Perturbation of the normalized Laplacian matrix for the prediction of missing links in real networks

Roya Aliakbarisani, Abdorasoul Ghasemi, and M. Ángeles Serrano

Abstract—The problem of predicting missing links in real-world networks is an active and challenging research area in both science and engineering. The goal is to model the process of link formation in a complex network based on its observed structure to unveil lost or unseen interactions. In this paper, we use perturbation theory to develop a general link prediction procedure, called Laplacian Perturbation Method (LPM), that relies on relevant structural information encoded in the normalized Laplacian matrix of the network. We implement a general algorithm for our perturbation method valid for different Laplacian-based link prediction schemes that successfully surpass the prediction accuracy of their standard non-perturbed versions in real-world and model networks. The suggested LPM for link prediction also exhibits higher accuracy compared to other extensively used local and global state-of-the-art techniques and, in particular, it outperform the Structural Perturbation Method (SPM), a popular procedure that perturbs the adjacency matrix of a network for inferring missing links, in many real-world and in synthetic networks. Taken together, our results show that perturbation methods can significantly improve Laplacian-based link prediction techniques, and feeds the debate on which representation, Laplacian or adjacency, better represents structural information for link prediction tasks in networks.

Index Terms—complex networks, link prediction, perturbation theory, normalized Laplacian matrix

1 Introduction

Link prediction methods in complex networks [1] aim at inferring missing or future links based on the observed structure and node attributes. The benefit of improving network reconstructions not only serves descriptive purposes but can also have profound effects in understanding the behavior of processes that run on networks, such as information cascades [2], [3]. As a consequence, link prediction techniques have been profusely used in different disciplines for discovering unknown protein-protein interactions in biological networks, suggesting new friends in social networks, proposing products in recommender systems, developing transportation and telecommunication networks, and many more. However, it is a challenging problem due to different issues. For instance, the stochastic nature of link formation processes imposes intrinsic upper bounds to link prediction accuracy, generally far from the absolute maximum, in real-world networks [4].

Most link prediction algorithms use the adjacency matrix as a representation of the observed network structure. Among them, local similarity-based link prediction methods [5] —such as the common neighbors index (CN) [6], the Adamic Adar index (AA) [7], the resource allocation index (RA) [8], and the Cannistraci-Hebb index (CH) [9]— extract required structural information about common neighbors and node degrees from the adjacency matrix. In parallel, global similarity-based link prediction methods that incorporate topological information about the whole network for estimating the similarity between unconnected node pairs [5] —such as the Structural Perturbation Method (SPM) [10] and the Katz index [11]— and commmunity-based methods for missing link prediction, such as the fast probability block model (FBM) [12], also depend on the adjacency matrix.

The Laplacian matrix gives an alternative representation of graph structure, and due to its specific features, several link prediction methods use it, or functions of it, as the source of similarity indices between node pairs [13], [14], [15]. Even if the Laplacian carries the same information as the adjacency matrix, the graph Laplacian has different properties and may fit better specific problems. For instance, the determinant of the Laplacian matrix specifies the number of spanning trees in a network [16] and its eigenvalues reveal how well complex networks are connected [17] and how fast they can spread information through their nodes [18]. From a geometrical perspective, the Laplacian matrix relates a network to its geometrical representation in terms of a simplex [19].

In this paper, our main contribution is to introduce a perturbative methodology that is able to improve Laplacian-based state-of-the-art link prediction techniques. To this end, we present the Laplacian perturbation method (LPM) that uses perturbation theory on the graph normalized Laplacian to predict missing links, inspired by previous work that used perturbation on the adjacency matrix of a graph [10]. The specific form of the selected normalization

- *R. Aliakbarisani is with the Department of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran, E-mail: r.aliakbarisani@ee.kntu.ac.ir.*
- *A. Ghasemi is with the Department of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran, E-mail: arghasemi@kntu.ac.ir.*
- *M. Á. Serrano is with the Departament de Física de la Matèria Condensada, Universitat de Barcelona, Martí i Franquès 1, 08028 Barcelona, Spain and Universitat de Barcelona Institute of Complex Systems (UBICS), Universitat de Barcelona, Barcelona, Spain and ICREA, Pg. Lluís Companys 23, E-08010 Barcelona, Spain. Email: marian.serrano@ub.edu.*

Manuscript received January xx, 2021; revised July xx, 2021.

is not trivial but turns out to be crucial to achieve good performance. We show how to obtain a perturbed version of the graph normalized Laplacian and introduce a general algorithm, that we name LPM algorithm for link prediction, that employs it as the source of structural information for inferring new links. Based on this algorithm, we propose some specific LPM link prediction techniques as alternatives to standard Laplacian-based methods. Experimental results over model and real-world networks indicate that these methods not only surpass their unperturbed counterparts in terms of precision and accuracy but also have better accuracy compared to other widely applied link prediction schemes in model and real-world networks. Finally, we observe positive correlations between the performance of LPM link prediction techniques and structural properties of model networks including degree heterogeneity and level of clustering, suggesting that LPM is more advantageous for identifying missing links in more complex structures.

The rest of this paper is organized as follows. Section 2 reviews previous work related to our research. Section 3 describes how to apply perturbation theory to the normalized Laplacian of a graph and presents a general algorithm for link prediction methods based on the perturbed Laplacian. Section 4 introduces specific LPM link prediction methods deploying the perturbed normalized Laplacian matrix to formulate their similarity indices. In Section 5, the prediction accuracy of the proposed LPM link prediction methods is measured over model and real-world networks, and the relations between LPM performance and specific structural network features are investigated for a family of network models. Finally, the paper is concluded in Section 6.

2 Related work

Link prediction is a challenging task and a diversity of strategies have been developed using methods from several scientific fields ranging from network science to machine learning. In network science, link prediction models are typically based on different graph proximity measures or on generative random graph models [20], [21]. In machine learning, schemes based on training graph neural networks to classify potential links have been successfully developed and some of them exhibit superior performance as compared to traditional heuristic-based methods [22], [23]. Most of these methods use the adjacency matrix to encode the graph. In this paper, we focus instead on Laplacian-based link prediction methods, some of which are described next.

2.1 Laplacian-based Link Prediction Methods

Consider an undirected unweighted network $G = (V, E)$, where V is the set of nodes and E is the set of links. The Laplacian matrix corresponding to this network is an $N \times N$ matrix $\mathbf{L}$, where $N = |V|$, defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{A}$ is the adjacency matrix of G with entries 1 or 0 depending on whether the corresponding pairs of nodes are connected, and $\mathbf{D}$ is a diagonal matrix of its node degrees [24]. Since $\mathbf{L}$ is not full rank, its rank is $N - 1$, we use the Moore-Penrose inverse or pseudo-inverse approach to compute its inversion denoted by $\mathbf{L}^\dagger$ [25]. As the graph Laplacian, its pseudo-inverse carries interesting information about network structure and processes. For example, the effective resistance between node pairs can be computed via $\mathbf{L}^\dagger$ by considering networks as electrical circuits [26], [27]. Also, one can use $\mathbf{L}^\dagger$ to compute the average number of steps that a random walker traverses from a source node to reach a destination and then go back to the source [13]. Finally, $\mathbf{L}^\dagger$ can be utilized to rank the nodes in a graph, e.g., in topological centrality [28], or to compare different networks in terms of structural robustness [29].

Furthermore, $\mathbf{L}^\dagger$ is a symmetric positive semidefinite matrix, i.e, all its eigenvalues are non-negative, and as such it is a graph kernel or Gram matrix $\mathscr{G}$ for a set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ [13]. That is, the elements of the Gram matrix are the inner products of their corresponding node vectors, $\mathscr{G}_{ij} = \mathbf{x}_i^T \mathbf{x}_j$ [30]. Therefore, $\mathbf{L}^\dagger$ is decomposable as $\mathbf{L}^\dagger = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ where $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues of $\mathbf{L}^\dagger$ sorted in decreasing order and $\mathbf{U}$ is a column matrix of the corresponding eigenvectors. This suggests that the column i of $\mathbf{\Lambda}^{1/2}\mathbf{U}^T$ is the corresponding node vector $\mathbf{x}_i$ for node i. Consequently, one can interpret the elements of $\mathbf{L}^\dagger$ as the similarity indices between the corresponding node vectors in terms of inner product and use them for link prediction [13]. The similarity function for this link prediction method is formulated as

$$S^{Pinv}(i,j) = L^\dagger_{ij}, \tag{1}$$

where $L^\dagger_{ij}$ is element (i,j) of the graph Laplacian pseudo-inverse.

Since $\mathbf{L}^\dagger$ contains the inner products of node vectors, it is used by another link prediction method to evaluate the similarity between node pairs via the cosine of the angles between their vectors as [13]

$$S^{Cos}(i,j) = \frac{L^\dagger_{ij}}{\sqrt{L^\dagger_{ii} L^\dagger_{jj}}}. \tag{2}$$

$\mathbf{L}^\dagger$ also encodes the average path length between nodes in a network. Let $n(i,j)$ denote the average commute time (ACT) between i and j, i.e., the average number of links that a random walker located at i takes to reach j for the first time and then go back to i. This quantity can be computed by the pseudo-inverse of the graph Laplacian using [13]

$$n(i,j) = |E| \left(L^\dagger_{ii} + L^\dagger_{jj} - 2L^\dagger_{ij} \right), \tag{3}$$

where $|E|$ is the number of links in the network. Assuming that the smaller the average commute time between two nodes the higher their similarity, an ACT similarity index [13] can be defined as

$$S^{ACT}(i,j) = \frac{1}{|E| \left(L^\dagger_{ii} + L^\dagger_{jj} - 2L^\dagger_{ij} \right)}. \tag{4}$$

Matrix-forest-based algorithm (MFA) [14], [15] is another Laplacian-based link prediction method measuring the similarity between two nodes in terms of relative forest accessibility from one node to another, which is related to the Laplacian matrix as

$$S^{MFA}(i,j) = \left[(\mathbf{I} + \mathbf{L})^{-1} \right]_{ij}, \tag{5}$$

where $\mathbf{I}$ is the identity matrix with the same dimension as $\mathbf{L}$. Consider F^i as the number of all spanning forests rooted at node i and F^{ij} as the number of the spanning forests rooted at i in which both nodes i and j has been assigned to the same tree. The element (i,j) of $(\mathbf{I}+\mathbf{L})^{-1}$ is equivalent to F^{ij}/F^i evaluating node similarity using the concept of spanning trees.

Finally, we should note that some Laplacian-based link prediction methods use various normalized versions of the network Laplacian matrix [31]. The normalized Laplacian matrix of a simple undirected unweighted network is a scaled version of the columns and rows of $\mathbf{L}$ that is more associated with the behavior of random walks [32]. There are different methods for normalizing the Laplacian matrix but the symmetric normalized Laplacian [33], $\boldsymbol{\mathcal{L}} = \mathbf{D}^{-1/2}(\mathbf{D}-\mathbf{A})\mathbf{D}^{-1/2}$, is more useful for link prediction as it is consistent with the fact that the similarity matrix corresponding to an undirected network must be symmetric. The similarity measures presented above can be defined in terms of this symmetric normalized Laplacian and, in this work, we use it for the design and analysis of perturbed Laplacian-based links prediction schemes.

2.2 Structural perturbation of the adjacency matrix

The idea of approximating the solution of a difficult problem as a deviation from a solvable simpler problem has been extended to matrices [34], such that the eigenpairs of a perturbed matrix can be approximated using those of the unperturbed one. Let λ_i be the eigenvalues and $\mathbf{x}_i$ be the corresponding eigenvectors of matrix $\mathbf{M}$. Consider $\tilde{\mathbf{M}} = \mathbf{M} + \boldsymbol{\Delta}\mathbf{M}$ as the perturbation of $\mathbf{M}$, where $\Delta\mathbf{M}$ is a small perturbation matrix with the same dimension as $\mathbf{M}$. Based on perturbation theory, the eigenvalues and eigenvectors of the perturbed matrix $\tilde{\mathbf{M}}$ can be found by correcting those of the unperturbed one as $\lambda_i + \Delta\lambda_i$ and $\mathbf{x}_i + \Delta\mathbf{x}_i$, respectively. The eigenvalues and eigenvectors of the perturbed matrix fulfill the eigenfunction equation

$$(\mathbf{M}+\Delta\mathbf{M})(\mathbf{x}_i+\Delta\mathbf{x}_i) = (\lambda_i+\Delta\lambda_i)(\mathbf{x}_i+\Delta\mathbf{x}_i). \quad (6)$$

If we assume that the perturbation does not significantly change the structure of the unperturbed matrix, the eigenvectors of $\mathbf{M}$ are unchanged by the perturbation, i.e., $\mathbf{x}_i + \Delta\mathbf{x}_i \approx \mathbf{x}_i$. Left-multiplying both sides of Eq. 6 by the transposed of the eigenvectors $\mathbf{x}_i^T$, keeping the first-order terms and ignoring the higher-order ones yields $\Delta\lambda_i$ as

$$\Delta\lambda_i \approx \frac{\mathbf{x}_i^T \Delta\mathbf{M}\mathbf{x}_i}{\mathbf{x}_i^T\mathbf{x}_i}. \quad (7)$$

The first-order matrix perturbation method outlined above can be used to characterize the structure of complex networks [10]. In this case, matrix $\mathbf{M}$ is a representation of the connectivity structure of a network and its eigenvectors reflect structural feaures [35] that are assumed to remain unchanged if the disturbance is weak.

Now, consider an observed network with a few missing links $G_{obs}(V, E_{obs})$. Link prediction methods aim to find the missing links of G_{obs} to build an inferred network $G_{inf}(V, E_{inf})$ as an approximation of the complete network G, where both G_{inf} and G have the same number of links. The first-order matrix perturbation method can be applied in combination with link prediction methods to approximate the structural features of G given G_{obs}. To this end, we randomly split the observed links into two disjoint subsets $E_{obs} = E_R + \Delta E$. A small fraction of links ΔE is considered as the disturbance and make the perturbation graph ΔG, and the remaining links E_R make a reduced graph $G_R(V, E_R)$. Then, matrix G_R is perturbed by ΔG and the resulting matrix is employed to formulate a similarity index for missing link prediction and finding G_{inf}.

For instance, the structural perturbation-based link prediction method SPM [10] chooses the adjacency matrix as the structural representation of a network. Therefore, after randomly splitting the links in E_{obs} into E_R and ΔE, and using the adjacency matrix $\mathbf{A}_R$ as G_R and $\Delta\mathbf{A}$ as ΔG, it perturbs $\mathbf{A}_R$ by $\Delta\mathbf{A}$. In other words, the perturbed matrix $\tilde{\mathbf{A}}_R = \mathbf{A}_R + \Delta\mathbf{A}$ can be approximated as

$$\tilde{\mathbf{A}}_R \approx \sum_{i=1}^{N} (\lambda_i + \Delta\lambda_i)\mathbf{x}_i\mathbf{x}_i^T, \quad (8)$$

where λ_i and $\mathbf{x}_i$ are the eigenvalues and eigenvectors of $\mathbf{A}_R$. Under the hypothesis that small random perturbations do not change the structural features of the network, the values in $\tilde{\mathbf{A}}_R$ corresponding to unconnected node pairs can be used to measure similarity scores for unobserved pairs. The procedure is repeated many times for different realizations of the perturbation set and the final perturbed matrix $\tilde{\mathbf{A}}$ is obtained as an average. Finally, SPM ranks its values in descending order so as to find the missing links and generates G_{inf} by adding to G_{obs} as many links at the top of the list as needed to supplement unobserved connections [10].

3 Perturbation of the Normalized Laplacian for the Prediction of Missing Links

First, we describe how to apply the first-order matrix perturbation method to the normalized Laplacian and second, we propose a general algorithm for predicting missing links in complex networks based on the perturbed Laplacian that can be leveraged by different Laplacian-based link prediction techniques.

3.1 Laplacian perturbation method

We consider the graph normalized Laplacian as the selected structural representation of a network and use it for link prediction. We apply the structural perturbation method to obtain $\tilde{\boldsymbol{\mathcal{L}}}$ with the assumptions that a group of links is predictable if removing them has only a small effect on the network's structural features and independent perturbations produce strongly correlated effects so that the perturbation adds useful structural information to G_{obs}. Consequently, the resulting matrix is employed to formulate link prediction similarity indices.

To obtaining $\tilde{\boldsymbol{\mathcal{L}}}$, we randomly split the observed links of a network into the two disjoint subsets $E_{obs} = E_R + \Delta E$, where a small fraction ρ of links is assigned to ΔE. We compute the normalized Laplacian for G_R and ΔG as $\boldsymbol{\mathcal{L}}_R = \mathbf{D}_R^{-1/2}(\mathbf{D}_R - \mathbf{A}_R)\mathbf{D}_R^{-1/2}$ and $\Delta\boldsymbol{\mathcal{L}} = \mathbf{D}_\Delta^{-1/2}(\mathbf{D}_\Delta -$

$\Delta\mathbf{A})\mathbf{D}_{\Delta}^{-1/2}$ respectively, where $\mathbf{D}_R$ and $\mathbf{D}_\Delta$ are the diagonal matrices of node degrees in G_R and ΔG. See Supplementary Material for more technical and practical information about the selected normalization protocol. Then, we approximate the perturbed Laplacian $\tilde{\boldsymbol{\mathcal{L}}}_R$ as $\tilde{\boldsymbol{\mathcal{L}}}_R = \boldsymbol{\mathcal{L}}_R + \Delta\boldsymbol{\mathcal{L}}$. We estimate the eigenvalues of $\tilde{\boldsymbol{\mathcal{L}}}_R$ as $\lambda_i + \Delta\lambda_i$ where λ_i are the eigenvalues of $\boldsymbol{\mathcal{L}}_R$ and $\Delta\lambda_i$ are computed using Eq. 7 by replacing $\Delta\mathbf{M}$ with $\Delta\boldsymbol{\mathcal{L}}$ and considering $\mathbf{x}_i$ as the eigenvectors of $\boldsymbol{\mathcal{L}}_R$. Finally, by keeping fixed the eigenvectors of $\boldsymbol{\mathcal{L}}_R$ before and after perturbation, $\tilde{\boldsymbol{\mathcal{L}}}_R$ is computed as

$$\tilde{\boldsymbol{\mathcal{L}}}_R \approx \sum_{i=1}^{N} (\lambda_i + \Delta\lambda_i)\mathbf{x}_i\mathbf{x}_i^T. \tag{9}$$

The resulting matrix $\tilde{\boldsymbol{\mathcal{L}}}$ is obtained by averaging $\tilde{\boldsymbol{\mathcal{L}}}_R$ in Eq. 9 over many realizations of the perturbation set, and it can be used to design Laplacian-based similarity indices as tools for the prediction of missing links. In the following section, we propose a general algorithm for perturbed Laplacian-based link prediction schemes.

Note that the correction to the eigenvalues $\Delta\lambda_i \approx \frac{\mathbf{x}_i^T \Delta\boldsymbol{\mathcal{L}}\mathbf{x}_i}{\mathbf{x}_i^T\mathbf{x}_i}$ is non-negative. It is due to the fact that $\Delta\boldsymbol{\mathcal{L}}$ is positive semidefinite, and therefore the numerator $\mathbf{x}_i\Delta\boldsymbol{\mathcal{L}}\mathbf{x}_i^T$ will be non-negative. Furthermore, the denominator is always equal to one when the eigenvectors are normalized to unit length. On the other hand, the eigenvalues of $\boldsymbol{\mathcal{L}}_R$, λ_i, fulfill $\lambda_i \geq 0$. Hence, the eigenvalues of $\tilde{\boldsymbol{\mathcal{L}}}_R$, $\lambda_i + \Delta\lambda_i$, and the eigenvalues of its pseudo-inverse $\tilde{\boldsymbol{\mathcal{L}}}_R^{\dagger}$, $\frac{1}{\lambda_i+\Delta\lambda_i}$, are also non-negative. It is also true for the average matrix $\tilde{\boldsymbol{\mathcal{L}}}$ and for its pseudo-inverse $\tilde{\boldsymbol{\mathcal{L}}}^{\dagger}$. As a result, the symmetric positive semidefinite matrix $\tilde{\boldsymbol{\mathcal{L}}}^{\dagger}$ is a Gram matrix and its elements are implicitly the inner product of node pairs feature vectors.

3.2 A general LPM algorithm for Laplacian-based link prediction

In the previous section, we described the application of the first-order structural perturbation method to the normalized Laplacian, that we name the Laplacian perturbation method LPM. In Algorithm. 1, we give a general algorithm for the prediction of missing links based on LPM.

Algorithm 1 LPM algorithm for link prediction

Input: An observed network $G_{obs}(V, E_{obs})$ and a similarity index generally defined as $S^{LPM} = f(\tilde{\boldsymbol{\mathcal{L}}})$
Output: A list of predicted links

1: $\tilde{\boldsymbol{\mathcal{L}}} \leftarrow \mathbf{0}$ ▷ It keeps the average of $\tilde{\boldsymbol{\mathcal{L}}}_R$
2: **for** $iter \leftarrow 1$ to n **do**
3: Randomly assign a fraction ρ of the links in E_{obs} to ΔE and constitute $\Delta G(V, \Delta E)$ with adjacency matrix $\Delta\mathbf{A}$
4: $E_R = E_{obs} - \Delta E$ and constitute $G_R(V, E_R)$ with adjacency matrix $\mathbf{A}_R$
5: $\Delta\boldsymbol{\mathcal{L}} = \mathbf{D}_{\Delta}^{-1/2}(\mathbf{D}_\Delta - \Delta\mathbf{A})\mathbf{D}_{\Delta}^{-1/2}$
6: $\boldsymbol{\mathcal{L}}_R = \mathbf{D}_{R}^{-1/2}(\mathbf{D}_R - \mathbf{A}_R)\mathbf{D}_{R}^{-1/2}$
7: Compute the eigenpairs of $\boldsymbol{\mathcal{L}}_R$: $(\lambda_i, \mathbf{x}_i)$
8: $\Delta\lambda_i \approx \frac{\mathbf{x}_i^T \Delta\boldsymbol{\mathcal{L}}\mathbf{x}_i}{\mathbf{x}_i^T\mathbf{x}_i}$
9: $\tilde{\boldsymbol{\mathcal{L}}}_R \approx \sum_{i=1}^{N}(\lambda_i + \Delta\lambda_i)\mathbf{x}_i\mathbf{x}_i^T$
10: $\tilde{\boldsymbol{\mathcal{L}}} \leftarrow \tilde{\boldsymbol{\mathcal{L}}} + \tilde{\boldsymbol{\mathcal{L}}}_R$
11: $\tilde{\boldsymbol{\mathcal{L}}} \leftarrow \tilde{\boldsymbol{\mathcal{L}}}/n$
12: Measure the similarity between node pairs using the resulting $\tilde{\boldsymbol{\mathcal{L}}}$ and the input similarity index $S^{LPM} = f(\tilde{\boldsymbol{\mathcal{L}}})$
13: Rank the unconnected node pairs (i, j) based on their similarity score $S^{LPM}(i, j)$ in descending order
14: **return** the links at the top of the ranked list as the predicted links

This algorithm takes an observed network G_{obs} and a generally defined similarity index S^{LPM}, which is a function of the average perturbed normalized Laplacian of G_{obs}, and returns a list of the predicted links. Steps 2 to 10 of Algorithm 1 are iterated n times with independent perturbation sets ΔE to obtain an ensemble of $\tilde{\boldsymbol{\mathcal{L}}}_R$. Its average over realizations, computed in step 11 and denoted as $\tilde{\boldsymbol{\mathcal{L}}}$, is used by the similarity index S^{LPM} in step 12 to assign each unconnected node pair a similarity value. Finally, unobserved links are ranked in descending order according to their similarity values to infer the missing links. Note that calculating a similarity index for every $\tilde{\boldsymbol{\mathcal{L}}}_R$ in the resulting ensemble and reporting the average accuracy will generally give lower performance as verified in empirical results. Hence, LPM is general enough to be applied to different Laplacian-based link prediction methods even if they take different similarity functions $f(\tilde{\boldsymbol{\mathcal{L}}})$ as input.

4 LPM LINK PREDICTION TECHNIQUES

In this section, we introduce specific LPM link prediction schemes that employ $\tilde{\boldsymbol{\mathcal{L}}}$ as the structural information source for inferring new links. To this end, we start from the previously described Laplacian-based link prediction methods Pinv, Cos, ACT, and MFA and adapt them to exploit perturbation theory in their procedures. All the proposed methods follow Algorithm 1 even though each takes its own specific similarity index as input.

4.1 Pinv-Per method

The perturbed version of the Pinv link prediction method, see Eq. (1), is called Pinv-Per. We first compute $\tilde{\boldsymbol{\mathcal{L}}}$ from the observed network by following the steps 1 to 11 of Algorithm 1. Therefore, the similarity index of the Pinv-Per can be formulated as

$$S^{Pinv-Per}(i, j) = \tilde{\mathcal{L}}_{ij}^{\dagger}, \tag{10}$$

where $\tilde{\mathcal{L}}_{ij}^{\dagger}$ is the element (i, j) of the pseudo-inverse of the average perturbed normalized Laplacian matrix.

4.2 Cos-Per method

The perturbed version of the Cos link prediction method is called Cos-Per and as its unperturbed counterpart calculates the cosine of the angle between node pairs feature vectors, in this case using $\tilde{\boldsymbol{\mathcal{L}}}^{\dagger}$. Therefore, the similarity index of this method is defined as

$$S^{Cos-Per}(i, j) = \frac{\tilde{\mathcal{L}}_{ij}^{\dagger}}{\sqrt{\tilde{\mathcal{L}}_{ii}^{\dagger}\tilde{\mathcal{L}}_{jj}^{\dagger}}}. \tag{11}$$

4.3 MFA-Per method

Based on the regularized Laplacian kernel [36], that is a general form of MFA formulated as $(\mathbf{I} + \alpha\boldsymbol{\mathcal{L}})^{-1}$, we suggest a perturbation based link prediction method applying $\tilde{\boldsymbol{\mathcal{L}}}$ in its similarity index as

$$S^{MFA-Per}(i, j) = [(\mathbf{I} + \alpha\tilde{\boldsymbol{\mathcal{L}}})^{-1}]_{ij} \tag{12}$$

In contrast to the unperturbed MFA, the elements of the MFA-Per do not reflect the relative forest accessibility between node pairs. However, when $\alpha = 1$, $(\mathbf{I}+\tilde{\boldsymbol{\mathcal{L}}})^{-1}$ is closely related to $\tilde{\boldsymbol{\mathcal{L}}}^{\dagger}$ such that they both have the same eigenvectors and their eigenvalues are related by ${\lambda_i}^{MFA-Per} = {\lambda_i}^{Pinv-Per}/1 + {\lambda_i}^{Pinv-Per}$. We have discussed above that ${\lambda_i}^{Pinv-Per}$, that correspond to the eigenvalues of the perturbed normalized Laplacian pseudo-inverse, are all non-negative so it will also be true for ${\lambda_i}^{MFA-Per}$. Therefore, $(\mathbf{I} + \tilde{\boldsymbol{\mathcal{L}}})^{-1}$ is again a Gram matrix. Indeed, similar to the Pinv-Per and Cos-Per, the elements of the MFA-Per can be seen as the inner product of node feature vectors. However, we should note that each method has a different set of node feature vectors.

4.4 ACT-Per method

The average commute time between two nodes of a network formulated in Eq. 3 is also computable via the pseudo-inverse of the network normalized Laplacian matrix $\boldsymbol{\mathcal{L}}^{\dagger}$ as $n(i,j) = |E|(\frac{\mathcal{L}^{\dagger}_{ii}}{d_i} + \frac{\mathcal{L}^{\dagger}_{jj}}{d_j} - \frac{2\mathcal{L}^{\dagger}_{ij}}{\sqrt{d_i d_j}})$, where d_i is the degree of node i [37]. Then, the similarity between node pairs in the perturbation approach can be estimated as

$$S^{ACT-Per}(i,j) = \frac{1}{|E|(\frac{\tilde{\mathcal{L}}^{\dagger}_{ii}}{\tilde{d}_i} + \frac{\tilde{\mathcal{L}}^{\dagger}_{jj}}{\tilde{d}_j} - \frac{2\tilde{\mathcal{L}}^{\dagger}_{ij}}{\sqrt{\tilde{d}_i \tilde{d}_j}})}, \qquad (13)$$

where $\tilde{d}_i$ is the degree of node i in the graph corresponding to $\tilde{\boldsymbol{\mathcal{L}}}$ that can be approximated by the degree d_i in the observed network.

Unlike Pinv-Per, Cos-Per, and MFA-Per that measure the similarity between node pairs based on the inner product of node feature vectors, we will prove that ACT-Per computes the reciprocal of the Euclidean distance between node pairs to gauge how similar the nodes in the pair are. The average commute time between two nodes in the graph associated to the perturbed Laplacian is given by [37]

$$n(i,j) = |E|(\mathbf{e}_i - \mathbf{e}_j)^T \tilde{\mathbf{D}}^{\frac{-1}{2}} \tilde{\boldsymbol{\mathcal{L}}}^{\dagger} \tilde{\mathbf{D}}^{\frac{-1}{2}} (\mathbf{e}_i - \mathbf{e}_j), \qquad (14)$$

where $\tilde{\mathbf{D}}$ is the diagonal matrix of node degrees and $\mathbf{e}_i$ is the ith column of the identity matrix of size N. By replacing $\tilde{\boldsymbol{\mathcal{L}}}^{\dagger}$ with its eigenvalue decomposition $\mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T$, where $\boldsymbol{\Sigma}$ is a diagonal matrix of the eigenvalues sorted in decreasing order and $\mathbf{V}$ is a column matrix of the corresponding eigenvectors of $\tilde{\boldsymbol{\mathcal{L}}}^{\dagger}$, and considering $\mathbf{y}_i = \boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{V}^T\tilde{\mathbf{D}}^{\frac{-1}{2}}\mathbf{e}_i$, Eq. 14 can be written as

$$n(i,j) = |E|(\mathbf{y}_i - \mathbf{y}_j)^T(\mathbf{y}_i - \mathbf{y}_j) = |E|\|(\mathbf{y}_i - \mathbf{y}_j)\|^2. \qquad (15)$$

Therefore, computing the average commute time between node pairs is equivalent to assigning each node i an extracted feature vector $\mathbf{y}_i$ and calculating the Euclidean distance between them. The smaller the Euclidean distance between two nodes, the more similar the nodes according to the ACT-Per similarity index.

4.5 Cos-CN method

Finally, we propose to combine the global structural information included in $\tilde{\boldsymbol{\mathcal{L}}}^{\dagger}$ with the local information from node degrees and common neighbors to design a similarity index called Cos-CN that is computed as

$$S^{Cos-CN}(i,j) = \sum_{z \in (\Gamma(i) \cap \Gamma(j))} \frac{1}{|\Gamma(z)|}[S^{Cos-Per}(i,z) + S^{Cos-Per}(j,z)], \qquad (16)$$

where $\Gamma(i)$ is the set of neighbors of node i and $|.|$ is the cardinality of a set. To measure the similarity between two nodes, the Cos-CN index calculates a weighted sum of Cos-Per similarity values between the given nodes and their common neighbors computed using Eq. 11 such that it assigns more weights to less connected common neighbors.

5 Experimental Results

We have used LPM link prediction techniques described above to predict missing links in model and real-worlds networks. In all the experimental tests, we randomly separated a small fraction q of links from the complete network to act as a probe set simulating missing links. The remaining set acts as training set and constitute G_{obs}. We then measure the link prediction methods performance in terms of the area under the receiver operating characteristic curve (AUC) [38] and the precision [39].

The AUC evaluates accuracy according to the entire list of unobserved links sorted in decreasing order of their similarity scores. It computes the probability that a link prediction technique assigns a higher similarity score to the corresponding node pair of a randomly selected missing link (i.e., a link in the probe set) than that of a randomly selected non-existing link (i.e., a link in $U - E$, where U is the universal set of all possible links and E is the set of links in the complete network). If in n independent pairwise comparisons of randomly chosen missing and non-existing links, n' times the missing link has higher score and n'' times the two links have the same scores, the AUC value will be

$$AUC = \frac{n' + 0.5n''}{n} \qquad (17)$$

On the other hand, precision focuses only on the links at the top of the ranking with highest similarity scores. Therefore, if a set of L links at the top of the ordered list contains L_r missing links, the precision will be equal to L_r/L.

Once the observed graphs are obtained, the perturbation sets ΔE are randomly constructed by separating a fraction $\rho = 0.1$ of the links from G_{obs}. The perturbation procedure is done $n = 10$ times over G_{obs} with independent sets ΔE, and parameter α in MFA-Per is set to 1. In addition, the largest connected component is considered in networks with more than one component, and the generated training sets contain no isolated nodes.

5.1 Network Models

We first consider synthetic networks produced by well-known probabilistic network models including the $\mathbb{S}^1$ model [40], generating maximally random networks with given level of clustering and sequence of expected degrees, the degree-corrected stochastic block model (dc-SBM) [41],

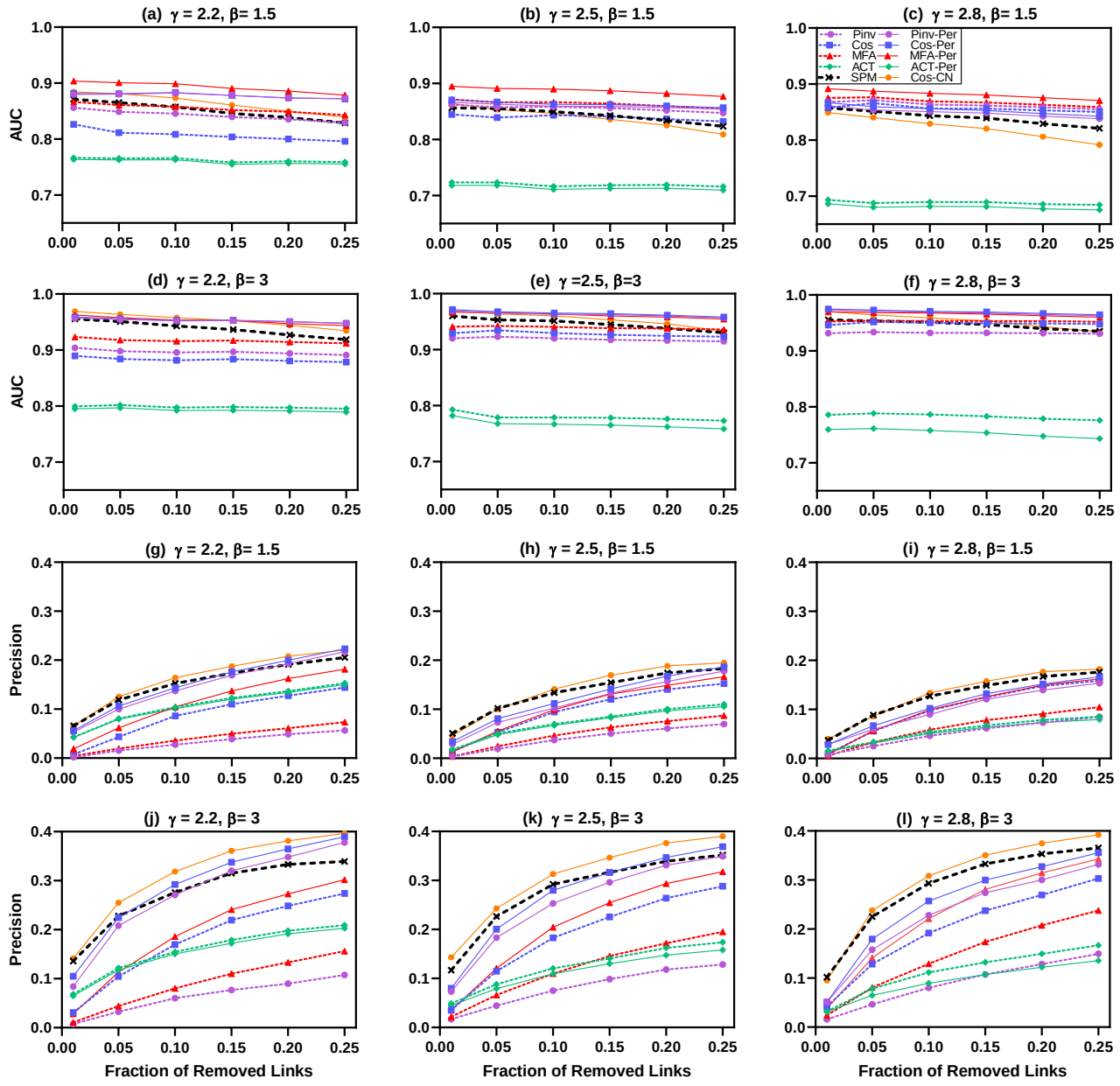

Fig. 1. The average AUC and precision of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q for the $\mathbb{S}^1$ model. Parameter β controlling the level of clustering is increased from up to down and the power-law degree distribution exponent γ is incremented from left to right. For almost all γ and β, LPM link prediction methods (except ACT-Per) defeat their unperturbed counterparts. They also exhibit higher AUC than SPM, and Cos-CN surpasses SPM in terms of precision. Moreover, the precision improvements achieved by LPM link prediction techniques are greater in more complex $\mathbb{S}^1$ networks containing larger hubs (smaller γ) and higher level of clustering (larger β).

a block model that generates random networks with heterogeneous node degrees and community structure and, finally, the soft configuration model (sCM) [42], generating maximally random networks with respect to a given expected degree sequence. See Supplementary Material for more details about these network models.

In the following experiments, for every ensemble characterized by specific model parameters, we have generated 10 different networks with $N = 300$ nodes and average degree of $\langle k \rangle = 10$. Then for every network and each value of q, we have constructed 10 disjoint training and probe sets on which the link prediction methods were applied. In dc-SBM networks, parameter λ making a balance between random and group structures has been fixed to $\lambda = 0.5$.

In Fig. 1, we show the performance of the different LPM link prediction methods in synthetic networks of the $\mathbb{S}^1$ ensemble compared to the unperturbed versions as a function of the fraction of removed links and for different values of the parameters of the models. The $\mathbb{S}^1$ model is a geometric network model [43] that gives a very good description of structural connectivity in real-world networks including typical features such as sparsity, the small-world

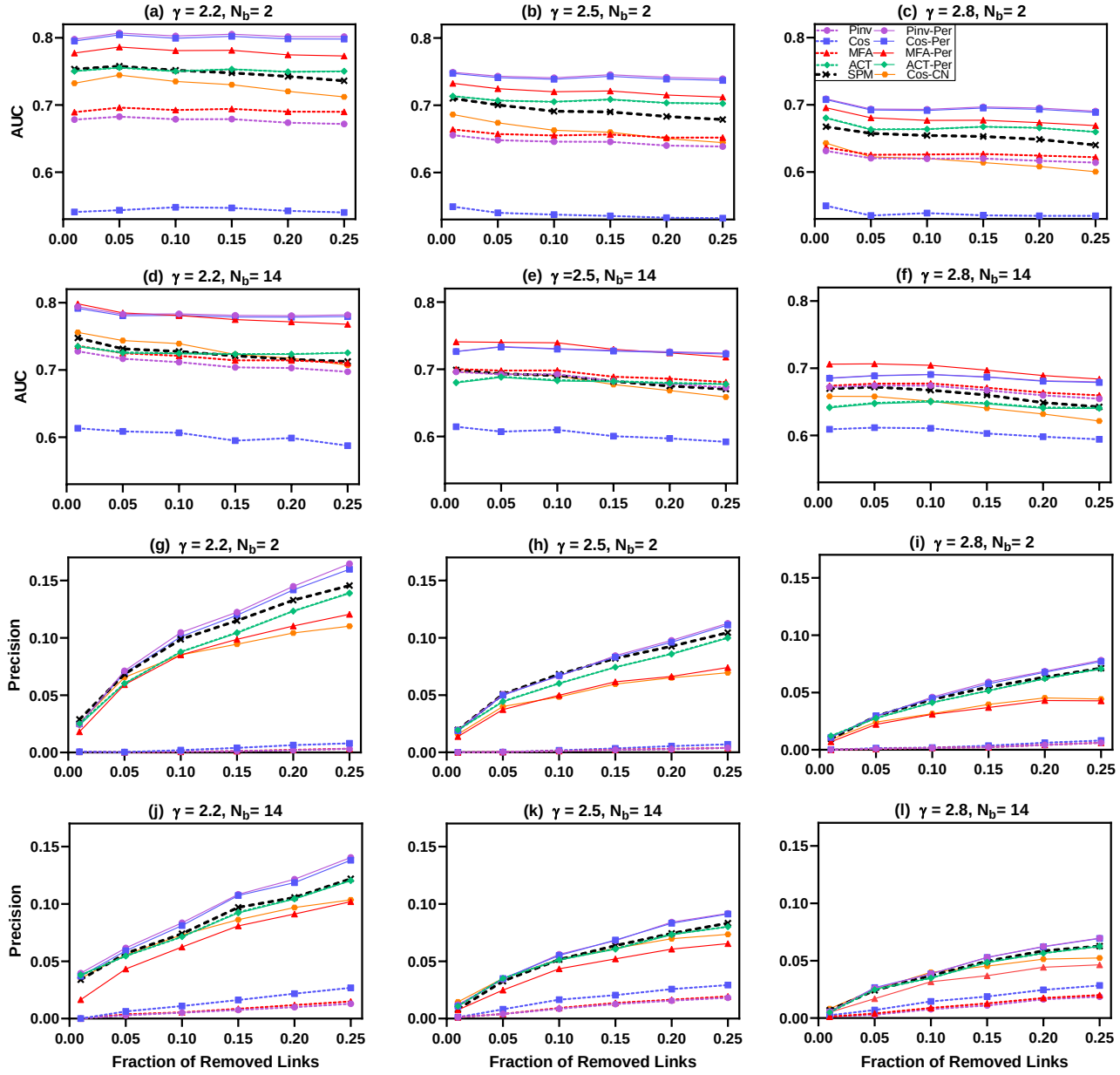

Fig. 2. The average AUC and precision of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q for the dc-SBM model. The number of equiprobable blocks N_b is increased from up to down and the power-law degree distribution exponent γ is incremented from left to right. For every γ and N_b, LPM link prediction methods (except ACT-Per) defeat their unperturbed counterparts. They also exhibit higher AUC values than SPM, and Pinv-Per and Cos-per outperform SPM with regard to precision. Moreover, for a fixed N_b, the increas of γ decreases the performance gaps between the best LPM link prediction method and SPM.

property, heterogeneous degree distributions, and high levels of clustering [44]. The combination of these models with statistical inference techniques allow to obtain maps of real networks in the hyperbolic plane where distances inform about the likelihood of connections [45]. Beyond visualization, these representations can be exploited for efficient navigation [46], [47], for uncovering patterns such as self-similarity [40], [47], [48], [49] and communities of strongly interacting nodes [50], [51], [52], [53], and to sustain a renormalization procedure that brings to light the multiscale nature of networks [47].

Fig. 1 proves that the perturbed link prediction methods are very efficient in $\mathbb{S}^1$ synthetic networks. LPM link prediction methods Pinv-Per, Cos-Per and MFA-Per almost always surpass their unperturbed counterparts both in terms of AUC and precision, whereas ACT-Per usually exhibits lower AUC and precision than ACT. It is also clear from Fig. 1(a-f) that LPM link prediction techniques, except ACT-Per and Cos-CN, have higher AUC than SPM. By varying the parameters of the model—γ, that controls the exponent in the power-law degree distribution, and β, that controls the level of clustering so that the higher the β the higher

TABLE 1
Performance of link prediction methods measured by AUC in a set of real-world networks. The probe sets contains a $q = 0.1$ fraction of the links in the complete networks and the presented results are the average of 10 independent runs. For LPM link prediction techniques, AUC values of their unperturbed counterparts are reported in the parenthesis.

Networks	CN	RA	AA	CH	FBM	SPM	Pinv-Per	Cos-Per	MFA-Per	Cos-CN
Karate	0.711	0.762	0.760	0.563	0.806	0.760	0.809 (0.747)	0.803 (0.721)	**0.840** (0.752)	0.768
Lesmis	0.95	**0.96**	**0.96**	0.904	0.936	0.945	0.942 (0.887)	0.94 (0.886)	0.948 (0.92)	**0.96**
Polbooks	0.90	0.91	0.909	0.802	0.899	0.901	0.883 (0.869)	0.894 (0.894)	**0.927** (0.906)	0.911
ACM2009	0.784	**0.789**	0.786	0.784	0.772	0.777	0.777 (0.772)	0.776 (0.682)	**0.789** (0.775)	0.788
WTW	0.811	0.858	0.854	0.699	**0.901**	0.819	**0.901** (0.767)	0.896 (0.666)	0.874 (0.792)	0.867
Congress vote	0.763	0.772	0.772	0.564	0.843	0.785	**0.864** (0.786)	0.862 (0.716)	0.811 (0.817)	0.771
USAir	0.958	**0.974**	0.969	0.937	0.951	0.944	0.968 (0.907)	0.968 (0.959)	0.962 (0.940)	**0.974**
Netscience	0.974	0.977	0.977	0.825	0.960	0.972	0.983 (0.946)	0.984 (0.967)	**0.985** (0.973)	0.977
Email	0.856	0.858	0.858	0.704	0.891	0.896	**0.928** (0.908)	**0.928** (0.908)	0.916 (0.919)	0.857
Neural	0.850	0.871	0.866	0.769	0.883	0.892	0.859 (0.861)	0.862(0.862)	**0.910** (0.873)	0.874
Infectious	0.943	0.948	0.947	0.863	0.956	0.944	0.956 (0.913)	**0.961** (0.946)	0.960 (0.960)	0.948
Metabolic	0.921	**0.959**	0.954	0.871	0.914	0.932	0.938 (0.889)	0.941 (0.891)	0.952 (0.905)	**0.959**
Polblogs	0.926	0.930	0.929	0.902	0.935	0.934	**0.945** (0.892)	**0.945** (0.928)	0.925 (0.907)	0.932

the level of clustering—keeping values in the typical range observed in real networks, we analize the dependency of LPM link prediction methods on the structural features of $\mathbb{S}^1$ synthetic networks. By fixing β and increasing γ in the rows of Fig. 1(a-f) from left to right, the AUC gap between the best LPM link prediction technique and SPM is approximately invariant, but it is decreased by growing β in the columns of Fig. 1(a-f) from up to down for a fixed γ.

In terms of precision, Fig. 1(g-l) show that Cos-CN stands out as the best link prediction technique. When β is constant, the precision gap between Cos-CN and SPM is larger for the $\mathbb{S}^1$ networks with more heterogeneous degree distribution, while the two precision curves become closer together as the heterogeneity of node degrees is decreased by increasing γ. In addition, for a constant γ, the precision improvement achieved by the Cos-CN is greater for networks with higher clustering coefficient, that is larger value of β, as shown in the columns of Fig. 1(g-l). That is to say, applying LPM for the prediction of missing links gives better performance in more complex $\mathbb{S}^1$ networks containing larger hubs and higher level of clustering. See Supplementary Material Figs. S7 and S8 displaying extensive comparisons of the methods performance in $\mathbb{S}^1$ networks for more values of parameter β.

Analogous results are displayed in Fig. 2 for synthetic dc-SBM networks. Again, the results highlight that perturbing the normalized Laplacian give rise to Pinv-Per, Cos-Per and MFA-Per methods with higher AUC and precision compared to their unperturbed counterparts. Nevertheless, both ACT and ACT-Per exhibit similar performance in these networks, in agreement with the behaviours observed in $\mathbb{S}^1$ networks. Also in accordance with the results for $\mathbb{S}^1$, LPM link prediction techniques, excluding ACT-Per and Cos-CN, defeat SPM with respect to AUC in dc-SBM, see Fig. 2(a-f). Moreover, we observe from Fig. 2(g-l) that Pinv-Per and Cos-Per surpass SPM in terms of precision. As presented by the rows of Fig. 2 from left to right, for a fixed number of blocks N_b, the AUC and precision gaps between the best LPM link prediction method and SPM are decreased with the increase of γ. Moreover, as anticipated, for a constant γ the AUC and precision gaps are not significantly changed with the increase of N_b, as shown by plots in the columns of Fig. 2 from up to down. It seems then that varying the local parameter N_b could not significantly affect the performance of LPM link prediction techniques, which are global. Figs. S9 and S10 in the Supplementary Material show results in synthetic dc-SBM networks for more values of parameter N_b.

The performance of the different LPM link prediction methods for synthetic networks produced by the sCM model is shown in Fig. S6 of the Supplementary Material for different values of the fraction of missing links and for exponents γ of the power-law degree distribution varied in the range of $[2, 3]$. Fig. S6 shows that Pinv-Per, Cos-Per and MFA-Per always surpass their unperturbed counterparts in terms of AUC and precision, while, ACT and ACT-per methods yield almost similar performance. The experiments also reveal that employing the perturbed normalized Laplacian matrices as the structural representations of sCM networks provides more accurate lists of predicted links as the Pinv-Per and Cos-Per could always outperform SPM in terms of both AUC and precision. We also observe that in sCM networks with more heterogeneous degree distribution (smaller values of γ), the performance gaps between LPM link prediction techniques and SPM are larger, while they are getting smaller as γ is increased.

According to the results for the synthetic networks, perturbing the normalized Laplacian matrix is beneficial to improve the performance of inner-product-based similarity indices—Pinv-Per, Cos-Per and MFA-Per.

Finally, Table. 1 in Supplementary Material compares AUC values of LPM link prediction methods, excluding ACT-Per which does not show good performance, with six widely applied link prediction schemes in the network models, when $q = 0.1$. Four of these schemes CN, AA, RA and CH extract local structural properties, while two of them SPM and FBM exploit global connectivity patterns for the prediction of missing link, see Supplementary Material

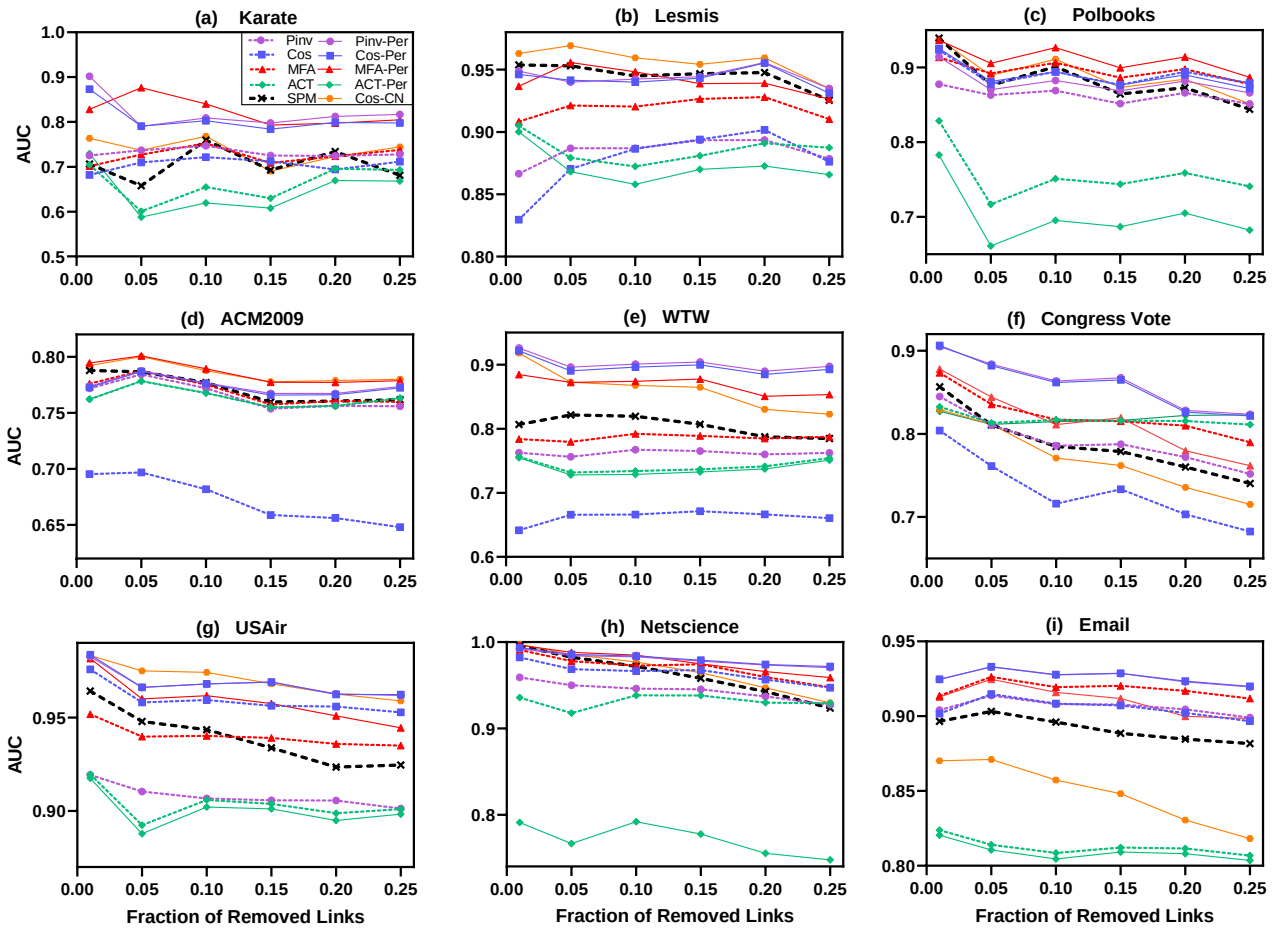

Fig. 3. AUC values of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q. For each real network and each value of q, we generated 10 different training and probe sets on which we applied these methods and computed the average AUC values. LPM link prediction methods (except ACT-Per) typically acheive higher AUC than their unperturbed counterparts, and SPM is also defeated by one or more LPM link prediction schemes.

for details. The results indicate that at least one LPM link prediction method has higher accuracy than state-of-the-art widely-applied link prediction schemes in model networks.

5.2 Real-world Networks

We assess the performance of LPM link prediction techniques in a collection of real-world complex networks from different domains, including social interactions between the members of a university karate club [54] (Karate), concurrences of characters in the novel Les Misérables by Victor Hugo [55] (Lesmis), co-purchasing of political books on Amazon [56] (Polbooks), face to face contacts between the participants of the ACM Hypertext 2009 Conference [57] (ACM2009), international trade network in 2013 [52] (WTW), politicians that mention one another in their speaking in the United States Congress [58] (Congress Vote), the US Air transportation network [59] (USAir), co-authorship in the field of network science [60] (Netscience) and email communication at the university Rovira i Virgili in Tarragona, Spain [61] (Email). More real-world networks are reported in Table 1 and Supplementary Material Table1, Figs. S11 and S12. A brief description of these networks is provided in Supplementary Material. For all networks, we consider undirected unweighted versions restricted to the giant connected component. Their basic topological characteristics can be found in Supplementary Material Table 2.

In complete agreement with results above for network models, Pinv-Per, Cos-per and MFA-Per typically exhibit higher AUC values than their unperturbed versions Pinv, Cos and MFA in real-world networks, as depicted in Fig. 3 and Fig. S11 in Supplementary Material. In addition, ACT-Per usually shows lower AUC than ACT, which once again confirms that perturbing the network normalized Laplacian can not improve the performance of distance-based similarity indices. On the other hand, it is apparent from these figures that SPM is beaten in terms of accuracy by one or more LPM link prediction techniques. Table 1 also compare the AUC values of the proposed perturbed schemes with the aforementioned state-of-the-art local and global link prediction methods, when the fraction of missing links q is set to 0.1. The results indicate that in all real-world networks at least one LPM link prediction scheme is at least as good as or better than them.

In terms of precision, the results also agree with those

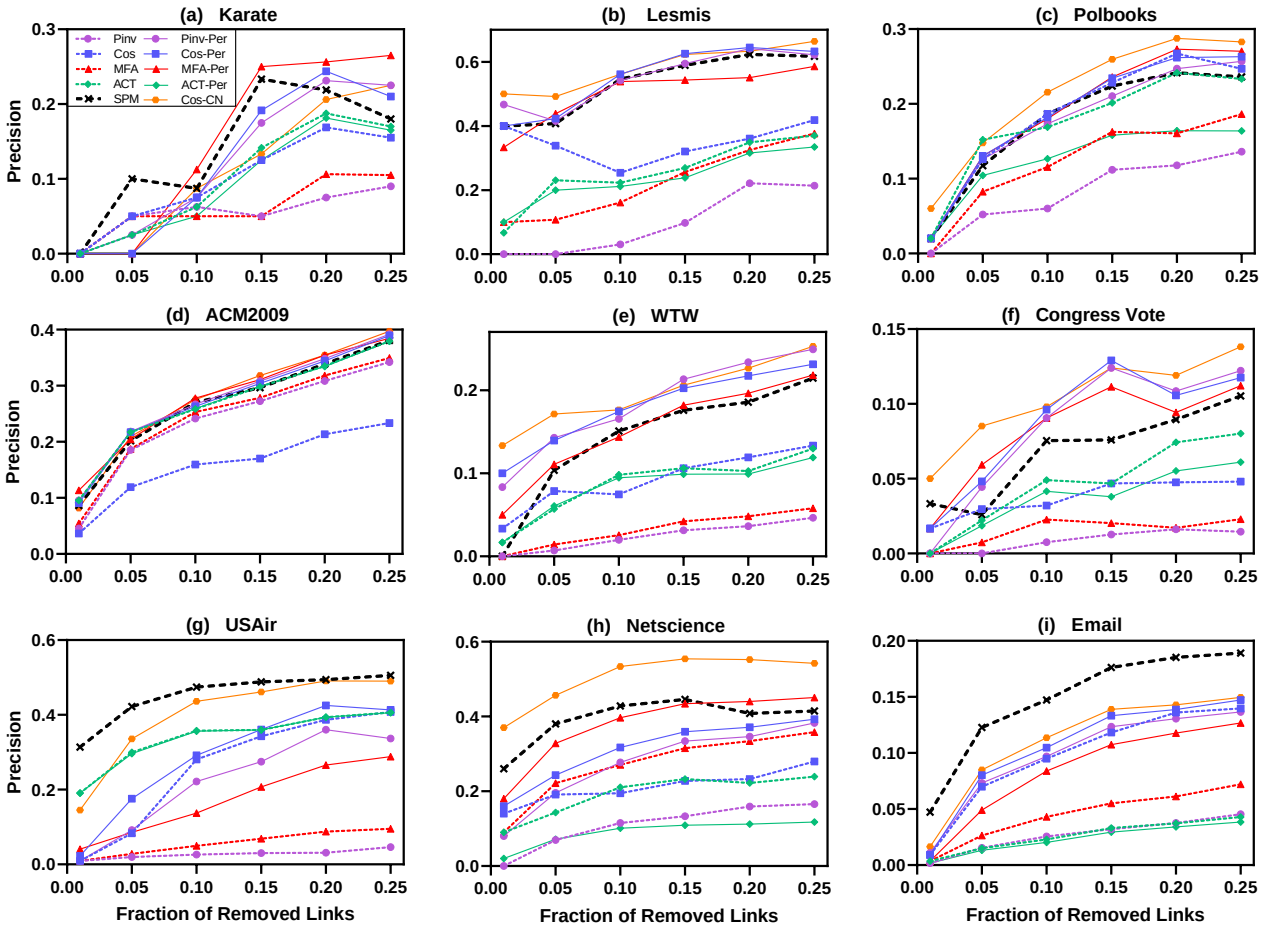

Fig. 4. Precision values of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q. For each real network and each value of q, we generated 10 different training and probe sets on which we applied link prediction methods and computed the average precision. LPM link prediction methods (except ACT-Per) usually outperform their unperturbed counterparts. Moreover, Cos-CN typically shows higher precision than other LPM link prediction schemes, and it also surpasses SPM in some of the real-networks.

obtained for model networks. The perturbed normalized Laplacian-based link prediction methods, excluding ACT-Per, typically outperform their unperturbed counterparts in all the analyzed real-world networks as shown in Fig. 4 and Fig. S12 in Supplementary Material. In addition, similar to the results achieved for the $\mathbb{S}^1$ model, that reflect the impact of properties observed in real-world networks such as heterogeneous degree distributions and strong clustering, Cos-CN usually has the best precision in real networks as compared to other LPM link prediction schemes, and it is also able to outperform SPM in some of them, see Fig. 4 and Fig. S12 in the Supplementary Material.

6 Conclusions

This paper presents a novel link prediction scheme that perturbs the normalized Laplacian of an observed network with the intention of extracting hidden structural information encoded in it to obtaining more accurate link prediction methods. We introduced a perturbative algorithm valid for any Laplacian-based link prediction method and found that perturbative Laplacian-based link prediction techniques outperform their unperturbed counterparts when similarity indices are based on the inner product of node feature vector —methods Pinv-Per, Cos-Per and MFA-Per. In contrast, methods that rely on Euclidean distances between nodes —ACT-Per— may not benefit from LPM. We measured performance in terms of AUC and precision and our results hold both for synthetic networks generated by realistic models and real-world networks.

Beyond Laplacian-based link prediction methods, we have also found that LPM link prediction techniques can also outperform a set of state-of-the-art local and global link prediction methods based on the adjacency matrix. In particular, we compared our approach with the SPM link prediction method that perturbs the adjacency matrix to infer new links. The results indicate that LPM exhibits higher AUC and precision than SPM in synthetic networks generated by realistic models. We also showed that, as the synthetic networks become more complex by increasing their degree heterogeneity and clustering coefficient, our methods manage to beat SPM with a larger precision gap.

In real-world networks, experimental results demonstrate that LPM link prediction schemes outperform SPM in terms of AUC. However, SPM achieves higher precision

than LPM in some real networks. This feeds the question of which representation, Laplacian or adjacency or a combination of the two, better represents structural information to achieve the best performance in terms of precision when predicting missing links in real-world networks. Also, it will be necessary to investigate how the choice depends on specific topological features of the networks under consideration. The questions are challenging and other algorithms that rely on spectral properties of graphs, such as spectral clustering for community detection [62], face the same problem when dealing with strongly heterogeneous networks, for which different choices of the representation matrix give rise to disparate inference. As a result of our work, we hope to have increased awareness about the importance of the choice of the representation matrix for link prediction algorithms and about the need of further research in this direction.

Finally, another interesting avenue for future work would be the design of new link prediction strategies in complex networks based on machine learning techniques that take as input the node feature vectors encoded by the perturbed normalized Laplacian.

ACKNOWLEDGMENTS

A. G. gratefully acknowledges support from the Alexander von Humboldt Foundation for his visiting research at the University of Passau, Germany. M.A.S. acknowledges support from the Spanish Agencia Estatal de Investigación project number PID2019-106290GB-C22/AEI/10.13039/501100011033, and from the Generalitat de Catalunya grant number 2017SGR1064.

REFERENCES

[1] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, "Link prediction techniques, applications, and performance: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 553, p. 124289, 2020.

[2] D. Helbing *et al.*, "Saving human lives: What complexity science and information systems can contribute," *Journal of Statistical Physics*, vol. 158, no. 3, pp. 735–781, 2015.

[3] M. Jalili and M. Perc, "Information cascades in complex networks," *Journal of Complex Networks*, vol. 5, no. 5, pp. 665–693, 2017.

[4] G. García-Pérez, R. Aliakbarisani, A. Ghasemi, and M. Á. Serrano, "Precision as a measure of predictability of missing links in real networks," *Physical Review E*, vol. 101, no. 5, p. 052318, 2020.

[5] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM Computing Surveys*, vol. 49, no. 4, pp. 1–33, 2016.

[6] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, p. 025102, 2001.

[7] L. A. Adamic and E. Adar, "Friends and neighbors on the Web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[8] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.

[9] A. Muscoloni, U. Michieli, and C. V. Cannistraci, "Local-ring network automata and the impact of hyperbolic geometry in complex network link-prediction," *arXiv preprint arXiv:1707.09496*, 2017.

[10] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, and H. E. Stanley, "Toward link predictability of complex networks," *Proceedings of the National Academy of Sciences*, vol. 112, no. 8, pp. 2325–2330, 2015.

[11] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[12] Z. Liu, J.-L. He, K. Kapoor, and J. Srivastava, "Correlations between community structure and link formation in complex networks," *PLoS ONE*, vol. 8, no. 9, p. e72908, 2013.

[13] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 355–369, 2007.

[14] P. Y. Chebotarev and E. V. Shamis, "The matrix-forest theorem and measuring relations in small social group," *Automation and Remote Control*, vol. 58, no. 9, pp. 1505–1514, 1997.

[15] ——, "On proximity measures for graph vertices," *Automation and Remote Control*, vol. 59, no. 10, pp. 1443–1459, 1998.

[16] G. Kirchhoff, "Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird," *Annalen der Physik und Chemie*, vol. 148, no. 12, pp. 497–508, 1847.

[17] N. M. M. de Abreu, "Old and new results on algebraic connectivity of graphs," *Linear Algebra and its Applications*, vol. 423, no. 1, pp. 53–73, 2007.

[18] L. Guo, C. Liang, and S. H. Low, "Monotonicity properties and spectral characterization of power redistribution in cascading failures," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Monticello, IL, USA: IEEE, 2017, pp. 918–925.

[19] K. Devriendt and P. Van Mieghem, "The simplex geometry of graphs," *Journal of Complex Networks*, vol. 7, no. 4, pp. 469–490, 2019.

[20] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[21] T. Vallès-Català, T. P. Peixoto, M. Sales-Pardo, and R. Guimerà, "Consistencies and inconsistencies between model selection and link prediction in networks," *Physical Review E*, vol. 97, no. 6, p. 062316, 2018.

[22] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[23] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Montréal, Canada: Curran Associates Inc., 2018, pp. 5171—5181.

[24] B. Mohar, E. Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk, "The Laplacian spectrum of graphs," *Graph Theory, Combinatorics, and Applications*, vol. 2, pp. 871–898, 1991.

[25] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*. New York, USA: Springer-Verlag, 2003.

[26] D. J. Klein and M. Randić, "Resistance distance," *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 81–95, 1993.

[27] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM Review*, vol. 50, no. 1, pp. 37–66, 2008.

[28] G. Ranjan and Z.-L. Zhang, "Geometry of complex networks and topological centrality," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 17, pp. 3833–3845, 2013.

[29] F. Dorfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 150–163, 2013.

[30] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, USA: Cambridge University Press, 2012.

[31] J. Kunegis and A. Lommatzsch, "Learning spectral graph transformations for link prediction," in *Proceedings of the 26th Annual International Conference on Machine Learning*. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 561–568.

[32] D. A. Spielman, "Algorithms, graph theory, and linear equations in Laplacian matrices," in *Proceedings of the International Congress of Mathematicians 2010 (ICM2010)*. Heyderabad, India: World Scientific, 2010, pp. 2698–2722.

[33] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI, USA: American Mathematical Society, 1997, no. 92.

[34] G. W. Stewart, *Matrix Perturbation Theory*. Cambridge, MA, USA: Academic Press, 1990.

[35] C. Godsil and G. F. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. New York, USA: Springer-Verlag, 2001, vol. 207.

[36] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines*. Springer, 2003, pp. 144–158.

[37] E. Bozzo, "The Moore–Penrose inverse of the normalized graph Laplacian," *Linear Algebra and its Applications*, vol. 439, no. 10, pp. 3038–3043, 2013.

[38] J. A. Hanley and B. J. McNeil, "A method of comparing the areas under receiver operating characteristic curves derived from the same cases," *Radiology*, vol. 148, no. 3, pp. 839–843, 1983.
[39] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.
[40] M. Á. Serrano, D. Krioukov, and M. Boguñá, "Self-similarity of complex networks and hidden metric spaces," *Physical Review Letters*, vol. 100, no. 7, p. 078701, 2008.
[41] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Physical Review E*, vol. 83, no. 1, p. 016107, 2011.
[42] J. Park and M. E. J. Newman, "Origin of degree correlations in the Internet and other networks," *Physical Review E*, vol. 68, no. 2, p. 026112, 2003.
[43] M. Boguñá, I. Bonamassa, M. De Domenico, S. Havlin, D. Krioukov, and M. Á. Serrano, "Network geometry," *Nature Reviews Physics*, vol. 3, no. 2, pp. 114–135, 2021.
[44] M. Boguñá, D. Krioukov, P. Almagro, and M. Á. Serrano, "Small worlds and clustering in spatial networks," *Physical Review Research*, vol. 2, no. 2, p. 023040, 2020.
[45] G. García-Pérez, A. Allard, M. Á. Serrano, and M. Boguñá, "Mercator: uncovering faithful hyperbolic embeddings of complex networks," *New Journal of Physics*, vol. 21, no. 12, p. 123033, 2019.
[46] M. Boguñá, F. Papadopoulos, and D. Krioukov, "Sustaining the Internet with hyperbolic mapping," *Nature Communications*, vol. 1, no. 1, p. 62, 2010.
[47] G. García-Pérez, M. Boguñá, and M. Á. Serrano, "Multiscale unfolding of real networks by geometric renormalization," *Nature Physics*, vol. 14, no. 6, pp. 583–589, 2018.
[48] M. Á. Serrano, D. Krioukov, and M. Boguñá, "Percolation in self-similar networks," *Physical Review Letters*, vol. 106, no. 4, p. 048701, 2011.
[49] M. Zheng, A. Allard, P. Hagmann, Y. Alemán-Gómez, and M. Á. Serrano, "Geometric renormalization unravels self-similarity of the multiscale human connectome," *Proceedings of the National Academy of Sciences*, vol. 117, no. 33, pp. 20 244–20 253, 2020.
[50] M. Á. Serrano, M. Boguñá, and F. Sagués, "Uncovering the hidden geometry behind metabolic networks," *Molecular BioSystems*, vol. 8, no. 3, pp. 843–850, 2012.
[51] K. Zuev, M. Boguñá, G. Bianconi, and D. Krioukov, "Emergence of soft communities from geometric preferential attachment," *Scientific Reports*, vol. 5, no. 1, p. 9421, 2015.
[52] G. García-Pérez, M. Boguñá, A. Allard, and M. Á. Serrano, "The hidden hyperbolic geometry of international trade: World Trade Atlas 1870–2013," *Scientific Reports*, vol. 6, no. 1, p. 33441, 2016.
[53] G. García-Pérez, M. Á. Serrano, and M. Boguñá, "Soft communities in similarity space," *Journal of Statistical Physics*, vol. 173, no. 3, pp. 775–782, 2018.
[54] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.
[55] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*. Reading, MA, USA: Addison-Wesley, 1993.
[56] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
[57] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, "What's in a crowd? Analysis of face-to-face behavioral networks," *Journal of Theoretical Biology*, vol. 271, no. 1, pp. 166–180, 2011.
[58] M. Thomas, B. Pang, and L. Lee, "Get out the vote: Determining support or opposition from Congressional floor-debate transcripts," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia: Association for Computational Linguistics, 2006, pp. 327–335.
[59] V. Batagelj and A. Mrvar, "Pajek datasets," 2006. [Online]. Available: http://vlado.fmf.uni-lj.si/pub/networks/data/
[60] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, p. 036104, 2006.
[61] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical Review E*, vol. 68, no. 6, p. 065103, 2003.
[62] J. F. Lutzeyer and A. T. Walden, "Comparing spectra of graph shift operator matrices," in *International Conference on Complex Networks and Their Applications*. Lisbon, Portugal: Springer, Cham, 2019, pp. 191–202.

Roya Aliakbarisani is currently a Ph.D candidate at K. N. Toosi university of technology, Tehran, Iran where she also received her M.Sc. in 2014 both in Artificial Intelligence. She has spent a sabbatical leave at the University of Barcelona, Barcelona, Spain from October 2017 to September 2018. Her research interests include network science, missing link prediction in complex networks and application of distance metric learning in networking.

Abdorasoul Ghasemi is an Associate Professor with the Faculty of Computer Engineering of K.N. Toosi University of Technology, Tehran, Iran. He received his Ph.D. and M.Sc. degrees from Amirkabir University of Technology (Tehran Polytechnique), Tehran, Iran, and his B.Sc. from the Isfahan University of Technology, all in Electrical Engineering. He has spent sabbatical leave with the computer science department at the University of California, Davis, CA, the USA (April 2017 to August 2018) and Max Planck Institute for the physics of complex systems, Dresden, Germany (Dec. 2020 to July 2021). He has been awarded the Alexander von Humboldt fellowship for experienced researchers in July 2021, working on resilient cyber-physical energy systems at the University of Passau, Germany. His research interests include network science and its engineering applications, including communications, energy, and cyber-physical systems using optimization and machine learning approaches.

M. Ángeles Serrano is an ICREA Research Professor at the Department of Condensed Matter Physics of the University of Barcelona and External Faculty at the Complexity Science Hub Vienna CSH. She obtained her Ph.D. in Physics at the University of Barcelona, a master in mathematics for finance from the CRM-Autonomous University of Barcelona, and completed her postdoctoral research at Indiana University (USA), the École Polytehcnique Fédérale de Lausanne (Switzerland) and IFISC Institute (Spain). She is interested in unraveling the universal principles and laws underlying the structure, function, and evolution of complex networks. Her research covers theoretical developments and applications to a variety of real systems, from international trade to the Internet and the brain. M. Ángeles obtained the Outstanding Referee Award of the American Physical Society. She is a founding member of Complexitat, the Catalan network for the study of complex systems, and a promoter member of UBICS, the Universitat de Barcelona Institute of Complex Systems.

Supplementary material
Perturbation of the normalized Laplacian matrix for the prediction of missing links in real networks

Roya Aliakbarisani, Abdorasoul Ghasemi, and M. Ángeles Serrano

S.1 BASELINE LINK PREDICTION METHODS

In this section, we review five extensively used local and global state-of-the-art link prediction schemes to which we have compared the performance of our suggested link prediction methods employing the proposed general link prediction algorithm called Laplacian perturbation method (LPM).

S.1.1 Common Neighbors (CN)

The simplest local link prediction method CN [S1] measures the similarity between unconnected node pairs by counting the number of neighbors they have in common as

$$S^{CN}(i,j) = |\Gamma(i) \cap \Gamma(j)|, \tag{S1}$$

where $\Gamma(i)$ is the set of the neighobrs of node i and $|.|$ refers to the cardinality of a set.

S.1.2 Resource Allocation (RA) & Adamic Adar (AA)

Two improved versions of the CN index assigning higher weight values to less-connected common neighbors using the reciprocal of their degrees or the logarithm of these values are called RA [S2] and AA [S3] respectively, and formulated as

$$S^{RA}(i,j) = \sum_{z \in (\Gamma(i) \cap \Gamma(j))} \frac{1}{|\Gamma(z)|}, \tag{S2}$$

$$S^{AA}(i,j) = \sum_{z \in (\Gamma(i) \cap \Gamma(j))} \frac{1}{log|\Gamma(z)|}. \tag{S3}$$

S.1.3 Cannistraci-Hebb (CH) network automata model

The CH index [S4] is a local similarity-based link prediction technique that not only considers the common neighbors of unconnected nodes, but also the number of the links between these neighbors called local-community to gauge the similarity of node pairs. It is computed as

$$S^{CH}(i,j) = \sum_{z \in (\Gamma(i) \cap \Gamma(j))} \frac{|\phi(z)|}{|\Gamma(z)|}, \tag{S4}$$

where $\phi(z) = (\Gamma(i) \cap \Gamma(j)) \cap \Gamma(z)$, i.e., it is the intersection of the common neighbors of node pair (i,j) with the neighbors of node z.

S.1.4 Fast probability Block Model (FBM)

The global similarity-based method FBM [S5] takes some subsets from all possible node partitions and computes the similarity between node pairs based on the group assignment of the nodes. In this procedure, all nodes are first randomly separated into two blocks, and the nodes in the maximum clique of every block are repeatedly selected and removed from

the block to form a group for the current partition. The remaining nodes not belonging to any group also set up another group together. Finally, the FBM similarity indices of node pairs are computed relying on their group assignment as

$$S^{FBM}(i,j) = \frac{1}{|P|}\sum_{p\in P} F(g_i, g_j), \tag{S5}$$

where P is the set of selected partitions, g_i is the group assignment of node i within partition p and function F is defined as

$$F(\alpha,\beta) = \begin{cases} \dfrac{r_\alpha}{2r_\alpha - l_\alpha}, \alpha = \beta, \\ \dfrac{l_{\alpha\beta}}{r_{\alpha\beta} + l_{\alpha\beta}}, \alpha \neq \beta, \end{cases} \tag{S6}$$

where l_α and r_α are the number of links and the maximum number of possible links between the nodes in group α, respectively. $l_{\alpha\beta}$ and $r_{\alpha\beta}$ are also the number of links between the nodes in groups α and β and the maximum number of possible links among them, respectively.

S.2 THE NORMALIZATION STEP IN LPM ALGORITHM

Fig. S1 highlights the importance of the normalization step in the general LPM algorithm for Laplacian-based link prediction presented in the main text. It compares the average performance of LPM link prediction methods versus the same perturbative Laplacian-based ones but using the unnormalized version of the graph Laplacian, distinguished from the proposed schemes by the suffix -UN. This figure reveals that the normalization step of LPM method is necessary for good performance.

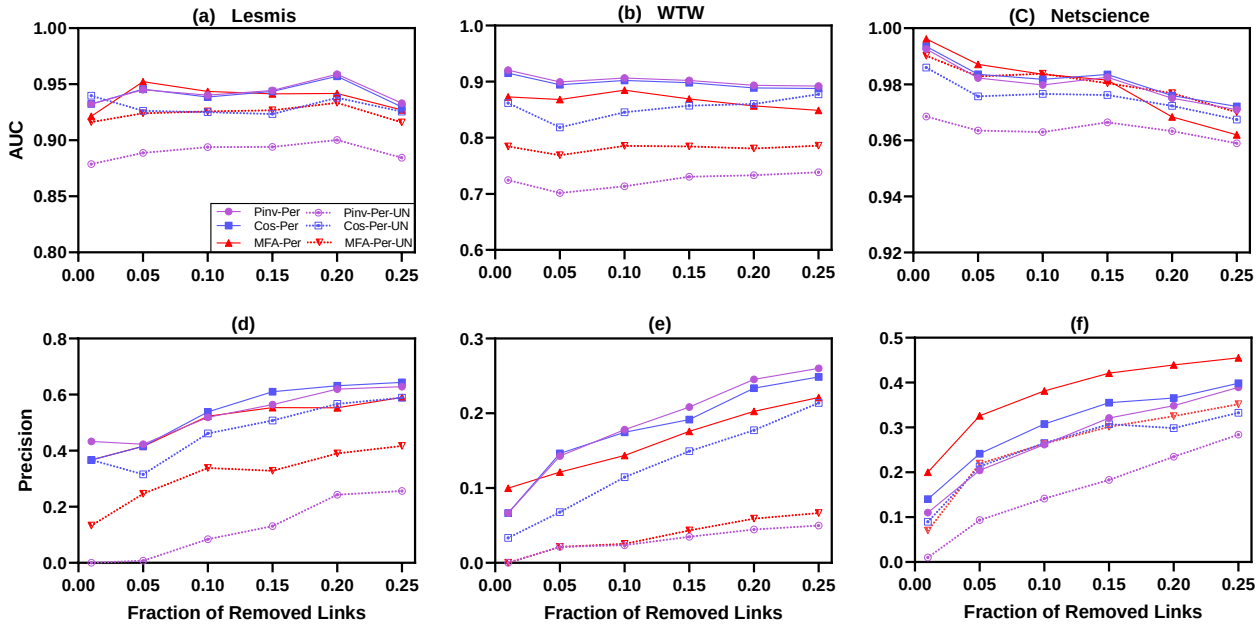

Fig. S1: The average AUC and precision of LPM link prediction methods applying the symmetric normalized Laplacian matrix versus their corresponding perturbative Laplacian-based counterparts employing the unnormalized version of the graph Laplacian, which are identified by the suffix **-UN**, as a function of the fraction of removed link q.

In LPM algorithm, after randomly division of an observed network into remaining (reduced) graph G_R and perturbation one ΔG, the Laplacian matrices of the resulting graphs $\mathbf{L}_R = \mathbf{D}_R - \mathbf{A}_R$ and $\Delta\mathbf{L} = \mathbf{D}_\Delta - \Delta\mathbf{A}$, where $\mathbf{A}_R$ and $\mathbf{D}_R$ are the adjacency and degree matrices of G_R and $\Delta\mathbf{A}$ and $\mathbf{D}_\Delta$ are those of ΔG, are normalized by their own degree matrices i.e., $\mathbf{D}_R$ and $\mathbf{D}_\Delta$ respectively (steps 5 and 6 of Algorithm 1 in the main text). Adding a small number of links to G_R changes a small amount of off-diagonal entries from 0 in $\boldsymbol{\mathcal{L}}_R$ to $-1/\sqrt{k_i^\Delta k_j^\Delta}$ in the resulting matrix $\tilde{\boldsymbol{\mathcal{L}}}_R$, where k_i^Δ stands for the degrees of nodes in the perturbation graph ΔG, which are typically low. As a result, these changes are of the order of the large majority of values in $\boldsymbol{\mathcal{L}}_R$, since most nodes are low degree, and would only affect a small number of entries.

In fact, this normalization protocol is a way to ensure that the role of hubs is not overemphasized. This can be understood by thinking what happens when both $\mathbf{L}_R$ and $\Delta\mathbf{L}$ are normalized by the same degree matrix, for instance,

$\mathbf{D}_R$. In that case, if the perturbation set contains links attached to hubs—this would be a very common situation since the observed links are randomly split in the remaining and perturbation sets—a number of off-diagonal entries would change from 0 in $\boldsymbol{\mathcal{L}}_R$ to $-1/\sqrt{k_i^R k_j^R}$ in the resulting matrix $\tilde{\boldsymbol{\mathcal{L}}}_R$, where k_i^R stands for the degrees of nodes in G_R. These degrees have typically large values for hubs, so that the change would be from 0 to a value that indicates the presence of a hub, a very strong signal in the context of the whole matrix even if the numerical value would be smaller as compared with the normalization provided by $\mathbf{D}_\Delta$. In addition, normalizing the graph Laplacian of ΔG by its own degrees yields a resulting $\Delta\boldsymbol{\mathcal{L}}$ which is congruent with the structure of this graph, while it is not the case when normalization is done by $\mathbf{D}_R$. Hence, in our framework, $\boldsymbol{\mathcal{L}}_R$ is perturbed using the normalized graph Laplacian associated with the structure of perturbation graph ΔG. In summary, a small perturbation means that a small number of entries in $\boldsymbol{\mathcal{L}}_R$ are affected by a change that is typical of low degree nodes, the most frequently found in complex networks, and not by a change associated to magnitudes typical of hubs, which are very rare events in networks and so in their matrix representation.

Moreover, in the normalized matrix $\Delta\boldsymbol{\mathcal{L}}$ achieved by $\mathbf{D}_R$ in which the diagonal and off-diagonal elements corresponding to the links in the perturbation set are k_i^Δ/k_i^R and $-1/\sqrt{k_i^R k_j^R}$ respectively, inhomogeneity of the degree in G_R make the resulting $\Delta\boldsymbol{\mathcal{L}}$ to be inhomogeneous. On the other hand, when $\Delta\boldsymbol{\mathcal{L}}$ is obtained using $\mathbf{D}_\Delta$, the diagonal elements associated with the links in the perturbation sets are always 1 and the off-diagonal elements are $-1/\sqrt{k_i^\Delta k_j^\Delta}$. Since the degree of nodes in ΔG, k_i^Δ, are low and close to each other, the elements of $\Delta\boldsymbol{\mathcal{L}}$ are homogeneous. A perturbation matrix with homogeneous elements ensures that the randomly selected perturbation links will play similar roles in the perturbation process.

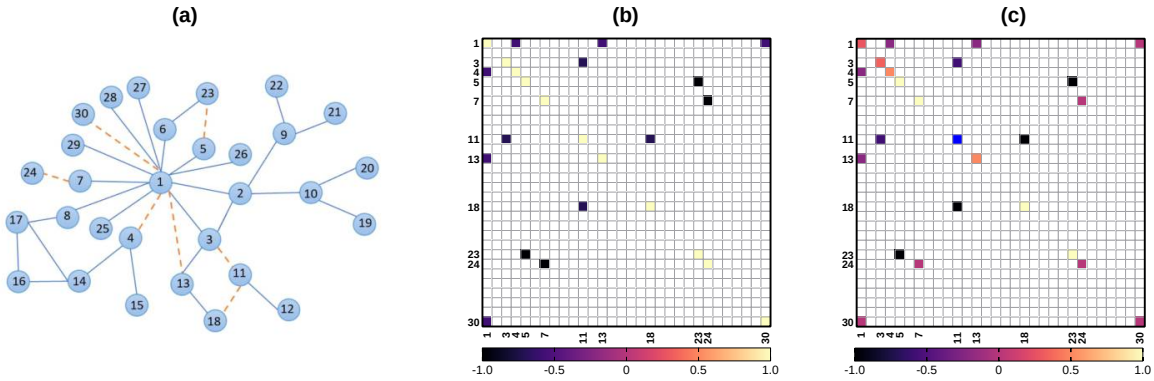

Fig. S2: (a) A small network of 30 nodes randomly divided into remaining graph G_R containing solid blue links and perturbation one ΔG including orange dashed links. (b) The normalized graph Laplacian of ΔG, $\Delta\boldsymbol{\mathcal{L}}$, when $\mathbf{D}_\Delta - \Delta\mathbf{A}$ is normalized using $\mathbf{D}_\Delta$. (c) The resulting $\Delta\boldsymbol{\mathcal{L}}$ when normalization is done using $\mathbf{D}_R$.

To better clarify this issue, we consider a simple example. Fig. S2(a) shows a network of 30 nodes with a hub in which a small fraction of the links shown by dashed lines has been randomly selected to constitute the perturbation graph ΔG. Panels (b) and (c) are the heatmap visualizations of the normalized Laplacian matrices corresponding to ΔG, $\Delta\boldsymbol{\mathcal{L}}$, when $\mathbf{D}_\Delta - \Delta\mathbf{A}$ is normalized using $\mathbf{D}_\Delta$ and $\mathbf{D}_R$, respectively.

Panel (b) highlights that the normalization provided by $\mathbf{D}_\Delta$ yields a $\Delta\boldsymbol{\mathcal{L}}$ with homogeneous elements corresponding to the links in the perturbation set. While in the case of using $\mathbf{D}_R$ in panel (c), the resulting $\Delta\boldsymbol{\mathcal{L}}$ is not homogeneous, and therefore the changes imposed by the perturbation links to $\boldsymbol{\mathcal{L}}_R$ are affected depending on whether they are attached to hubs or to low degree nodes. Moreover, in panel (c), the value of the diagonal element corresponding to the node number 11 (shown by blue as it is out of the range of the heatmap) is equal to 2 which exceeds the allowed range of values for the elements of normalized graph Laplacians.

Now, in the following, we evaluate how different normalization approaches affect the performance of LPM method. To this end, we use three alternative normalization schemes in steps 5 and 6 of Algorithm 1 in the main text to obtain the normalized Laplacian matrices for the remaining and perturbation graphs.

The first normalization method employs the degree matrix of the observed network $\mathbf{D}$ to normalize $\mathbf{D}_R - \mathbf{A}_R$ and $\mathbf{D}_\Delta - \Delta\mathbf{A}$. This being so, both $\mathbf{D}_R - \mathbf{A}_R$ and $\mathbf{D}_\Delta - \Delta\mathbf{A}$ are normalized by the same degree matrix. In this case, steps 5 and 6 in Algorithm 1 are substituted with

$$\boldsymbol{\mathcal{L}}_R = \mathbf{D}^{-1/2}(\mathbf{D}_R - \mathbf{A}_R)\mathbf{D}^{-1/2}, \tag{S7}$$

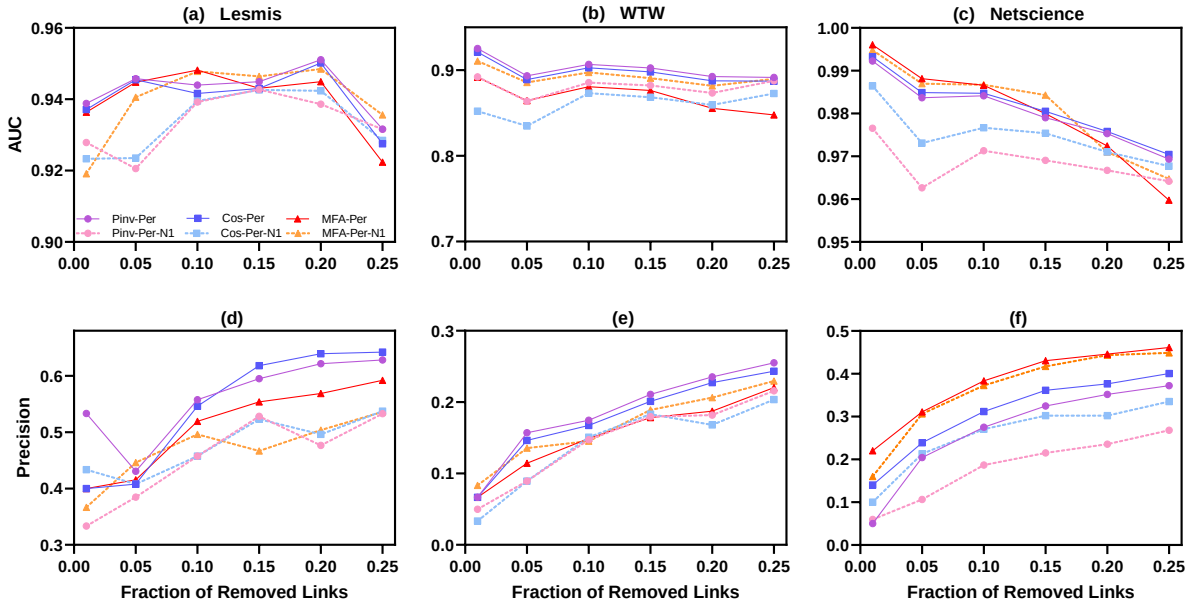

Fig. S3: The average AUC and precision of LPM link prediction methods and their counterparts employing Eqs. S7 and S8 to normalize remaining and perturbation graph Laplacians denoted by **-N1** suffix.

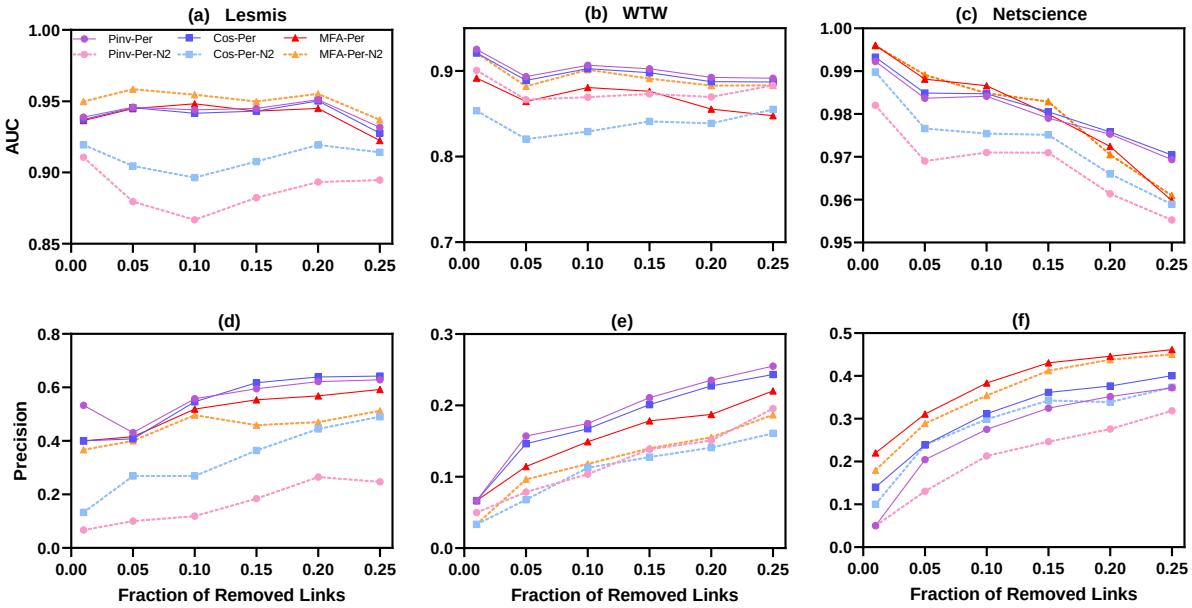

Fig. S4: The average AUC and precision of LPM link prediction methods and their counterparts employing Eqs. S9 and S10 to normalize remaining and perturbation graph Laplacians denoted by **-N2** suffix.

$$\Delta\boldsymbol{\mathcal{L}} = \mathbf{D}^{-1/2}(\mathbf{D}_{\Delta} - \Delta\mathbf{A})\mathbf{D}^{-1/2}. \tag{S8}$$

Fig. S3 compares the average AUC and precision of LPM link prediction methods with the corresponding perturbative Laplacian-based link prediction counterparts utilizing Eqs. S7 and S8. These link prediction schemes are denoted by the suffix **-N1**.

In the second normalization approach, $\mathbf{D}_R - \mathbf{A}_R$ and $\mathbf{D}_{\Delta} - \Delta\mathbf{A}$ are normalized by the same degree matrix corresponding to the remaining network $\mathbf{D}_R$. As a result, the normalization steps of LPM algorithm are implemented using

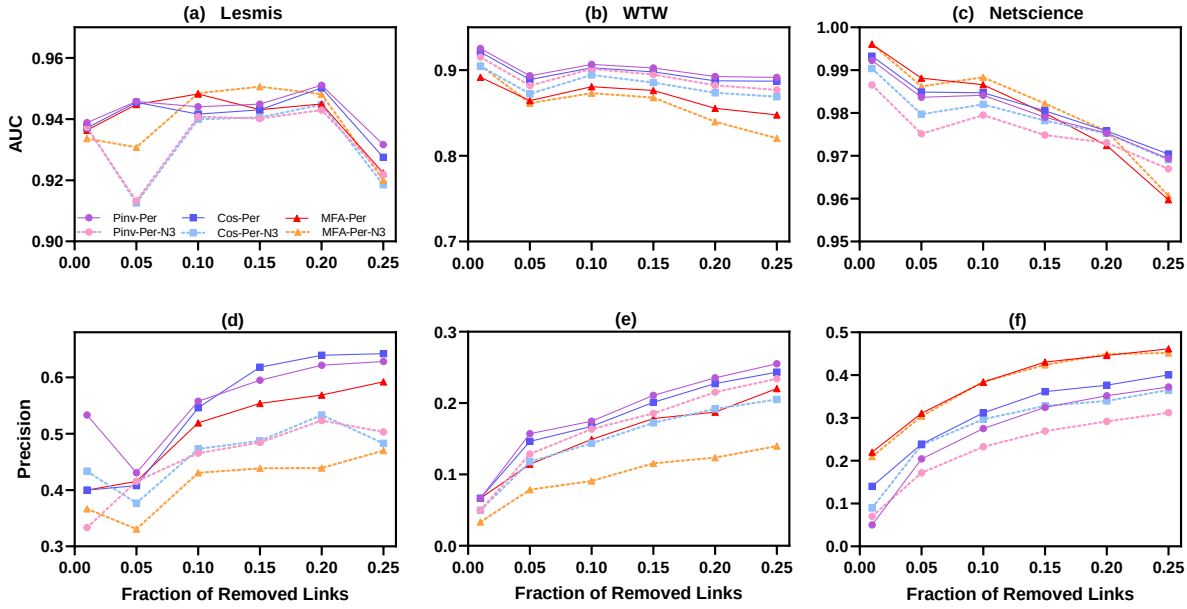

Fig. S5: The average AUC and precision of LPM link prediction methods and their counterparts employing Eqs. S11 and S12 to normalize remaining and perturbation graph Laplacians denoted by **-N3** suffix

$$\boldsymbol{\mathcal{L}}_R = \mathbf{D}_R^{-1/2}(\mathbf{D}_R - \mathbf{A}_R)\mathbf{D}_R^{-1/2}, \tag{S9}$$

$$\Delta\boldsymbol{\mathcal{L}} = \mathbf{D}_R^{-1/2}(\mathbf{D}_\Delta - \Delta\mathbf{A})\mathbf{D}_R^{-1/2}. \tag{S10}$$

Fig. S4 illustrates the performance of the proposed LPM link prediction methods in the main text with the corresponding perturbative Laplacian-based counterparts that employ the normalizations in Eqs. S9 and S10, distinguished by the suffix **-N2**.

Finally, in the third scheme, normalization is done such that the sum of the normalized Laplacians associated with remaining and perturbation graphs $\boldsymbol{\mathcal{L}}_R + \Delta\boldsymbol{\mathcal{L}}$ gives exactly the normalized Laplacian of observed networks $\boldsymbol{\mathcal{L}}$, i.e., $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}}_R + \Delta\boldsymbol{\mathcal{L}}$. Therefore, the normalization is done as

$$\boldsymbol{\mathcal{L}}_R = \mathbf{D}_R^{-1/2}(\mathbf{D}_R - \mathbf{A}_R)\mathbf{D}_R^{-1/2}, \tag{S11}$$

$$\Delta\boldsymbol{\mathcal{L}} = (\mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{A})\mathbf{D}^{-1/2}) - \mathbf{D}_R^{-1/2}(\mathbf{D}_R - \mathbf{A}_R)\mathbf{D}_R^{-1/2}, \tag{S12}$$

where $\mathbf{D}$ and $\mathbf{A}$ are the degree and adjacency matrices of the observed network. The results of applying this normalization approach to LPM method are shown in Fig. S5, where the suffix **-N3** identifies the new schemes.

On the whole, Figs. S3, S4 and S5 highlights that although the combination of the structural perturbation theory with each of the three normalization approaches discussed above leads to link prediction methods with superior performance than standard Laplacian-based schemes (the performance of the standard methods has been shown in Figs. 3 and 4 (b,e,h) in the main text), the normalization of the remaining and perturbation graphs with their own degree matrices that is used in the proposed LPM algorithm yields the best accuracy.

S.3 NETWORK MODELS

In this section, we outline the probabilistic network models employed in the main text to evaluate the performance of LPM link prediction methods. Furthermore, we investigate the dependency of LPM method performance on the structural properties of network models by varying model parameters values in the typical ranges observed in real networks.

S.3.1 Soft Configuration Model (sCM)

In sCM [S6], after assigning each node i an expected degree κ_i, the node pairs are connected with a probability given by

$$p_{ij} = \frac{\mu\kappa_i\kappa_j}{1+\mu\kappa_i\kappa_j}, \tag{S13}$$

where the free parameter μ controls the number of the links in the generated network. In a network of N nodes and average degree $\langle k\rangle$ when μ is set to $\frac{1}{\langle k\rangle N}$, the degree of each node i in the resulting network k_i is approximately equal to its expected one, i.e., $k_i \approx \kappa_i$.

S.3.2 $\mathbb{S}^1$ model

In the $\mathbb{S}^1$ model [S7], every node i is specified by latent variables (κ_i, θ_i), where κ_i is the node's hidden degree sampled from a probability density function $\rho(\kappa)$, and θ_i is its angular coordinate sampled uniformly at random from $[0, 2\pi]$ so as to map the nodes to a circle of radius $R = \frac{N}{2\pi}$. The node pairs are then connected with a probability given by

$$p_{ij} = \frac{1}{1+(\frac{R\Delta\theta_{ij}}{\mu\kappa_i\kappa_j})^\beta}, \tag{S14}$$

where $\Delta\theta_{ij} = \pi - |\pi - |\theta_i - \theta_j||$, as well as, β and μ are free parameters controlling the level of clustering and the average degree of the resulting network, respectively.

S.3.3 Degree-corrected Stochastic Block Model (dc-SBM)

In dc-SBM [S8], each node i is characterized by an expected degree κ_i and a group assignment g_i. In a network with N_b blocks, the number of the connections between group pairs is controlled by a symmetric $N_b \times N_b$ matrix of parameters ω computed as

$$\omega_{rs} = (1-\lambda)\omega_{rs}^{random} + \lambda\omega_{rs}^{planted}, \tag{S15}$$

where ω^{random} defines a fully random network with a specific degree sequence and without any group structure calculated as $\omega_{rs}^{random} = \kappa'_r\kappa'_s/2|E|$ in which κ'_r is the sum of the node expected degrees in group r and $|E|$ is the total number of links in the network. Moreover, $\omega^{planted}$ specifies a network with group structure, which in the simplest way, can be a diagonal matrix of parameters κ' representing a network with isolated communities. Furthermore, the free parameter λ makes a balance between these two types of structures. The number of links connecting two nodes of dc-SBM networks follows a Poisson distribution with mean of $\theta_i\theta_j\omega_{g_ig_j}$, where $\theta_i = \kappa_i/\kappa'_{g_i}$. As the probability of occurring multi-edges is low in the sparse-network limit, $\theta_i\theta_j\omega_{g_ig_j}$ can estimate the connection probability of node pairs as well. In this paper, to avoid values greater than 1, the connection probability of dc-SBM is computed as

$$p_{ij} = \frac{\theta_i\theta_j\omega_{g_ig_j}}{1+\theta_i\theta_j\omega_{g_ig_j}}. \tag{S16}$$

S.3.4 Experimental results for network models

In this section, we compare the performance of different LPM link prediction methods with their unperturbed Laplacian-based counterparts in synthetic networks of the sCM, $\mathbb{S}^1$ and dc-SBM ensembles. Fig. S6 shows the area under the receiver operating characteristic curve (AUC) and precision of LPM link prediction techniques and their unperturbed versions as a function of fraction of removed links for sCM networks when exponents γ of the power-law degree distribution is varied in the range of [2, 3]. Analogous results are shown in Figs. S7 and S8 for synthetic $\mathbb{S}^1$ networks with different values of parameter γ and various level of clustering β. Finally, Figs. S9 and S10 highlights the same results for synthetic networks generated by the dc-SBM model with increasing number of equiprobable blocks N_b and values of γ. In addition, Table. 1 compares the average AUC of LPM link prediction techniques with the presented baseline schemes in the network models.

S.4 Real-world networks

In this section, we investigate the performance of LPM and Laplacian-based link prediction methods in more real-world networks, as illustrated in Figs. S11 and S12. These figures show that LPM link prediction methods (except ACT-Per) surpass their standard Laplacian-based link prediction counterparts in almost all real-world networks, and they beat SPM

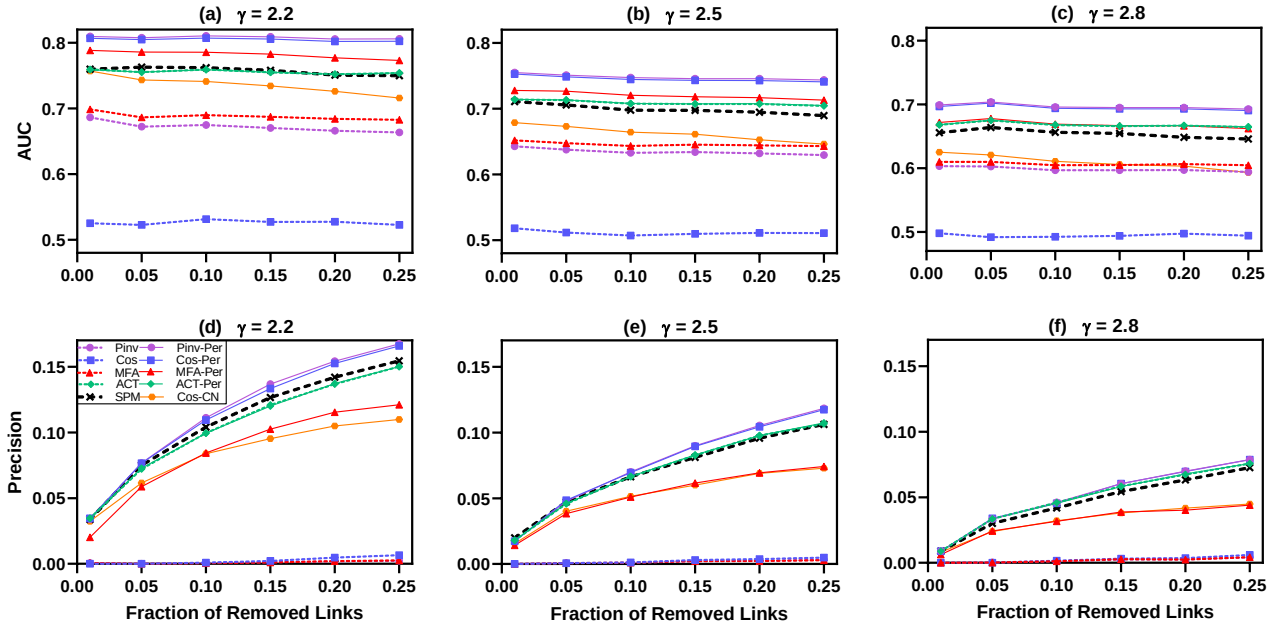

Fig. S6: The average AUC and precision of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q for synthetic sCM networks. For each value of the power-law degree distribution exponent γ increased from left to right, we have generated 10 different networks with $N = 300$ nodes and the average node degree of $\langle k \rangle = 10$. For every network and each value of q, we have also constructed 10 disjoint training and probe sets. In all the plots, the black dashed curves represent the average performance of SPM.

TABLE 1: Performance of link prediction methods measured by AUC for synthetic and real networks. The probe sets contain a $q = 0.1$ fraction of the links in the complete networks and the presented results are the average of 10 independent runs. In all the netwok models, we have set $\gamma = 2.2$. In $\mathbb{S}^1$, we have fixed $\beta = 1.5$, and in dc-SBM, $N_b = 14$ and $\lambda = 0.5$. For LPM link prediction methods, AUC values of their unperturbed counterparts are reported in parentheses.

Networks	CN	RA	AA	CH	FBM	SPM	Pinv-Per	Cos-Per	MFA-Per	Cos-CN
sCM	0.731	0.732	0.736	0.655	0.787	0.762	**0.811** (0.675)	0.807 (0.531)	0.786 (0.690)	0.741
$\mathbb{S}^1$	0.855	0.870	0.869	0.768	0.879	0.858	0.882 (0.846)	0.883 (0.808)	**0.899** (0.858)	0.873
dc-SBM	0.727	0.735	0.737	0.622	0.774	0.727	**0.784** (0.712)	0.781 (0.607)	0.781 (0.721)	0.739
Iceland	0.862	**0.885**	**0.885**	0.601	0.879	0.807	**0.885** (0.777)	0.881 (0.772)	0.850 (0.846)	0.880
Word Adjacency	0.677	0.676	0.678	0.570	0.726	0.731	**0.763** (0.671)	0.758 (0.608)	0.720 (0.686)	0.681
Haggle	0.960	0.961	0.961	0.943	0.965	0.954	**0.974** (0.890)	**0.974** (0.905)	0.957 (0.926)	0.968
Tortoise	0.886	0.887	0.887	0.722	0.892	0.870	0.916 (0.890)	**0.917** (0.910)	0.897 (0.888)	0.876
FB-Food	0.911	0.915	0.914	0.769	0.935	0.939	0.958 (0.916)	**0.960** (0.955)	0.947 (0.945)	0.914

in some of them. Table. 1 also compares the performance of LPM with the state-of-the-art link prediction schemes measured by AUC in real-world network. A brief description of the networks used in these experiments are provided in the following section.

In the cases in which LPM outperforms SPM, the improvement has its roots in structural information encoded in the graph Laplacian that emerges when the inverse or pseudo-inverse is applied. To show this, we have used the elements of the perturbed graph Laplacian $\tilde{\boldsymbol{\mathcal{L}}}$, without applying the inverse or pseudo-inverse operator, as the source of similarity indices for link prediction. Since Laplacian matrices characterize the links in graphs by negative values, we decided to use the absolute values of the elements in the perturbed Laplacian matrix corresponding to unconnected node pairs as the similarity scores (experimental results, not shown here, reveal that using the real negative values of $\tilde{\boldsymbol{\mathcal{L}}}$ elements gives very low accuracy). We have implemented two versions of this scheme based on the normalized and the unnormalized graph Laplacians called $|NLAP|$ and $|UnLAP|$, respectively.

Fig. S13 compares the performance of SPM perturbing the adjacency matrix with $|NLAP|$ and $|UnLAP|$ which similarly perturb the normalized and unnormalized graph Laplacians in some real-world networks. Although, the three methods

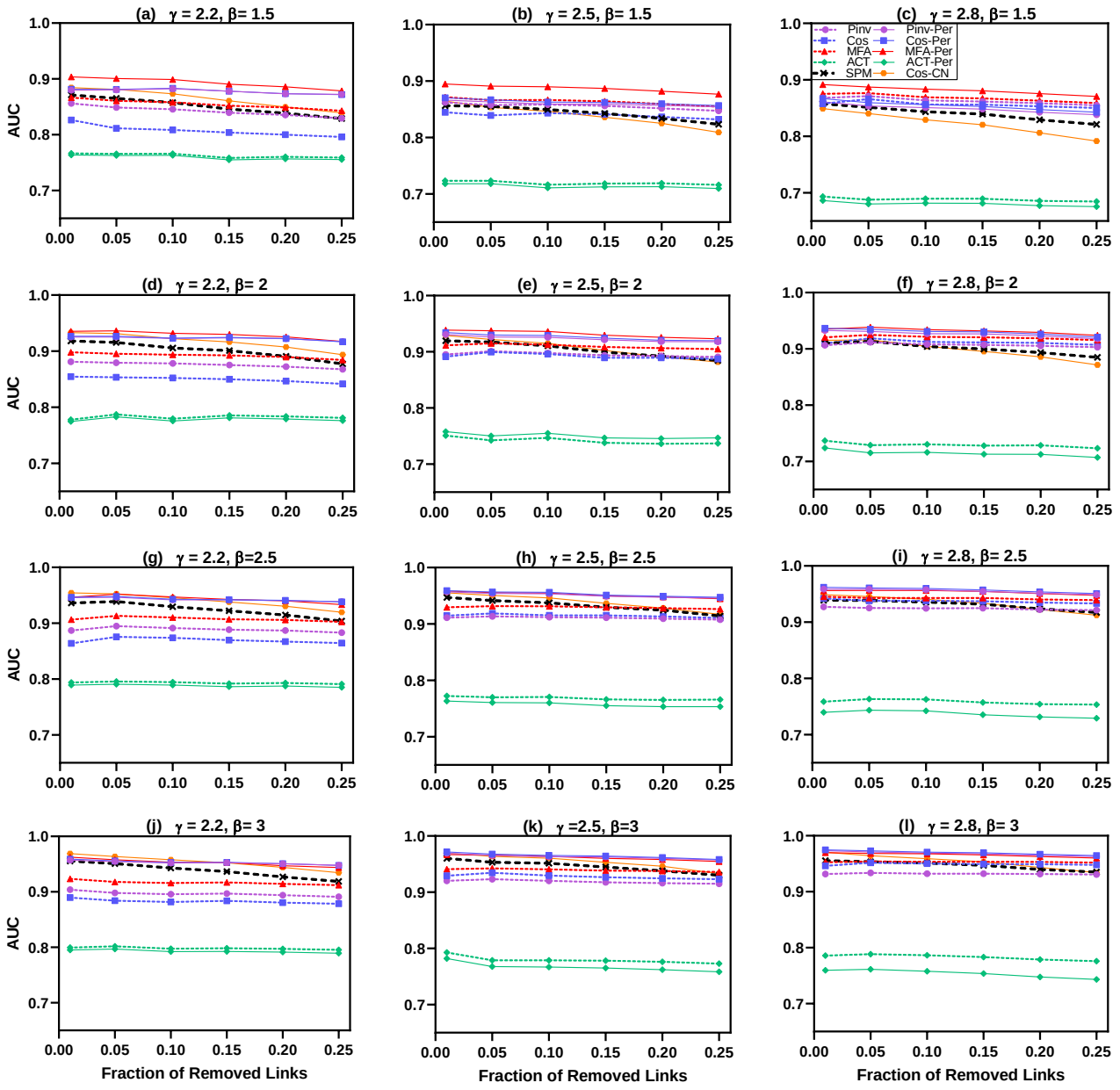

Fig. S7: The average AUC of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q for synthetic $\mathbb{S}^1$ networks. The number of nodes in the networks is $N = 300$ and the average node degree is $\langle k \rangle = 10$. Parameter β controlling the level of clustering is increased from up to down and the power-law degree distribution exponent γ is incremented from left to right. For every ensemble characterized by specific γ and β, we have generated 10 different networks, and for every network and each value of q, we have also constructed 10 disjoint training and probe sets. In all the plots, the black dashed curves represent the average AUC of SPM.

employ the same source of structural information for link prediction, $|NLAP|$ and $|UnLAP|$ usually exhibit inferior accuracy than SPM. It means that when Laplacian matrix is considered as the structural representation of a network, using merely the absolute values of perturbed graph Laplacian elements associated with potential links does not lead to effective link prediction methods. However, LPM link prediction techniques that extract more structural information encoded in the pseudo-inverse of the perturbed graph Laplacian mapping nodes to an appropriate feature space and computing node vectors inner products or the cosine of the angle between them, can improve the performance of $|NLAP|$ and $|UnLAP|$ as also depicted in Fig. S13. In other words, in spite of using the same perturbation procedure in $|NLAP|$, $|UnLAP|$ and LPM method, the improvement only happens when extra information from the graph Laplacian is exploited.

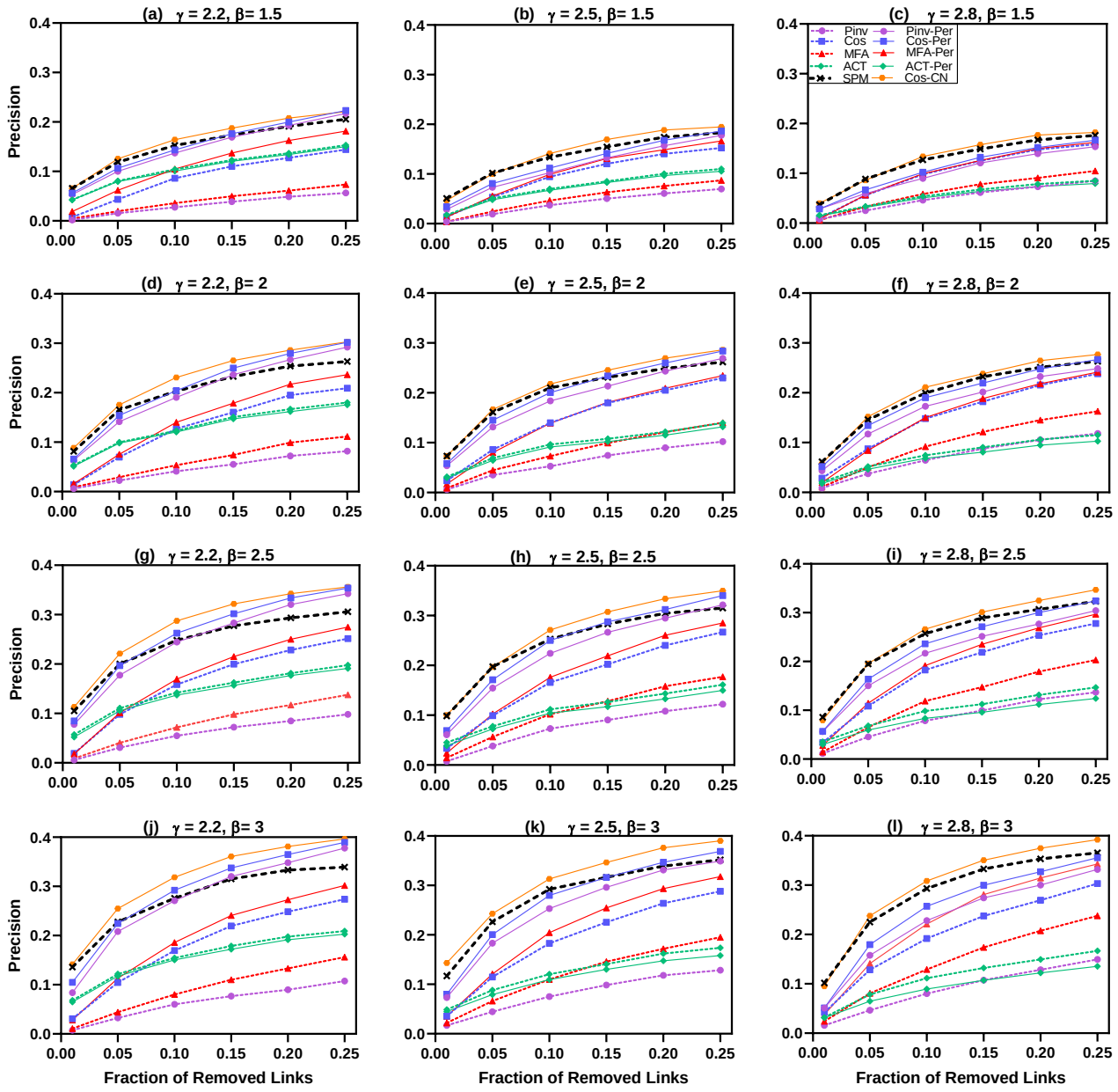

Fig. S8: The average precision of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q for synthetic $\mathbb{S}^1$ networks. The number of nodes in the networks is $N = 300$ and the average node degree is $\langle k \rangle = 10$. Parameter β controlling the level of clustering is increased from up to down and the power-law degree distribution exponent γ is incremented from left to right. For every ensemble characterized by specific γ and β, we have generated 10 different networks, and for every network and each value of q, we have also constructed 10 disjoint training and probe sets. In all the plots, the black dashed curves represent the average precision of SPM.

S.4.1 Data description

Here, the performance of LPM link prediction methods has been assessed in a set of real-world networks from disparate area including male homosexual relationships in Iceland [S9] (Iceland), adjacent common adjectives and nouns in the novel David Copperfield [S10] (Word Adjacency), people's communications masured by wireless devices [S11] (Haggle), neural network in a type of worm called Caenorhabditis elegans or C.elegans [S12] (Neural), face to face contacts between the participants of the exhibition INFECTIOUS: STAY AWAY 2009 [S13] (Infectious), metabolic network of C.elegans [S14] (Metabolic), social network of dessert tortoises [S15] (Tortoise), mutually liked blue verified Facebook pages in food category [S16] (FB-Food) and hyperlinks between political weblogs [S17] (Polblogs).

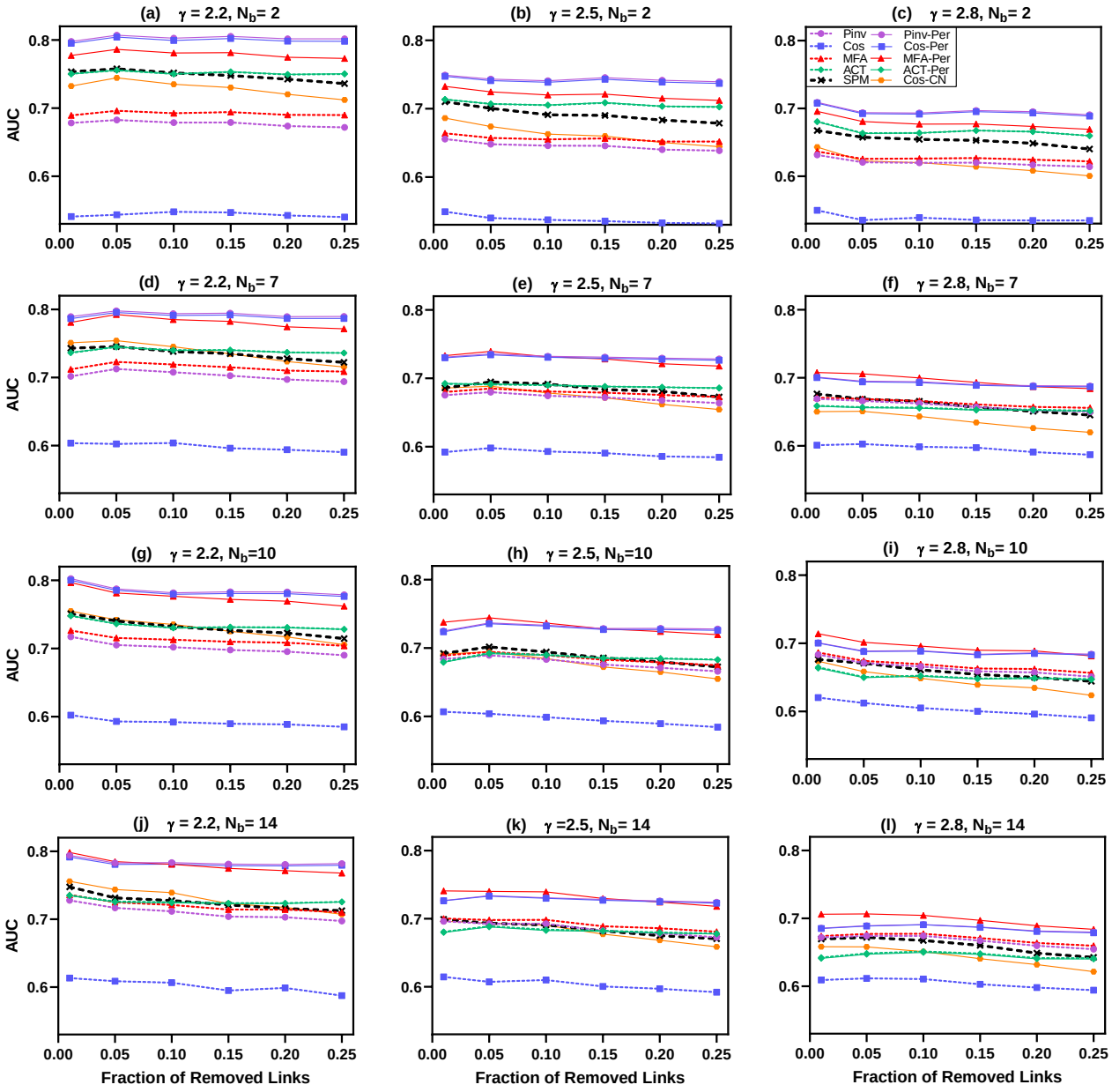

Fig. S9: The average AUC of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q for synthetic dc-SBM networks. The number of nodes in the networks is $N = 300$, the average node degree is $\langle k \rangle = 10$ and parameter λ making a balance between random and group structures has been fixed to $\lambda = 0.5$. The number of equiprobable blocks N_b is increased from up to down and the power-law degree distribution exponent γ is incremented from left to right. For every ensemble characterized by specific γ and N_b, we have generated 10 different networks, and for every network and each value of q, we have also constructed 10 disjoint training and probe sets. In all the plots, the black dashed curves represent the average AUC of SPM.

Table. 2 also reports the basic topological characteristics of all the real-world networks used in this paper. In all the experiments, undirected unweighed versions of the networks are used by ignoring self loops, the directions and weights of links, and replacing multi-edges with a single link in case of existence. In addition, for the networks with more than one component the giant connected component is considered as the complete network. In this case, the number of nodes and links in the orginal network are shown in the parenthesis in Table. 2. These networks can be downloaded from the Koblenz Network Collection [S18] and the Network Data Repository [S19].

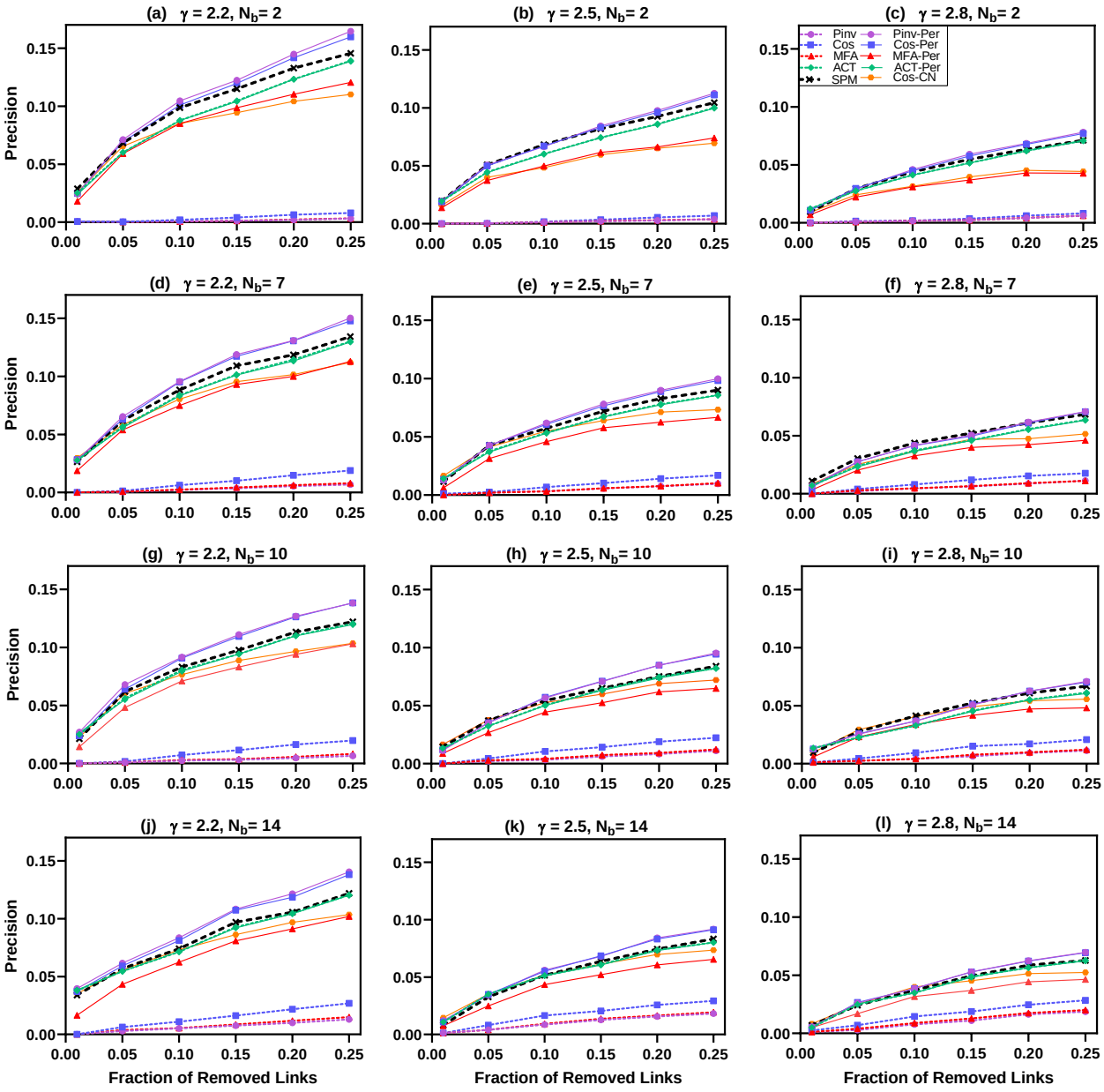

Fig. S10: The average precision of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q for synthetic dc-SBM network. The number of nodes in the networks is $N = 300$, the average node degree is $\langle k \rangle = 10$ and parameter λ making a balance between random and group structures has been fixed to $\lambda = 0.5$. The number of equiprobable blocks N_b is increased from up to down and the power-law degree distribution exponent γ is incremented from left to right. For every ensemble characterized by specific γ and N_b, we have generated 10 different networks, and for every network and each value of q, we have also constructed 10 disjoint training and probe sets. In all the plots, the black dashed curves represent the average precision of SPM.

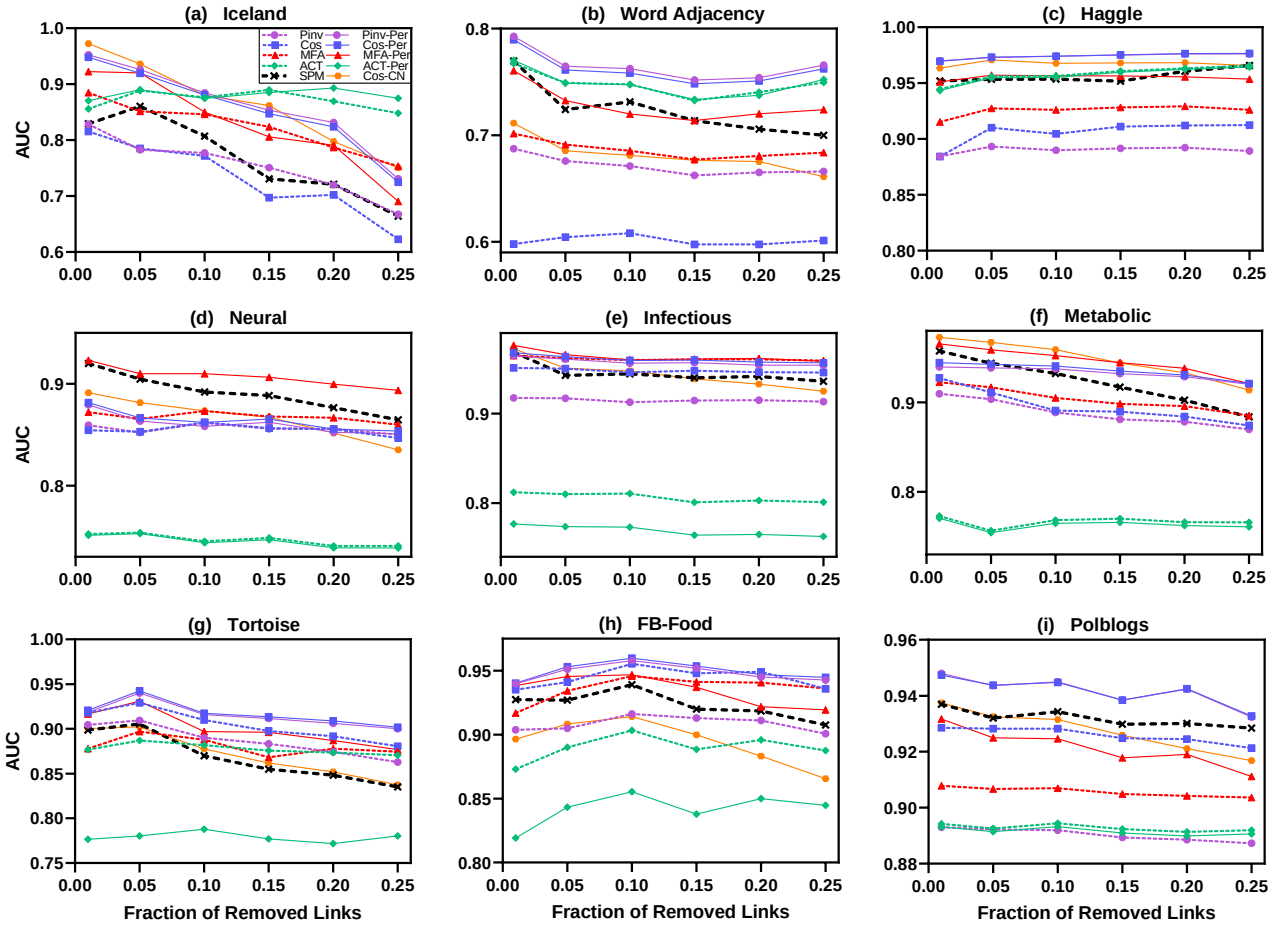

Fig. S11: AUC values of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q. For each real network and each value of q, we have generated 10 different training and probe sets. In all the plots, the black dashed curves represent the average AUC of SPM.

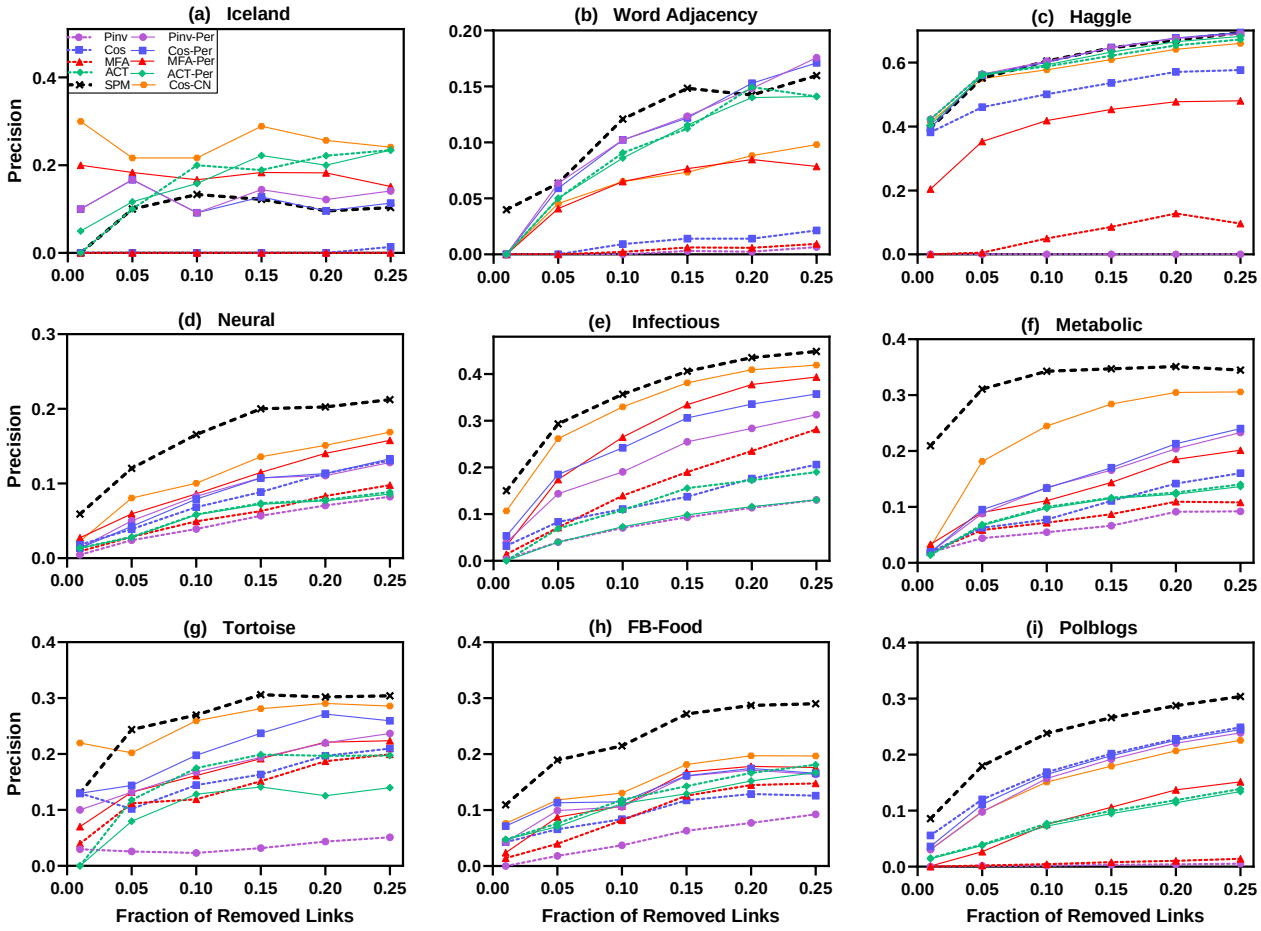

Fig. S12: Precision values of LPM and the corresponding unperturbed Laplacian-based link prediction methods as a function of the fraction of removed links q. For each real network and each value of q, we have generated 10 different training and probe sets. In all the plots, the black dashed curves represent the average precision of SPM.

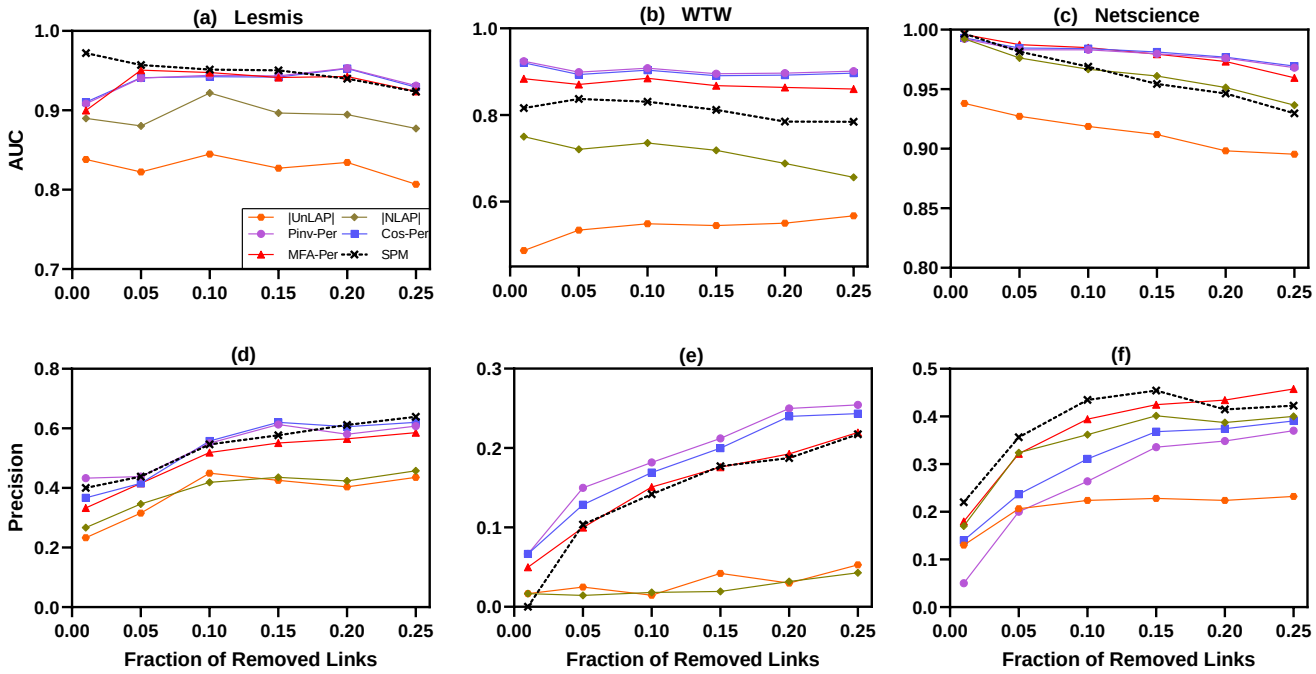

Fig. S13: The average AUC and precision of LPM link prediction methods versus $|UnLAP|$ and $|NLAP|$ schemes employing the elements of perturbed unnormalized and normalized graph Laplacian (without applying the pseudo-inverse operator) as similarity scores of node pairs, respectively. The black dashed curves represent the average performance of SPM.

TABLE 2: The basic topological charachteristics of the real-world networks. N: number of nodes, $|E|$: number of links, $\langle k\rangle$: average degree, C: clustering coefficient [S12], H: degree heterogeniety measured as $H = \frac{\langle k^2\rangle}{\langle k\rangle^2}$, r: assortative coefficient [S20] and $\langle d\rangle$: average shortest distance. For the networks with more than one component, N and $|E|$ of the original networks are shown in the parenthesis.

Networks	N	$\|E\|$	$\langle k\rangle$	C	H	r	$\langle d\rangle$	Ref
Karate	34	78	4.588	0.571	1.693	-0.476	2.408	[S21]
Lesmis	77	254	6.597	0.573	1.827	-0.165	2.641	[S22]
Polbooks	105	441	8.4	0.488	1.421	-0.128	3.079	[S23]
ACM2009	113	2196	38.867	0.535	1.223	-0.123	1.656	[S13]
WTW	189	550	5.82	0.573	4.281	-0.299	2.491	[S24]
CongressVote	219	521	4.758	0.255	2.365	-0.340	3.315	[S25]
USAir	332	2126	12.807	0.625	3.464	-0.208	2.738	[S26]
Netscience	379 (1589)	914 (2742)	4.823	0.741	1.663	-0.082	6.042	[S10]
Email	1133	5451	9.622	0.220	1.942	0.078	3.606	[S27]
Iceland	75	114	3.04	0.287	2.75	-0.401	3.2	[S9]
Word Adjacency	112	425	7.59	0.173	1.815	-0.13	2.536	[S10]
Haggle	274	2124	15.504	0.633	3.656	-0.474	2.424	[S11]
Neural	297	2148	14.465	0.292	1.801	-0.163	2.455	[S12]
Infectious	410	2765	13.488	0.456	1.388	0.226	3.631	[S13]
Metabolic	453	2025	8.940	0.647	4.485	-0.226	2.664	[S14]
Tortoise	496 (787)	984 (1713)	3.968	0.336	1.612	0.345	7.933	[S15]
FB-Food	620	2091	6.745	0.331	2.952	-0.032	5.089	[S16]
Polblogs	1222	16714	27.355	0.32	2.971	-0.221	2.738	[S17]

REFERENCES

[S1] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, p. 025102, 2001.
[S2] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.
[S3] L. A. Adamic and E. Adar, "Friends and neighbors on the Web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.
[S4] A. Muscoloni, U. Michieli, and C. V. Cannistraci, "Local-ring network automata and the impact of hyperbolic geometry in complex network link-prediction," *arXiv preprint arXiv:1707.09496*, 2017.
[S5] Z. Liu, J.-L. He, K. Kapoor, and J. Srivastava, "Correlations between community structure and link formation in complex networks," *PLoS ONE*, vol. 8, no. 9, p. e72908, 2013.
[S6] J. Park and M. E. J. Newman, "Origin of degree correlations in the Internet and other networks," *Physical Review E*, vol. 68, no. 2, p. 026112, 2003.
[S7] M. Á. Serrano, D. Krioukov, and M. Boguñá, "Self-similarity of complex networks and hidden metric spaces," *Physical Review Letters*, vol. 100, no. 7, p. 078701, 2008.
[S8] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Physical Review E*, vol. 83, no. 1, p. 016107, 2011.
[S9] S. Haraldsdottir, S. Gupta, and R. M. Anderson, "Preliminary studies of sexual networks in a male homosexual community in Iceland," *Journal of Acquired Immune Deficiency Syndromes*, vol. 5, no. 4, pp. 374–381, 1992.
[S10] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, p. 036104, 2006.
[S11] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 606–620, 2007.
[S12] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[S13] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, "What's in a crowd? Analysis of face-to-face behavioral networks," *Journal of Theoretical Biology*, vol. 271, no. 1, pp. 166–180, 2011.
[S14] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, no. 2, p. 027104, 2005.
[S15] P. Sah, K. E. Nussear, T. C. Esque, C. M. Aiello, P. J. Hudson, and S. Bansal, "Inferring social structure and its drivers from refuge use in the desert tortoise, a relatively solitary species," *Behavioral Ecology and Sociobiology*, vol. 70, no. 8, pp. 1277–1289, 2016.
[S16] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, "GEMSEC: graph embedding with self clustering," in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2019, pp. 65–72.
[S17] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: divided they blog," in *Proceedings of the 3rd International Workshop on Link Discovery*. Chicago, IL, USA: Association for Computing Machinery, 2005, pp. 36–43.
[S18] J. Kunegis, "KONECT: The Koblenz network collection," in *Proceedings of the 22nd International Conference on World Wide Web*. Rio de Janeiro, Brazil: Association for Computing Machinery, 2013, pp. 1343–1350.
[S19] R. A. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, Austin, Texas, USA, 2015.
[S20] M. E. J. Newman, "Assortative mixing in networks," *Physical Review Letters*, vol. 89, no. 20, p. 208701, 2002.
[S21] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.
[S22] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*. Reading, MA, USA: Addison-Wesley, 1993.
[S23] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
[S24] G. García-Pérez, M. Boguñá, A. Allard, and M. Á. Serrano, "The hidden hyperbolic geometry of international trade: World Trade Atlas 1870–2013," *Scientific Reports*, vol. 6, no. 1, p. 33441, 2016.
[S25] M. Thomas, B. Pang, and L. Lee, "Get out the vote: Determining support or opposition from Congressional floor-debate transcripts," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia: Association for Computational Linguistics, 2006, pp. 327–335.
[S26] V. Batagelj and A. Mrvar, "Pajek datasets," 2006. [Online]. Available: http://vlado.fmf.uni-lj.si/pub/networks/data/
[S27] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical Review E*, vol. 68, no. 6, p. 065103, 2003.