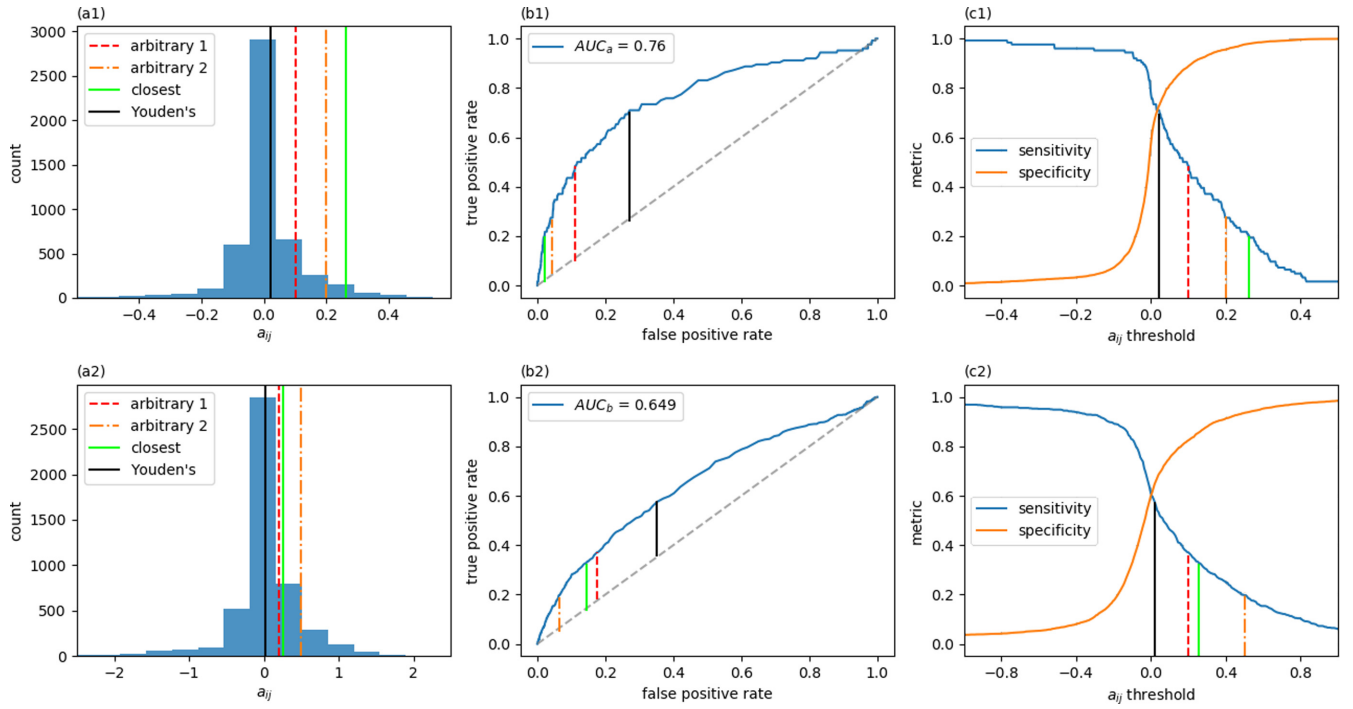

FIG. 9. Performance panel of graph reconstruction of: (1) an Erdös-Rényi graph G_1 of $N_1 = 100$ nodes, $E_1 = 248$ edges and density $d_1 = 0.05$ on which we run a Gierer-Meinhardt model with kinetic parameters $a_1 = 0.06$ and $b_1 = 0.72$, and diffusion parameters $\epsilon_1 = 0.029$ and $\sigma_1 = 10.02$, in order to excite a narrow range of the graph eigenvectors around the mode $\alpha_1 = 10$. We obtain five Turing patterns and use them to infer the original graph structure after random deletion of 50% of all the network edges. (a1) Histogram of the inferred adjacency matrix elements (a_{ij}) for $i, j = 1 \ldots N_1$. (b1) ROC curves: true positive rates as a function of false positive rates. (c1) Sensitivity and specificity as a function of the applied (a_{ij}) threshold. (2) an Erdös-Rényi graph G_2 of $N_2 = 100$ nodes, $E_2 = 1485$ edges and density $d_2 = 0.3$ on which we run a Gierer-Meinhardt model with kinetic parameters $a_2 = 0.09$ and $b_2 = 0.79$, and diffusion parameters $\epsilon_2 = 0.007$ and $\sigma_2 = 10.02$, in order to excite a narrow range of the graph eigenvectors around the mode $\alpha_2 = 10$. We obtain five Turing patterns and use them to infer the original graph structure after random deletion of 50% of all the network edges. (a2) Histogram of the inferred adjacency matrix elements (a_{ij}) for $i, j = 1 \ldots N_2$. (b2) ROC curves: true positive rates as a function of false positive rates. (c2) Sensitivity and specificity as a function of the applied (a_{ij}) threshold. The prediction of missing links requires the selection of discrimination thresholds for the inferred adjacency matrix elements (a_{ij}). In the two present examples, we select four thresholds, depicted as vertical lines: *arbitrary 1* and *arbitrary 2* are chosen arbitrarily, *closest* is the threshold approximating the original graph connectivity, assuming this is known empirically, and *Youden's* is the threshold maximizing the Youden's index, that is, the sum of sensitivity and specificity [75].

In both cases, we observe that, at any step of the deletion process, $AUC_m \gtrsim AUC_{r,i}$ for $i = 1 \ldots n_r$, i.e., the reconstruction using matched eigenvalues performs almost consistently as good as the best recorded reconstruction derived over the accumulated $n_r = 100$ trials using random coefficients. As eigenvectors are randomly selected from a comparatively small interval, the corresponding eigenvalues are not so different from each other. This explains why the matching of coefficients enhances the AUC only by about 3.2% for eigenvectors [see Fig. 8(a)] and 2.1% for patterns [see Fig. 8(b)] on average, in the present case. We expect this increase to be more significant if eigenvectors (or patterns) are selected from a broader range.

3. *Resilience to topological damage*

As observed in Fig. 5, edge deletion causes increasingly poor matching between the original eigenvectors $\vec{\phi}$ and the eigenvectors of the incomplete graph $\vec{\phi}^{(D)}$ [see Fig. 8(c)]. Yet, its adverse effects on the AUC – beyond a certain limit – remains minimal. The limited, though visible, importance of coefficient matching is due to the selection of patterns from a comparatively small region of the spectrum. As the spectrum is locally rather flat, the impact of coefficient matching is reduced.

Next, we define the 'coefficient drift' as $\Delta_\alpha(e) = |\Lambda^{(D)}_{\beta^*}(e) - \Lambda_\alpha|/e$ for e deleted edges, that is, the difference between the matched coefficients $\Lambda^{(D)}_{\beta^*}$ and the ideal original eigenvalues Λ_α normalized by the number of deleted edges e. Its variation, as a function of the number of deleted edges, is illustrated in Fig. 8(d). We argue that, although the matched eigenvalues $\Lambda^{(D)}_{\beta^*}$ ranking can vary highly under edge deletion, the amplitude of the coefficient mismatching remains constrained in the same narrow range even after the deletion of a large proportion of the graph edges, an observation that can be partly explained by the interlacing theorem [56]. Consequently, increased mismatchings only have a limited negative impact on the final prediction.

TABLE III. Inference results for G_1 using 'collective patterns' ($AUC_1 = 0.76$) vs 'pairwise correlations' ($AUC_{1*} = 0.625$). Bold text indicates better predictions.

	Arbitrary 1		Arbitrary 2		Closest		Youden's	
	Patterns	Pairwise	Patterns	Pairwise	Patterns	Pairwise	Patterns	Pairwise
threshold	0.1		0.2		0.263	1.871	0.022	0.371
number of edges	**707**	2325	**366**	2179	247	**248**	**1489**	1937
sensitivity	0.484	**0.629**	0.274	**0.589**	**0.202**	0.081	**0.728**	0.629
specificity	**0.889**	0.543	**0.956**	0.575	**0.979**	0.977	**0.728**	0.629
balanced accuracy	**0.686**	0.543	**0.615**	0.582	**0.59**	0.529	**0.719**	0.601
precision	**0.103**	0.035	**0.14**	0.035	**0.203**	0.083	**0.064**	0.039
F1-score	**0.17**	0.066	**0.186**	0.067	**0.202**	0.082	**0.118**	0.073

IV. DISCUSSION

Network inference methods have been developed to estimate missing edges, or even whole networks, from available data. The bulk of network inference methods evaluates pairwise relationships between nodes. Here we formulate a new paradigm of network inference evaluating data as self-organized collective patterns. We illustrate this approach for Turing patterns in reaction-diffusion dynamics on graphs, a case where patterns are related to eigenvectors of the graph's Laplacian matrix.

In this instance, the task of inferring missing edges is a truncated and inexact version of spectral decomposition where accuracy is constrained by the segment of eigenvectors available, as well as the rank stability of the eigenvalues as a function of the percentage of missing edges.

We first show that this approach works in a stylized form, where patterns are assumed to be eigenvectors. Then we illustrate how it can be operationalized also for (simulated) Turing patterns. We observe only a weak dependence of inference quality on global network architecture. However, the eigenvalue range (indirectly representing the spatial scale of the patterns) available has a dramatic effect on the final output (performance and/or quality).

Several limitations of our method need to be mentioned. Our approach is restricted to undirected graphs, as we require real eigenvectors and eigenvalues of the Laplacian matrix. We acknowledge that, in many applications, graphs will be directed (see, e.g., [57–60]). We anticipate, however, that our approach might work for approximately symmetric interaction matrices, like connectomes in neuroscience [61,62].

It would be interesting to further elaborate on possible generalizations of the approach to account for different network settings such as non-normal [63], directed [64], modular [65], and multilayer networks [66,67], as often encountered in real-world reaction-diffusion systems [68–72].

Additionally, our present framework stands on the validity of the linear stability analysis. This limits the accuracy of the inverse problem is two ways: (i) It neglects the contribution of stable modes [see Fig. 8(b)]. (ii) Equating patterns and eigenvectors is only possible close to the Turing instability. Nonlinear combinations of eigenvectors become relevant with increasing distance from the bifurcation point.

Moreover, specific topological features can hinder the quality of our predictions. As depicted in Fig. 7, eigenvectors – and by extension, patterns – need to be 'global' enough to contain network-wide information. Localized eigenvectors, such as those observed in modular networks, are poorly informative. Nonetheless, network architectural features might also facilitate such inference, e.g., in cases where the missing edges are not uniformly distributed but degree-dependent or module-based. Note that for a uniform random distribution of missing edges (the case discussed in the present work) we have not found a strong network dependence (see Appendix B).

While we are convinced that any type of self-organized patterns could be operationalized in this way, we here focus on patterns approximating eigenvectors of the Laplacian matrix of the graph. It is clear, of course, that eigenvectors and spatial embeddings of networks (via 'graph layout algorithms' like spring embedding) are not independent (see, e.g., [42] or a comparison of spring embedding and self-organized excitation waves on graphs). In this sense, on a formal level,

TABLE IV. Inference results for G_2 using 'collective patterns' ($AUC_2 = 0.649$) vs 'pairwise correlations' ($AUC_{2*} = 0.519$). Bold text indicates better predictions.

	Arbitrary 1		Arbitrary 2		Closest		Youden's	
	Patterns	Pairwise	Patterns	Pairwise	Patterns	Pairwise	Patterns	Pairwise
threshold	0.2		0.5		0.257	1.155	0.02	−0.536
number of edges	**1626**	2550	**1112**	2218	**1484**	1483	**2391**	3356
sensitivity	0.371	**0.443**	0.197	**0.348**	**0.327**	0.177	**0.575**	0.709
specificity	**0.825**	0.576	**0.936**	0.65	**0.856**	0.826	**0.648**	0.354
balanced accuracy	**0.598**	0.51	**0.566**	0.499	**0.592**	0.501	**0.612**	0.531
precision	**0.311**	0.183	**0.396**	0.175	**0.328**	0.178	**0.259**	0.19
F1-score	**0.338**	0.259	**0.263**	0.233	**0.328**	0.178	**0.357**	0.3

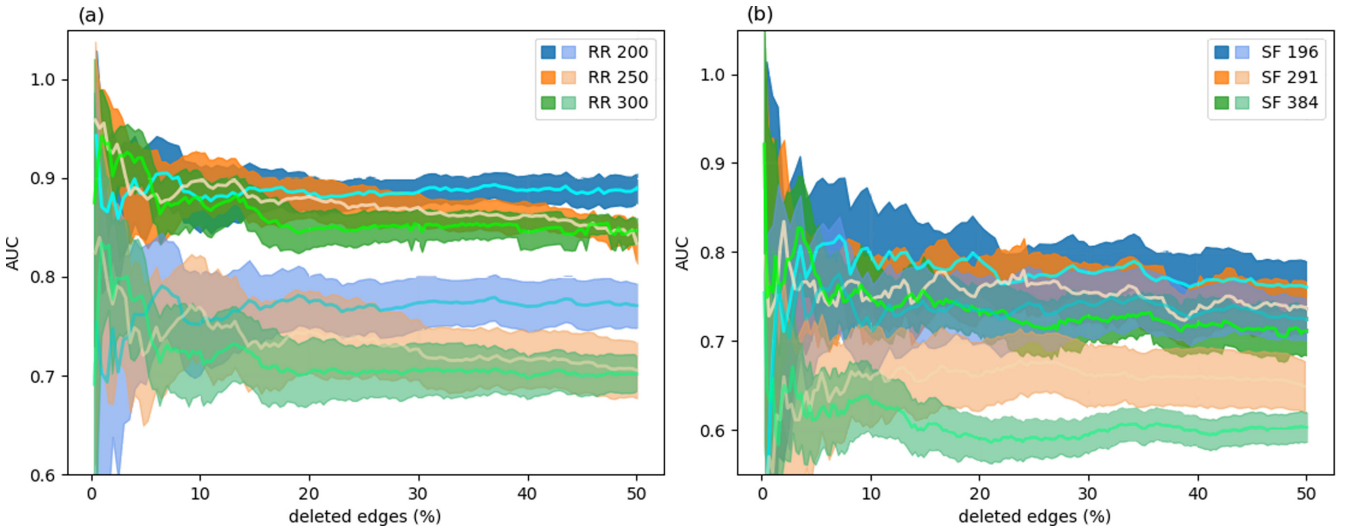

FIG. 10. AUC estimation of the reconstruction of: (a) Three regular random graphs G_i of $N = 100$ nodes, degree $d_i = 4, 5, 6$, and $E_i = 200, 250, 300$ edges, as a function of the percentage of deleted edges. In each case and for a $n_R = 100$ iterations each, we use a set of five eigenvectors randomly picked from the interval $\vec{\phi}_3 \ldots \vec{\phi}_{17}$. We infer the original graph structures after random deletion of a single edge and up to 50% of all network edges. For each graph, the resulting AUCs are depicted as bands (standard deviation) of which average is indicated by a line within each band. These values are compared with those drawn from pairwise correlations, depicted in corresponding pale colors. (b) Three scale-free graphs G_i of $N = 100$ nodes, attachment parameter $m_i = 2, 3, 4$, and $E_i = 196, 291, 384$ edges, as a function of the percentage of deleted edges. In each case and for a $n_R = 100$ iterations each, we use a set of five eigenvectors randomly picked from the interval $\vec{\phi}_3 \ldots \vec{\phi}_{17}$. We infer the original graph structures after random deletion of a single edge and up to 50% of all network edges. For each graph, the resulting AUCs are depicted as bands (standard deviation) of which average is indicated by a line within each band. These values are compared with those drawn from pairwise correlations, depicted in corresponding pale colors.

we expect that our approach can help understand and refine embedding-based edge prediction algorithms (see, e.g., [20]).

An important component of our pattern-based network completion algorithm is the robustness of eigenvalue under removal of edges, as these eigenvalues derived from the incomplete network serve as coefficients in our truncated spectral decomposition. This robustness (or, conversely, fluctuation) of the rank under decreasing network information [see Fig. 8(c)] is also of relevance to the fascinating topic of ranking nodes in incomplete or evolving complex networks, which has been addressed by [73,74].

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 859937.

APPENDIX A

In this Appendix we present two worked-out examples illustrating our method. Figure 9 summarizes our prediction results. Tables III and IV provide the numerical details.

Graph sparsity entails a number of negative tests largely outweighing the number of positive tests: Most pairs of nodes, in the original graph, are not connected by an edge, thus, not predicting any link leads to a significant number of TN and therefore, a misleading large specificity, as pointed out in [76]. To better assess our results for the different chosen thresholds, beyond the previously introduced (i) sensitivity $TPR = TP/(TP + FN)$ and (ii) specificity $TNR = TN/(TN + FP)$, we measure:

(iii) The balanced accuracy $BA = (TPR + TNR)/2$, the geometric average between sensitivity and specificity.

(iv) The precision, also called the Positive Predictive Value $PPV = TP/(TP + FP)$, which does not resort to TNs and outputs the proportion of TPs out of the all positive calls.

(v) The F1-score $F1 = 2PPV * TPR/(PPV + TPR)$, the harmonic average between precision and sensitivity.

APPENDIX B

This Appendix summarizes the statistics of our method for different network architectures (beyond the ER graphs studied in the main text): regular random (RR) graphs and scale-free (SF) graphs (i.e., BA graphs, [77], generated via preferential attachment). The results are given in Fig. 10. These support our previous findings regarding the influence of network architecture via eigenvectors localization, see Sec. III C).

[1] A.-L. Barabási, Nat. Phys. **8**, 14 (2012).
[2] J.-F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou *et al.*, Nature (London) **437**, 1173 (2005).
[3] D. Szklarczyk, A. L. Gable, D. Lyon, A. Junge, S. Wyder, J. Huerta-Cepas, M. Simonovic, N. T. Doncheva, J. H. Morris, P. Bork *et al.*, Nucleic Acids Res. **47**, D607 (2019).
[4] R. L. Cross, R. L. Cross, and A. Parker, *The Hidden Power of Social Networks: Understanding How Work*

Really Gets Done in Organizations (Harvard Business Press, Boston, Massachusetts, 2004)
[5] C. Prell, *Social Network Analysis: History, Theory and Methodology* (Sage, London, 2012).
[6] A. Santos-Zavaleta, H. Salgado, S. Gama-Castro, M. Sánchez-Pérez, L. Gómez-Romero, D. Ledezma-Tejeida, J. S. García-Sotelo, K. Alquicira-Hernández, L. J. Muñiz-Rascado, P. Peña-Loredo *et al.*, Nucleic Acids Res. **47**, D212 (2019).
[7] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C. J. Honey, V. J. Wedeen, and O. Sporns, PLoS Biol **6**, e159 (2008).
[8] O. Sporns, Ann. N.Y. Acad. Sci. **1224**, 109 (2011).
[9] P. Killworth and H. Bernard, Human Organization **35**, 269 (1976).
[10] A. Clauset, C. Moore, and M. E. Newman, Nature (London) **453**, 98 (2008).
[11] R. Guimerà and M. Sales-Pardo, Proc. Natl. Acad. Sci. **106**, 22073 (2009).
[12] A. S. Schwartz, J. Yu, K. R. Gardenour, R. L. Finley, Jr., and T. Ideker, Nat. Methods **6**, 55 (2009).
[13] T. Martin, B. Ball, and M. E. J. Newman, Phys. Rev. E **93**, 012306 (2016).
[14] M. E. Newman, Nat. Phys. **14**, 542 (2018).
[15] A. Ghasemian, H. Hosseinmardi, A. Galstyan, E. M. Airoldi, and A. Clauset, Proc. Natl. Acad. Sci. **117**, 23393 (2020).
[16] Y.-X. Zhu, L. Lü, Q.-M. Zhang, and T. Zhou, Physica A **391**, 5769 (2012).
[17] S. Bai, L. Li, J. Cheng, S. Xu, and X. Chen, Complexity **2018**, 7312603 (2018).
[18] X. Zhu, H. Tian, and S. Cai, Physica A **413**, 515 (2014).
[19] J. Ding, L. Jiao, J. Wu, Y. Hou, and Y. Qi, Physica A **417**, 76 (2015).
[20] H. Cai, V. W. Zheng, and K. C.-C. Chang, IEEE Trans. Knowl. Data Eng. **30**, 1616 (2018).
[21] I. Brugere, B. Gallagher, and T. Y. Berger-Wolf, ACM Comput. Surv. **51**, 1 (2018).
[22] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky, Proc. Natl. Acad. Sci. **107**, 6286 (2010).
[23] D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, A. Aderhold, R. Bonneau, Y. Chen *et al.*, Nat. Methods **9**, 796 (2012).
[24] A. C. Babtie, T. E. Chan, and M. P. H. Stumpf, Curr. Opin. Syst. Biol. **5**, 72 (2017).
[25] X. Dong, A. Yambartsev, S. A. Ramsey, L. D. Thomas, N. Shulzhenko, and A. Morgun, arXiv:1411.0595.
[26] M.-T. Hütt and A. Lesne, in *Systems Medicine: Integrative, Qualitative and Computational Approaches*, edited by O. Wolkenhauer (Elsevier, Cambridge, Massachusetts, 2020), pp. 77–85.
[27] A. Messé, M.-T. Hütt, P. König, and C. C. Hilgetag, Sci. Rep. **5**, 7870 (2015).
[28] V. Voutsa, D. Battaglia, L. J. Bracken, A. Brovelli, J. Costescu, M. Diaz Munoz, B. D. Fath, A. Funk, M. Guirro, T. Hein, C. Kerschner, C. Kimmich, V. Lima, A. Messé, A. J. Parsons, J. Perez, R. Pöppl, C. Prell, S. Recinos, Y. Shi *et al.*, J. R. Soc. Interface. **18**, 20210486 (2021).
[29] C. Martin and P. Niemeyer, Network Science **7**, 180 (2019).
[30] T. Hoffmann, L. Peel, R. Lambiotte, and N. S. Jones, Sci. Adv. **6**, eaav1478 (2020).
[31] J. Dai, K. Huang, Y. Liu, C. Yang, and Z. Wang, IEEE Syst. J. **15**, 1959 (2020).
[32] M. Asllani, T. Carletti, F. Di Patti, D. Fanelli, and F. Piazza, Phys. Rev. Lett. **120**, 158301 (2018).
[33] I. Adam, D. Fanelli, T. Carletti, and G. Innocenti, Eur. Phys. J. B **92**, 99 (2019).
[34] R. Burioni, M. Casartelli, M. di Volo, R. Livi, and A. Vezzani, Sci. Rep. **4**, 1 (2014).
[35] I. Adam, G. Cecchini, D. Fanelli, T. Kreuz, R. Livi, M. di Volo, A. L. A. Mascaro, E. Conti, A. Scaglione, L. Silvestri *et al.*, Chaos Solitons Fractals **140**, 110235 (2020).
[36] P. Nyczka, M.-T. Hütt, and A. Lesne, Physica A **566**, 125631 (2021).
[37] H. Nakao and A. S. Mikhailov, Nat. Phys. **6**, 544 (2010).
[38] M.-T. Hütt, D. Armbruster, and A. Lesne, Phys. Rev. E **105**, 014304 (2022).
[39] X. Diego, L. Marcon, P. Müller, and J. Sharpe, Phys. Rev. X **8**, 021071 (2018).
[40] M. Müller-Linow, C. C. Hilgetag, and M.-T. Hütt, PLoS Comput. Biol. **4**, e1000190 (2008).
[41] S. Jahnke, R.-M. Memmesheimer, and M. Timme, Phys. Rev. E **89**, 030701(R) (2014).
[42] P. Moretti and M.-T. Hütt, Proc. Natl. Acad. Sci. USA **117**, 18332 (2020).
[43] M. Nitzan, J. Casadiego, and M. Timme, Sci. Adv. **3**, e1600396 (2017).
[44] T. Gross, M. J. Wongchenko, Y. Yan, and N. Blüthgen, Bioinformatics **35**, i634 (2019).
[45] M. Timme, Phys. Rev. Lett. **98**, 224101 (2007).
[46] S. G. Shandilya and M. Timme, New J. Phys. **13**, 013004 (2011).
[47] F. Harary, *Graphs and Combinatorics* (Springer, Tokyo, 1974), pp. 18–28.
[48] C. D. Godsil and B. D. McKay, J. Comb. Theory, Ser. B **30**, 285 (1981).
[49] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Noise Reduction in Speech Processing* (Springer, Berlin, 2009), pp. 1–4.
[50] A. M. Turing, Bull. Math. Biol. **52**, 153 (1990).
[51] M. Mimura and J. Murray, J. Theor. Biol. **75**, 249 (1978).
[52] A. Gierer and H. Meinhardt, Kybernetik **12**, 30 (1972).
[53] T. Fawcett, Pattern Recognition Letters **27**, 861 (2006).
[54] S. Hata and H. Nakao, Sci. Rep. **7**, 1 (2017).
[55] V. I. Levenshtein *et al.*, Soviet Physics Doklady **10**, 707 (1966).
[56] W. H. Haemers, Linear Algebra Appl. **226–228**, 593 (1995).
[57] E. A. Leicht and M. E. J. Newman, Phys. Rev. Lett. **100**, 118703 (2008).
[58] F. D. Malliaros and M. Vazirgiannis, Phys. Rep. **533**, 95 (2013).
[59] M. Asllani, R. Lambiotte, and T. Carletti, Sci. Adv. **4**, eaau9403 (2018).
[60] R. S. MacKay, S. Johnson, and B. Sansom, R. Soc. Open Sci. **7**, 201138 (2020).
[61] O. Sporns, G. Tononi, and R. Kötter, PLoS Comput. Biol. **1**, e42 (2005).
[62] N. T. Markov, M. M. Ercsey-Ravasz, A. Ribeiro Gomes, C. Lamy, L. Magrou, J. Vezoli, P. Misery, A. Falchier, R. Quilodran, M.-A. Gariel *et al.*, Cereb. Cortex **24**, 17 (2014).
[63] R. Muolo, M. Asllani, D. Fanelli, P. K. Maini, and T. Carletti, J. Theor. Biol. **480**, 81 (2019).

[64] M. Asllani, J. D. Challenger, F. S. Pavone, L. Sacconi, and D. Fanelli, Nat. Commun. **5**, 4517 (2014).
[65] B. A. Siebert, C. L. Hall, J. P. Gleeson, and M. Asllani, Phys. Rev. E **102**, 052306 (2020).
[66] M. Asllani, D. M. Busiello, T. Carletti, D. Fanelli, and G. Planchon, Phys. Rev. E **90**, 042814 (2014).
[67] N. E. Kouvaris, S. Hata, and A. D. Guilera, Sci. Rep. **5**, 10840 (2015).
[68] M. Rietkerk and J. Van de Koppel, Trends Ecol. Evol. **23**, 169 (2008).
[69] Q.-X. Liu, A. Doelman, V. Rottschäfer, M. de Jager, P. M. Herman, M. Rietkerk, and J. van de Koppel, Proc. Natl. Acad. Sci. **110**, 11905 (2013).
[70] C. Granell, S. Gómez, and A. Arenas, Phys. Rev. Lett. **111**, 128701 (2013).
[71] J. Klaise and S. Johnson, Chaos: An Interdiscip. J. Nonlinear Sci. **26**, 065310 (2016).
[72] R. E. Ulanowicz, R. D. Holt, and M. Barfield, Ecol. Lett. **17**, 127 (2014).
[73] H. Liao, M. S. Mariani, M. Medo, Y.-C. Zhang, and M.-Y. Zhou, Phys. Rep. **689**, 1 (2017).
[74] L. V. Bozhilova, A. V. Whitmore, J. Wray, G. Reinert, and C. M. Deane, BMC Bioinf. **20**, 446 (2019).
[75] W. J. Youden, Cancer **3**, 32 (1950).
[76] T. Saito and M. Rehmsmeier, PloS One **10**, e0118432 (2015).
[77] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).