

The maximum capability of a topological feature in link prediction

Yijun Ran,¹ Xiao-Ke Xu,² and Tao Jia^{1,*}

¹*College of Computer and Information Science,*

Southwest University, Chongqing, 400715, P. R. China

²*College of Information and Communication Engineering,*

Dalian Minzu University, Dalian 116600, P. R. China

Abstract

Link prediction aims to predict links of a network that are not directly visible, with profound applications in biological and social systems. Despite intensive utilization of the topological feature in this task, it is unclear to what extent a particular feature can be leveraged to infer missing links. Here, we show that the maximum capability of a topological feature follows a simple mathematical expression, which is independent of how an index gauges the feature. Hence, a family of indexes associated with one topological feature shares the same performance limit. A feature's capability is lifted in the supervised prediction, which in general gives rise to better results compared with unsupervised prediction. The universality of the pattern uncovered is empirically verified by 550 structurally diverse networks, which can be applied to feature selection and the analysis of network characteristics associated with a topological feature in link prediction.

* tjia@swu.edu.cn

Complex systems can be described by networks, in which nodes are the components of the system and links are the interactions between the components [1, 2]. Link prediction is a task to infer missing connections that should exist but are not directly visible due to the incomplete information of the system [3–6], which has wide applications in predicting molecular interactions [7, 8], drug targets [9, 10], protein-protein interactions [11, 12], recommendations on online social platforms [13, 14] and online shopping [15, 16]. Because the topology information is usually directly available for a given network, significant effort has been devoted to utilizing topological features to predict missing links. Indeed, despite recent developments in computational tools [17, 18], especially the network embedding by deep learning techniques [19–21], the topological feature is still widely used in link prediction due to its simplicity, interpretability, and overall good performance [22–29].

As the complexity underlying the connection of a network differs, the prediction of missing links can be easier in some networks than in others. Recent studies start to explore the link predictability of a network [30–32], corresponding to the extreme performance that any prediction method can ever reach. However, if we focus on a particular method based on a particular topological feature, its performance limit is not well understood. The extent to which a prediction method can be further improved remains unknown, nor there is any theoretical guidance on the condition that a new feature should be considered for a more accurate prediction.

Here, we ask a question different from link predictability, yet equally important: to what extent could a topological feature be leveraged in link prediction? Seemingly, this question can only be answered on a case-by-case basis, as a myriad of factors could play a role. One topological feature can be quantified through multiple indexes, giving rise to different index values. The obtained index value can be directly used for prediction, corresponding to an unsupervised heuristic approach [33–37], or further processed by a machine learning algorithm for a supervised prediction [38–42]. The prediction outcome can be gauged by different measures such as AUC and precision [30, 31, 39–45]. Hence, a unified answer is not expected.

Nevertheless, we find that the link prediction performance follows simple patterns. The maximum capability of a topological feature has a precise mathematical expression, which depends on the extent to which the feature is presented in missing and nonexistent links, but is independent of the way that an index quantifies the feature. Hence, a family of indexes

based on the same topological feature shares the same upper bound of prediction accuracy. The potential of all other indexes can be readily estimated through one single measurement. We also demonstrate that the supervised prediction in principle gives a more accurate result compared with the unsupervised prediction. But this improvement is not merely by pushing the performance to the same upper bound. Instead, the maximum capability of the topological feature is lifted by utilizing the supervised method, which can be mathematically quantified. Therefore, the advances by applying machine learning tools can be estimated. The finding can be applied to optimize the feature and the method selection, and also advances our understanding of network characteristics associated with the utilization of a topological feature in link prediction. Taking the common neighbor feature as an example, we quantitatively show how the motif of open and closed triangles determines the prediction outcome, which can not be fully explained by a network’s clustering coefficient. Our work benefits from a recently announced large corpus of 550 structurally diverse networks [39, 46], allowing us to empirically verify the universality of the pattern uncovered.

Results

Problem definition. We select 18 indexes commonly used in link prediction, ranging from the traditional common neighbor index (CN) in social networks [33] to the recently proposed paths of length three index (L3) for protein-protein interaction networks [12]. According to the associated topological feature, these indexes can be classified into 4 families: common neighbor [7, 33, 47–49], path [50–54], heterogeneity [55], and path of length three [12, 56]. A given feature can be quantified differently by indexes in the same family. For example, the feature common neighbor can be expressed as the number of common neighbors [33] or the percentage of the neighborhood overlap [14]. The path feature can be quantified by different combinations of paths [26, 56, 57]. We list the 18 indexes of the 4 features in Table. I with detailed descriptions presented in Supplementary Information S1.

Essentially, the link prediction is to assign a score S_{ab} to two nodes a and b , which is proportional to the chance that nodes a and b should be truly connected. An index value for a topological feature can be utilized in two ways. One is to input the value to a machine learning based classifier, which finds a mapping function $y = f(x)$ to transfer the index value x to the score value y . The other is to directly use the index value as the score, corresponding to a simple mapping function $y = x$. The two approaches, with the same input, differ only on the choice of the mapping function. They are named differently

depending on the taxonomy applied [22–24, 26, 27, 29]. The former is sometimes called algorithm-based or learning-based approach, whereas the latter is named feature-based, similarity-based, or heuristic approach. To unify the name, we call the former *supervised* approach and the latter *unsupervised* approach in this paper.

We adopt the most common settings of the link prediction problem [22, 27, 29, 34]. Assume an undirected simple network composed of N nodes and L links, in which a node can not connect to itself (no self-loops) nor share more than one link with another node (no repeated links). Because missing links are actually unknown, a prediction can hardly be tested. To make the problem technically testable, a small portion of existing links are removed from the original network. They are considered as missing links to be inferred, which form the positive set L^P . As the control group of L^P , the negative set L^N is constructed by randomly picking node pairs that are not connected in the original network, forming a set of nonexistent links. L^N and L^P are usually given the same size to balance the sample [58, 59]. In this work, we randomly remove 20% of L links from the network and keep the rest 80% of links, from which the feature index value is quantified. For the 20% removed links, half of them (10% of L links) are selected as the testing set, and the other half is particularly used as the training set in the supervised approach. More details can be found in Supplementary Information S2.

In general, link prediction is treated as a binary classification problem. The prediction performance is gauged by the extent to which L^P outscores L^N . One measure most commonly applied is AUC [39–43]. The AUC can be calculated by a sampling method [34, 48, 53, 55]. In each comparison, we randomly pick one node pair from L^P and one from L^N , and compare their scores. If out of n comparisons, there are n' times that samples from L^P have a higher score than that from L^N , and n'' times that they have the same score, the AUC can be calculated as

$$\text{AUC} = \frac{n' + 0.5n''}{n}. \quad (1)$$

The performance can also be measured by precision [30, 31, 40, 44, 45], which is to quantify the percentage of true missing links in the top- k node pairs with the highest score. If we select L_k node pairs with the highest score, in which L_r pairs are included in L^P , the precision is

$$\text{Precision} = \frac{L_r}{L_k}. \quad (2)$$

An example of the link prediction task and the quantification by AUC and precision is illustrated in Supplementary Figure S1.

The framework proposed in this study applies to prediction capability quantified by both AUC and precision. However, to avoid switching between different measures and to make the paper's flow more consistent, we choose to present results based on AUC in the main text. Results related to precision are discussed in Supplementary Information S3.

The capability of a feature in the unsupervised prediction. The index value is directly used to predict missing links in the unsupervised approach. Hence, how the index values are distributed in L^P and L^N determines the prediction performance. Different indexes gauge the same feature differently. But since they are proposed to quantify the feature, they should follow one rule in common: entities that do not hold the feature have the same and the lowest value. Without loss of generality, these entities are often assigned by the value 0 (see further discussion in Supplementary Information S4). For instance, a pair of nodes without any overlap of neighborhoods has the value 0 for any indexes based on the common neighbor feature. Likewise, all indexes using network distance assign the value 0 to a pair of nodes not connected by a path. Denote L_1 and L_2 by the subset of node pairs that hold the topological feature in L^P and L^N (index value greater than 0), respectively (Fig. 1a). Consequently, $\bar{L}_1 = L^P \setminus L_1$ and $\bar{L}_2 = L^N \setminus L_2$ are the subset of node pairs with index value 0. The prediction performance mainly relies on the values in L_1 and L_2 , as node pairs in \bar{L}_1 and \bar{L}_2 all have the same value 0 and are indistinguishable by their index values.

Let us first consider the worst scenario, when the index values are assigned such that the highest value in L_1 is less than the lowest in L_2 (Fig. 1b). A positive sample outscores a negative sample only when one is from L_1 and the other is from \bar{L}_2 . A positive sample and a negative sample have the same score only when they are from $\bar{L}_1 \cup \bar{L}_2$. Denoting $p_1 = |L_1|/|L^P|$ and $p_2 = |L_2|/|L^N|$, the AUC in the worst scenario can be calculated as

$$\text{AUC}_{\text{lower}} = \frac{n'}{n} + \frac{1}{2} \frac{n''}{n} = p_1(1 - p_2) + \frac{1}{2}(1 - p_1)(1 - p_2) = \frac{1}{2} + \frac{p_1 - p_2 - p_1 p_2}{2}. \quad (3)$$

On the contrary, the best scenario is when the lowest value in L_1 is greater than the highest in L_2 (Fig. 1c). Node pairs in L_1 outscore all negative samples, which gives the AUC upper bound

$$\text{AUC}_{\text{upper}} = \frac{n'}{n} + \frac{1}{2} \frac{n''}{n} = p_1 + \frac{1}{2}(1 - p_1)(1 - p_2) = \frac{1}{2} + \frac{p_1 - p_2 + p_1 p_2}{2}. \quad (4)$$

For general cases, the index values distributed in $\overline{L_1}$ and $\overline{L_2}$ have overlaps. Hence, the performance lies between AUC_{lower} and AUC_{upper} .

Eq. (3) and Eq. (4) suggest that the performance of the unsupervised link prediction depends on how well an index can rank positive samples ahead of the negative samples. Two factors set the bound of this process. First, there are $(1 - p_1) + (1 - p_2)$ samples with the same index value that can not be effectively ranked. Second, for the rest $p_1 + p_2$ samples, once the set L_1 and L_2 are well separated and properly ranked by their values, no further improvement can be achieved. Note that we start with the performance of an index, but Eq. (3) and Eq. (4) only depends on p_1 and p_2 . The performance limit of an index only relies on the extent to which a topological feature is presented in missing and nonexistent links, not on how the index quantitatively gauges the feature. Different indexes in the same family can give rise to different prediction results, with different distances to the upper bound set by Eq. (4). But they all fall within the same range determined by the properties of the topological feature (Supplementary Figures S2). Hence, Eq. (4) not only depicts the performance limit of one single index but also effectively gives the maximum capability of a topological feature in unsupervised link prediction that can be ever reached by any index associated with it.

Furthermore, Eq. (3) and Eq. (4) indicate that the gap between the upper and lower bound is $p_1 \times p_2$. For common neighbor feature and path feature whose $p_1 \times p_2$ values are small (Supplementary Table S1), the performance of their indexes should not fluctuate significantly and is mainly determined by $p_1 - p_2$. Hence, it is predicted that the AUC by different indexes in different networks, regardless of their types and sizes, scale as $p_1 - p_2$, which is empirically verified in Fig. 2 (extended discussion is presented in Supplementary Information S5).

The capability of a feature in the supervised prediction. In the supervised approach, the index value is the input of a machine learning based classifier. We use the Random Forest classifier [35, 39, 53] in the main text. Similar results by Gradient Boosting [42, 60] and AdaBoost [39, 61] are presented in Supplementary Information S6. The classifier finds a mapping function $y = f(x)$ to transform the input x to the score y for prediction, with the aim to further improve the prediction performance. Therefore, a more accurate prediction is expected as the non-fixed mapping function provides more flexibility to properly rank samples in L^P and L^N . Indeed, the best index value ranking is not the

optimal score ranking. Because node pairs in \bar{L}_1 are ranked behind L_2 , part of the negative samples still outscore positive samples. With the mapping function, the score ranking of L_1 , L_2 and $\bar{L}_1 \cup \bar{L}_2$ can be rearranged. In Supplementary Information S7, we compare the AUC value under different rankings of the three sets. The optimal score ranking is when samples in L_1 rank ahead of L_2 and samples in \bar{L}_1 and \bar{L}_2 (note that they have the same index value x , hence with the same score y) lie between L_1 and L_2 (Fig. 1d). In this case, no negative sample has a higher score than the positive sample. Hence, if the classifier could find the right mapping function, the prediction's upper bound becomes

$$\text{AUC}'_{\text{upper}} = \frac{n'}{n} + \frac{1}{2} \frac{n''}{n} = p_1 + (1 - p_1)p_2 + \frac{1}{2}(1 - p_1)(1 - p_2) = \frac{1}{2} + \frac{p_1 + p_2 - p_1p_2}{2}. \quad (5)$$

The capability of a topological feature in supervised link prediction ($\text{AUC}'_{\text{upper}}$) is empirically verified in 550 networks (Supplementary Figure S3). In general, the performance of supervised prediction improves, in line with the expectation (Supplementary Figure S4). More importantly, Eq. (5) suggests that the improvement is not merely by getting closer to the original upper bound. Instead, the maximum capability of a feature is lifted by an extent $\Delta = (1 - p_1)p_2$. Therefore, if the results about the maximum capability were correct, we would expect that when an unsupervised prediction is already close to the maximum capability $\text{AUC}_{\text{upper}}$, the prediction performance would be further improved in this network by Δ if a machine learning algorithm is applied. This hypothesis is verified in Fig. 3, supporting the proposed capability of a feature.

Applications. In link prediction, a critical choice we typically face is to decide whether to switch to another topological feature for a better prediction or to keep the same topological feature but try other indexes. Traditionally, such a decision can only be made by numerating the performance of all indexes associated with a feature. With the mathematical expression of $\text{AUC}_{\text{upper}}$ and $\text{AUC}'_{\text{upper}}$, however, the capability of a feature can be easily estimated from the measurement of one single index. Because p_1 and p_2 are only related to node pairs that hold the topological feature, their values are simultaneously known once an index is applied to L^P and L^N . The capability can therefore serve as theoretical guidance for feature selection. Take the two networks in Table II as an example. When using the index LHN-I to make an unsupervised link prediction, we obtain $\text{AUC} = 0.497$ in both networks. It is obvious that LHN-I should not be considered as its prediction is worse than a random guess. But the next question is, should we try other indexes of the common neighbor feature

or should we turn to a different topological feature? With the p_1 and p_2 values obtained through the calculation of LHN-I, Eq. (4) gives the maximum capability of the common neighbor feature: $AUC_{upper} = 0.497$ for network A and $AUC_{upper} = 0.974$ for network B. Consequently, the strategy is to switch to a new topological feature in network A, as any indexes related to common neighbor are doomed to fail. For network B, however, we can keep using the common neighbor feature and try other indexes. Indeed, for 8 indexes in the family of common neighbor, the prediction results halt at $AUC = 0.497$ in network A, but increase to $AUC = 0.77$ in network B, supporting the decision made through the estimated maximum capability (see another example in Supplementary Information S8).

Likewise, the value $\Delta = (1 - p_1)p_2$ theoretically predicts the benefit of applying a supervised method, which can be helpful in method selection. When the performance of unsupervised prediction is similar in two networks (Table III), can we decide and explain if an improvement is expected by adopting a machine learning based classifier? With the p_1 and p_2 value, first, we can tell that the unsupervised prediction in both networks are close to the upper bound AUC_{upper} . Furthermore, Δ predicts that the upper bound AUC'_{upper} can increase as much as 0.111 in network C, but could not go any further in network D. Hence, the machine learning algorithm can be beneficial only in Network C, which is verified by the performance of supervised prediction in Table III.

Moreover, once the theoretical expression of p_1 and p_2 are derived, we can quantitatively explore the structural characteristics that make a topological feature effective in link prediction. Take the common neighbor as an example. It is one of the most frequently used topological features. Yet, we still lack a quantitative assessment of the type of networks in which the common neighbor feature would work or fail. One may intuitively expect that the performance of the common neighbor is associated with the clustering coefficient C , as this feature is more prominent in clustered networks [62, 63]. However, numerical results show that the clustering coefficient can not fully explain the prediction performance (Figs. 4a, b). While the AUC reaches a high value and saturates when C is sufficiently large, it fluctuates significantly for small values of C . This is a particular case when the supervised approach is applied: even when the network is not clustered at all ($C = 0$), a supervised prediction can still give rise to an AUC value close to 0.8.

To explore structural characteristics of the common neighbor feature, we derive the ana-

lytical expression of p_1 and p_2 as (see the deduction in Supplementary Information S9)

$$p'_1 = \frac{3 * N_{\Delta} - S_{\Delta}}{L} \quad (6)$$

$$p'_2 = \frac{N_{\wedge} - S_{\wedge}}{\frac{N(N-1)}{2} - L}. \quad (7)$$

In Eq. (6) and Eq. (7), N_{Δ} and N_{\wedge} are the number of closed and open triangles, respectively. S_{Δ} is the number of times that a link is shared by multiple triangles, and S_{\wedge} is the number of times that an unconnected node pair is shared by other open triangles. The mathematical expression by Eq. (6) and Eq. (7) quantitatively reveal structural characteristics as well as their interplay related to the common neighbor feature in link prediction. p'_1 depends on the number of closed triangles, hence strongly correlated with the clustering coefficient C (Fig. 4c). But p'_2 depends on the number of open triangles that is unrelated to C (Fig. 4d). This explains why the clustering coefficient alone is insufficient to characterize the capability of the common neighbor feature, as the prediction performance is determined by a combination of both p'_1 and p'_2 . What is more, the denominator in the expression of p'_2 scales with N^2 . Therefore, it is expected that prediction results in large networks should demonstrate a stronger dependence with C , as the p'_2 in general vanishes quickly with N (Supplementary Figure S6a). But networks with many “leaves” structures have the number of open triangles scaling non-linearly with N , giving rise to a non-zero p'_2 value for large N (Supplementary Figure S6b). In such networks, even though the network is sufficiently large, the abundance of local clusters is still insufficient to quantify the capability of the common neighbor feature.

Discussion

To summarize, we quantify the maximum capability of a topological feature in link prediction by analyzing the upper bound of prediction accuracy. Given a variety of indexes, different approaches to utilize the index, and different measures of prediction performance, there seems no simple answer to this question. Nevertheless, we identify regularities underlying the link prediction task, leading to the mathematical expression for the performance upper bound by a topological feature. The maximum capability of a feature only depends on the extent to which the feature is held in missing and nonexistent links but does not depend on how a related index gauges the feature. Hence, a family of indexes associated with one topological feature shares the same performance upper bound, which can be used

to decide if a new feature or a new index is needed to advance the prediction. The capability of a feature is lifted by applying the supervised approach, whose magnitude can be theoretically derived, allowing us to estimate the benefit of applying machine learning algorithms in link prediction. Finally, using the common neighbor feature as an example, we quantitatively show how the interplay of different structural characteristics determines the prediction performance in different networks, which can not be fully explained by the clustering coefficient. Our theoretical results are verified by 550 empirically networks, demonstrating a strong universality.

To make the flow of the paper more consistent, we choose to present the performance by AUC measure in the main text. It is noteworthy that the framework also works for precision. Indeed, results presented in Supplementary Information S3 bring insights on how to interpret the performance by precision measure. As the precision involves a hyper-parameter L_k for the cutoff of the top-k node pairs (Eq. (2)), the outcome may change with the choice of L_k . For example, a prediction measured by AUC is poor due to the small p_1 value of the network. But the precision can be high if L_k is chosen such that $L_k < p_1|L^P|$ (see an example in Supplementary Information S10). This phenomenon urges more caution when comparing link prediction precision among networks with different sizes. At least the p_1 values of different networks should be considered for a more comprehensive interpretation. Indeed, the precision and AUC reflect different aspects of prediction performance. When L_k is relatively small, precision is mainly affected by how close the actually score ranking is to the optimal ranking, as the precision upper bound is 100% if only the top tires are considered (Eq. (S20)). On the contrary, the AUC measure, which does not rely on a hyper-parameter, is more capable to provide the overall extent to which the positive samples outscore the negative samples [58, 59].

When making use of a new topological feature, the AUC_{upper} and AUC'_{upper} derived in this work can serve as the pre-evaluation of its potential. It is interesting to note that even for the unsupervised prediction, the average accuracy by indexes of common neighbor, path, and path of length three is close to the feature's maximum capability. This suggests that future development on indexes related to these features may only bring marginal advances. The feature of heterogeneity usually has a large p_1 value, hence demonstrating a higher capability. But related indexes are coarse-gained, which can not efficiently separate positive and negative samples. Hence, its high capability is not fully utilized. A promising direction

is to design a new index by combining multiple features [53, 57, 64] or to use a stacking method for prediction [39]. The network embedding can merge and learn different topological and proximity features [19–21, 65], which may have great potential in link prediction problems. The relationship between the capability of a feature and the link predictability of the network is another intriguing question to tackle. Identifying the single or a set of topological features that have the capability closest to the link predictability may reveal the intrinsic characteristics of the network. Taken together, our work uncovers a regularity in the link prediction problem, which not only provide a theoretical upper bound for leveraging a topological feature but also shed light on a range of related questions that can be analyzed in the future.

Code availability

The code used in this study is available at <https://github.com/YijunRan/MCLP>.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62173065), the Industry-University-Research Innovation Fund for Chinese Universities (No. 2021ALA03016), University Innovation Research Group of Chongqing (No. CXQT21005).

Author contributions

YJ. R., XK. X. and T. J. designed the research. YJ. R. performed numerical analyses. YJ. R. and T. J. derived the analytical results. YJ. R. and T. J. prepared the initial draft. XK. X. and T. J. revised the draft to the final version of the paper.

Competing financial interests

The authors declare no competing financial interests.

Features	Indexes (Abbr.)
Common Neighbor	Common Neighbor Index (CN)[33], Adamic-Adar Index (AA)[47], Resource Allocation Index (RA)[48], Salton Index (Salton)[31], Sørensen Index (SI)[40], Hub Promoted Index (HPI)[7], Hub Depressed Index (HDI)[48], Jaccard Index (Jaccard)[33], Leicht-Holme-Newman Index (LHN-I)[49]
Path	Local Path Index (LP)[50], Katz Index (Katz)[53], FriendLink Index (FL)[51], Shortest Path Length Index (SPL)[53, 54], Relation Strength Similarity Index (RSS)[52]
Heterogeneity	Heterogeneity Index (HEI)[55], Homogeneity Index (HOI)[55]
Path of Length Three	Paths of Length Three Index (L3)[12], L3-based Cannistraci-Hebb Index (CH2-L3)[56]

TABLE I: The 18 indexes related to 4 topological features

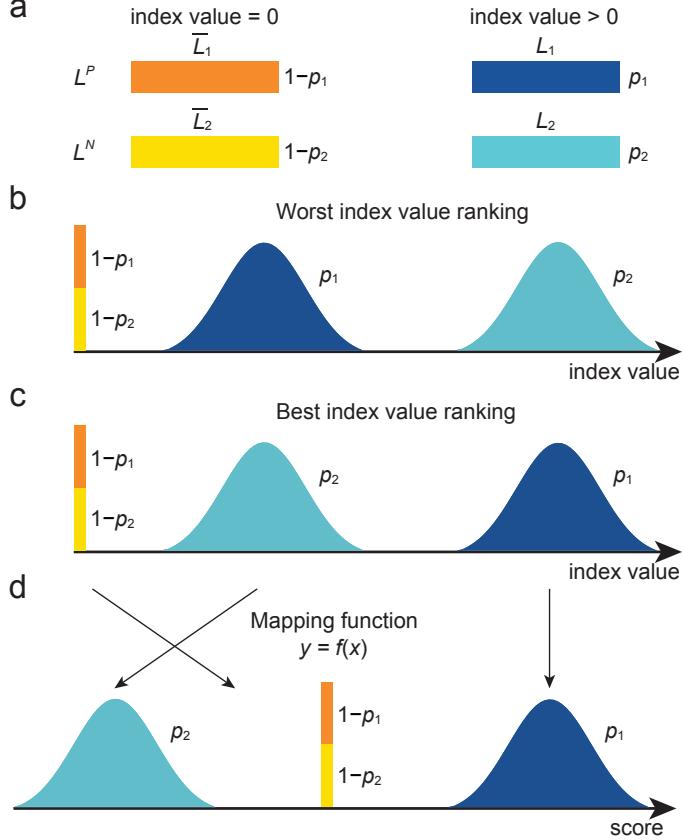


FIG. 1: An illustration of different link prediction performance. (a) Samples in the positive set L^P can be divided into two subsets based on whether the feature is held or not. L_1 is the subset of L^P in which node pairs hold the feature, whereas the complement set \overline{L}_1 is composed of node pairs that do not hold the feature. As the index is designed to quantify the feature, it should assign non-zero values to samples in L_1 and value 0 to samples in \overline{L}_1 . Similarly, the negative set L^N can also be divided into two subsets L_2 and \overline{L}_2 . Assume that L_1 takes a fraction p_1 of L^P and L_2 takes a fraction p_2 of L^N . Because samples in \overline{L}_1 and \overline{L}_2 have the same index value 0, the prediction performance mainly relies on the ranking of L_1 and L_2 . (b) The worst index value ranking is when L_2 is systematically ranked ahead of L_1 . (c) The best index value ranking is just the opposite when L_1 is systematically ranked ahead of L_2 . Note that in both cases, \overline{L}_1 and \overline{L}_2 are always ranked behind L_1 and L_2 in the unsupervised approach. (d) In supervised prediction, the machine learning based classifier can find a mapping function $y = f(x)$ to transfer the index value to the score for prediction. Hence, the relative position among L_1 , L_2 and $\overline{L}_1 \cup \overline{L}_2$ can be further optimized. Because samples in $\overline{L}_1 \cup \overline{L}_2$ have the same index value, they should have the same score. The optimal score ranking is to assign a score to $\overline{L}_1 \cup \overline{L}_2$ that makes it lie between L_1 and L_2 . In this case, no negative samples have a higher score than positive samples. Note that different scenarios described here can also be used to explain different precision values obtained (see Supplementary Information S3).

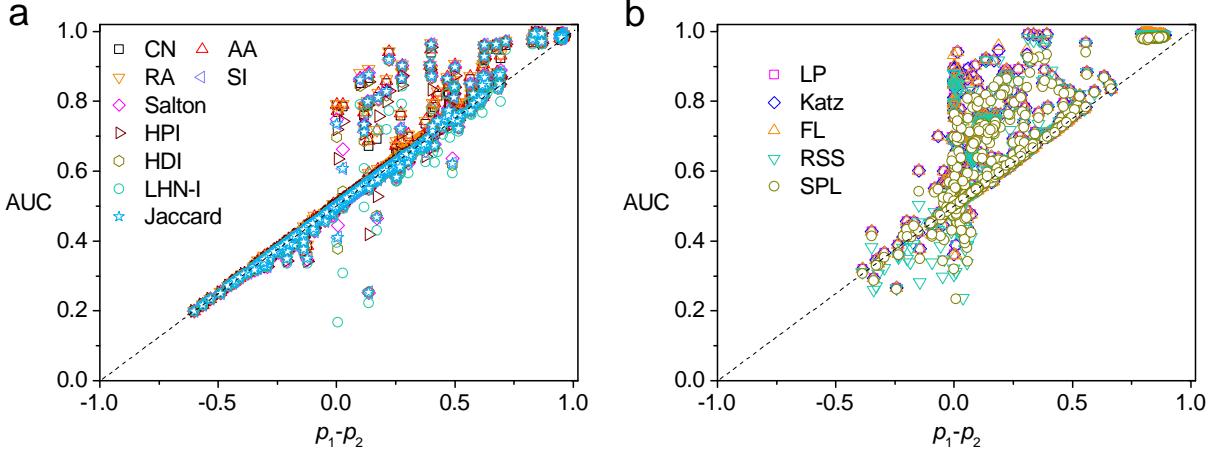


FIG. 2: **The scaling of the AUC values.** Eq. (3) and Eq. (4) suggest that the actual prediction by an index fluctuates within $p_1 \times p_2$. Therefore, for the common neighbor feature (**a**) and the path feature (**b**) whose $p_1 \times p_2$ values are small, the link prediction performance by different indexes roughly scales as $p_1 - p_2$. For each network, we randomly generate 200 pairs of L^P and L^N sets (Supplementary Information S2). The p_1 , p_2 and the corresponding AUC obtained may vary slightly in different sampled L^N 's and L^P 's. In the figure, we use the average value.

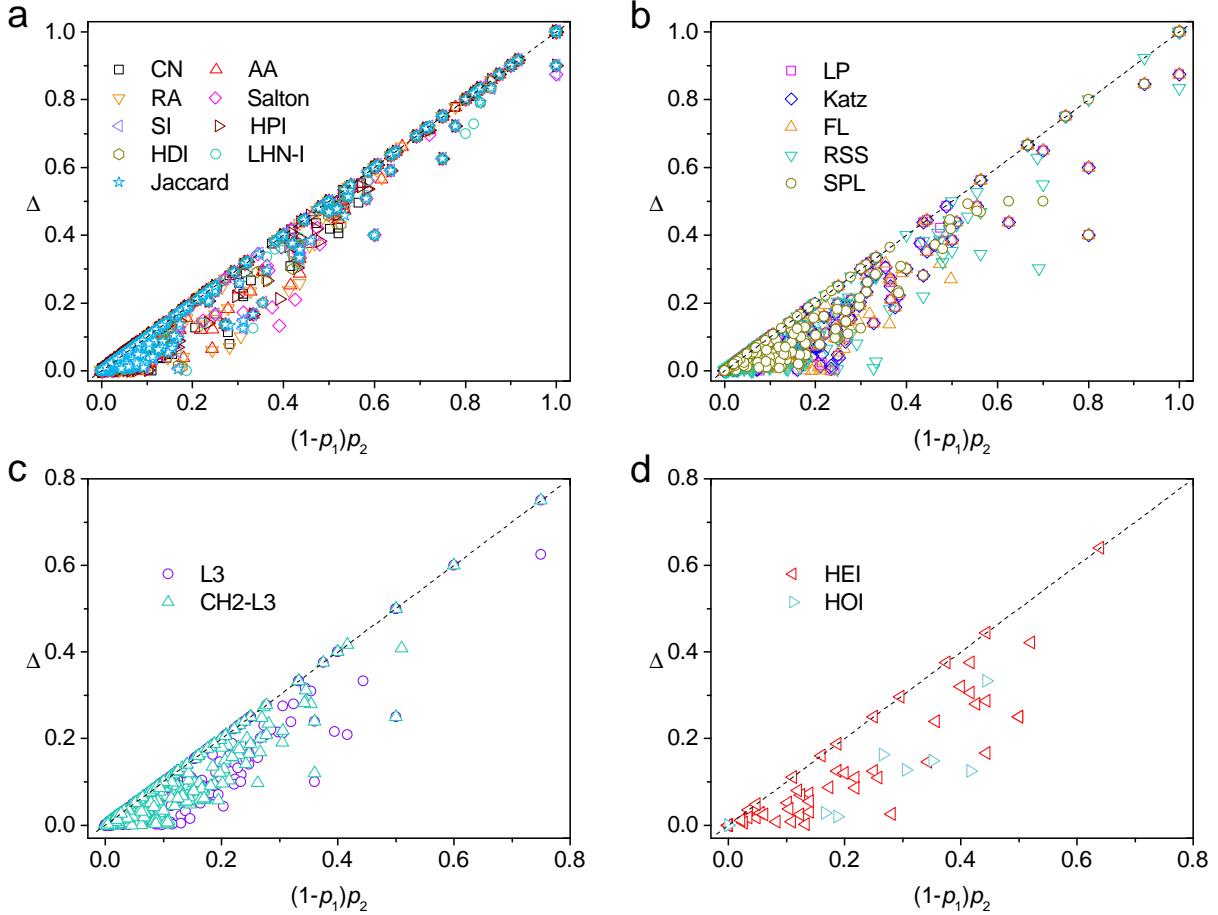


FIG. 3: The actual and predicted improvement by the supervised approach. Eq. (5) suggests that the supervised approach can lift the capability of a feature by $(1 - p_1)p_2$. To test it, we select networks in which the unsupervised prediction by an index is already close to its upper bound (measured AUC is more than 95% of AUC_{upper}). For these networks, we input the same index values of L^P and L^N to the classifier to obtain the supervised prediction results. Δ for a given network is measured as the AUC difference between the supervised and unsupervised prediction. For each network, we randomly generate 200 pairs of L^P and L^N sets (Supplementary Information S2). Therefore, for one network and one index, there could be multiple pairs of L^N and L^P in which the unsupervised prediction is close to AUC_{upper} . If this is the case, we choose the L^N and L^P that give rise to the highest AUC. The empirically measured Δ for different networks and indexes are close to $(1 - p_1)p_2$, in line with the prediction by Eq. (5).

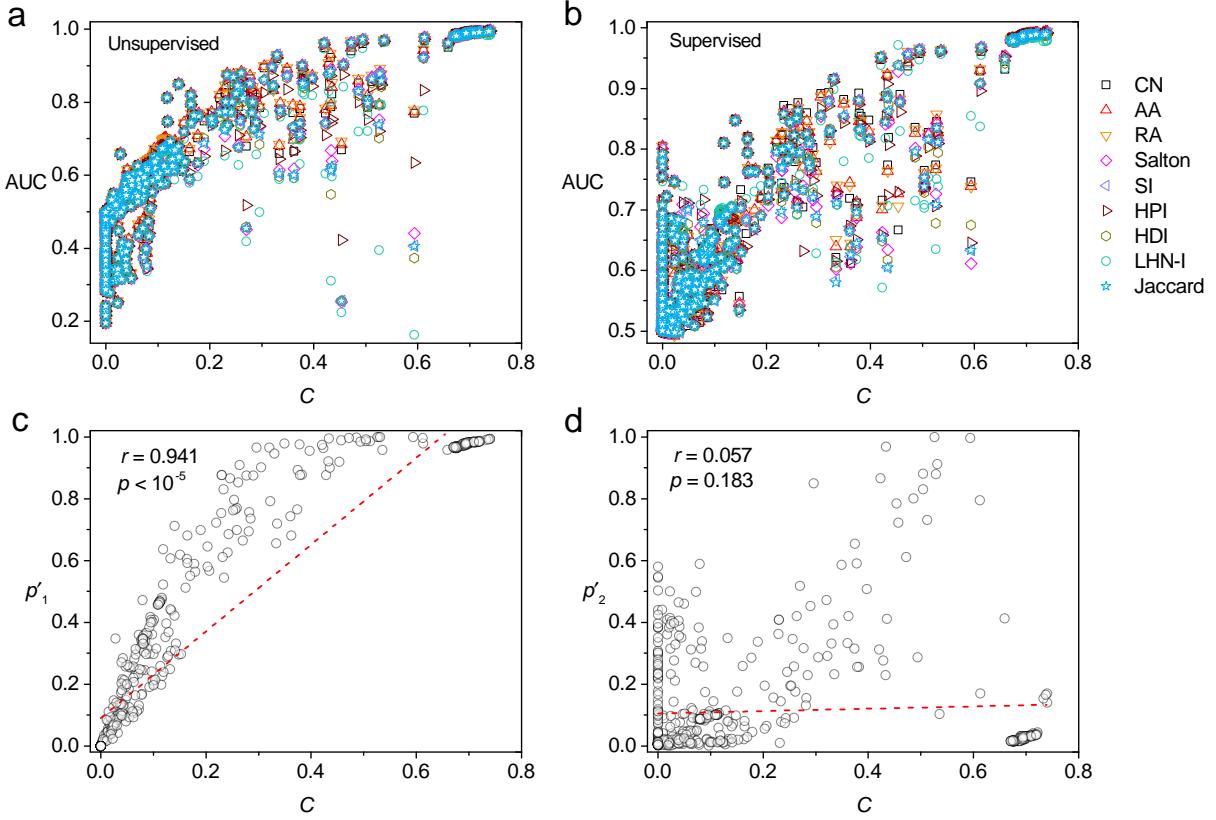


FIG. 4: The structural characteristics related to the common neighbor feature in link prediction. (a, b) It is intuitively expected that the clustering coefficient C is directly related to the performance of indexes based on the common neighbor feature. But the supervised (a) and unsupervised (b) prediction results show that C can not fully explain the effectiveness of the common neighbor feature. AUC demonstrates a significant variability for some C values in both cases. (c, d) According to Eq. (6) and Eq. (7), p_1 depends on the number of closed triangles, and p_2 depends on the number of open triangles. Therefore, p_1 should demonstrate a strong correlation with the clustering coefficient C and p_2 should be independent of C , which is empirically confirmed. The r is the Pearson correlation coefficient, and the p -value is from the Student's t-test.

	LHN-I	p_1	p_2	AUC _{upper}	CN	AA	RA	Salton	SI	HPI	HDI	Jaccard
Network A	0.497	0.0	0.006	0.497	0.497	0.497	0.497	0.497	0.497	0.497	0.497	0.497
Network B	0.497	0.972	0.872	0.974	0.766	0.768	0.755	0.725	0.705	0.717	0.679	0.705

TABLE II: The performance of the unsupervised prediction using LHN-I is the same in both networks A and B. However, using the p_1 and p_2 value, the capability of the common neighbor feature can be estimated. The common neighbor feature is not suitable for network A but has potential in network B. The estimation is confirmed by the prediction results of other indexes. The network A is “Norwegian_Board_of_Directors_net2mode_2010-09-01”, and the network B is “577ee40d58d31bd664bac0ef” in the data set by [39, 46].

	Unsupervised					Supervised								
	CN	p_1	p_2	AUC _{upper}	Δ	CN	AA	RA	Salton	SI	HPI	HDI	LHN-I	Jaccard
Network C	0.778	0.667	0.333	0.778	0.111	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889	0.889
Network D	0.773	0.545	0.0	0.773	0.0	0.773	0.773	0.773	0.773	0.773	0.773	0.773	0.773	0.773

TABLE III: The unsupervised prediction using index CN gives rise to similar AUC values in networks A and B. Using the p_1 and p_2 value, the improvement by applying a supervised method can be estimated in both networks. The actual supervised prediction results confirm the estimation. The network C is “Malaria_var_DBLa.HVR_networks.HVR_networks_3”, and the network D is “57574842bd3e93b53c695556” in the data set by [39, 46].

-
- [1] Barabási, A.-L. *Network science* (Cambridge University Press, 2016).
 - [2] Newman, M. *Networks* (Oxford University Press, 2018).
 - [3] Clauset, A., Moore, C. & Newman, M. E. Hierarchical structure and the prediction of missing links in networks. *Nature* **453**, 98–101 (2008).
 - [4] Guimerà, R. & Sales-Pardo, M. Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences* **106**, 22073–22078 (2009).
 - [5] Guimerà, R. One model to rule them all in network science? *Proceedings of the National Academy of Sciences* **117**, 25195–25197 (2020).
 - [6] Fajardo-Fontiveros, O., Guimerà, R. & Sales-Pardo, M. Node metadata can produce predictability crossovers in network inference problems. *Physical Review X* **12**, 011010 (2022).
 - [7] Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N. & Barabási, A. L. Hierarchical organization of modularity in metabolic networks. *Science* **297**, 1551 (2002).
 - [8] Barzel, B. & Barabási, A.-L. Network link prediction by global silencing of indirect correlations. *Nature Biotechnology* **31**, 720–725 (2013).
 - [9] Cheng, F., Kovács, I. A. & Barabási, A.-L. Network-based prediction of drug combinations. *Nature Communications* **10**, 1–11 (2019).
 - [10] Gysi, D. M. *et al.* Network medicine framework for identifying drug-repurposing opportunities for covid-19. *Proceedings of the National Academy of Sciences* **118**, 1–11 (2021).
 - [11] Yu, H. *et al.* High-quality binary protein interaction map of the yeast interactome network. *Science* **322**, 104–110 (2008).
 - [12] Kovács, I. A. *et al.* Network-based prediction of protein interactions. *Nature Communications* **10**, 1–8 (2019).
 - [13] Li, S. *et al.* Friend recommendation for cross marketing in online brand community based on intelligent attention allocation link prediction algorithm. *Expert Systems with Applications* **139**, 112839 (2020).
 - [14] Santos, F. P., Lelkes, Y. & Levin, S. A. Link recommendation algorithms and dynamics of polarization in online social networks. *Proceedings of the National Academy of Sciences* **118**, 1–9 (2021).
 - [15] Godoy-Lorite, A., Guimerà, R., Moore, C. & Sales-Pardo, M. Accurate and scalable social rec-

- ommendation using mixed-membership stochastic block models. *Proceedings of the National Academy of Sciences* **113**, 14207–14212 (2016).
- [16] Kart, O., Ulucay, O., Bingol, B. & Isik, Z. A machine learning-based recommendation model for bipartite networks. *Physica A: Statistical Mechanics and its Applications* **553**, 124287 (2020).
 - [17] Gu, W., Tandon, A., Ahn, Y.-Y. & Radicchi, F. Principled approach to the selection of the embedding dimension of networks. *Nature Communications* **12**, 1–10 (2021).
 - [18] Xue, J. *et al.* Quantifying the spatial homogeneity of urban road networks via graph neural networks. *Nature Machine Intelligence* **4**, 246–257 (2022).
 - [19] Perozzi, B., Al-Rfou, R. & Skiena, S. Deepwalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 701–710 (ACM, 2014).
 - [20] Grover, A. & Leskovec, J. Node2vec: scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–864 (ACM, 2016).
 - [21] Wang, D., Cui, P. & Zhu, W. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 1225–1234 (ACM, 2016).
 - [22] Wang, P., Xu, B., Wu, Y. & Zhou, X. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences* **58**, 1–38 (2015).
 - [23] Martínez, V., Berzal, F. & Cubero, J.-C. A survey of link prediction in complex networks. *ACM Computing Surveys* **49**, 1–33 (2016).
 - [24] Haghani, S. & Keyvanpour, M. R. A systemic analysis of link prediction in social network. *Artificial Intelligence Review* **52**, 1961–1995 (2019).
 - [25] Cao, R.-M., Liu, S.-Y. & Xu, X.-K. Network embedding for link prediction: the pitfall and improvement. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **29**, 103102 (2019).
 - [26] Kumar, A., Singh, S. S., Singh, K. & Biswas, B. Link prediction techniques, applications, and performance: a survey. *Physica A: Statistical Mechanics and its Applications* **553**, 124289 (2020).
 - [27] Daud, N. N., Ab Hamid, S. H., Saadoon, M., Sahran, F. & Anuar, N. B. Applications of link prediction in social networks: a review. *Journal of Network and Computer Applications* **166**,

- 102716 (2020).
- [28] Mara, A. C., Lijffijt, J. & De Bie, T. Benchmarking network embedding models for link prediction: are we making progress? In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics*, 138–147 (IEEE, 2020).
 - [29] Zhou, T. Progresses and challenges in link prediction. *Iscience* **24**, 103217 (2021).
 - [30] Lü, L., Pan, L., Zhou, T., Zhang, Y.-C. & Stanley, H. E. Toward link predictability of complex networks. *Proceedings of the National Academy of Sciences* **112**, 2325–2330 (2015).
 - [31] Sun, J. *et al.* Revealing the predictability of intrinsic structure in complex networks. *Nature Communications* **11**, 1–10 (2020).
 - [32] Tang, D. *et al.* Predictability of real temporal networks. *National Science Review* **7**, 929–937 (2020).
 - [33] Liben-Nowell, D. & Kleinberg, J. The link prediction problem for social networks. *Journal of the Association for Information Science and Technology* **58**, 1019–1031 (2007).
 - [34] Lü, L. & Zhou, T. Link prediction in complex networks: a survey. *Physica A: Statistical Mechanics and its Applications* **390**, 1150–1170 (2011).
 - [35] Wang, D., Pedreschi, D., Song, C., Giannotti, F. & Barabási, A.-L. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1100–1108 (ACM, 2011).
 - [36] Muniz, C. P., Goldschmidt, R. & Choren, R. Combining contextual, temporal and topological information for unsupervised link prediction in social networks. *Knowledge-Based Systems* **156**, 129–137 (2018).
 - [37] Lee, Y.-L. & Zhou, T. Collaborative filtering approach to link prediction. *Physica A: Statistical Mechanics and its Applications* **578**, 126107 (2021).
 - [38] Benson, A. R., Abebe, R., Schaub, M. T., Jadbabaie, A. & Kleinberg, J. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences* **115**, E11221–E11230 (2018).
 - [39] Ghasemian, A., Hosseini Mardi, H., Galstyan, A., Airolidi, E. M. & Clauset, A. Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences* **117**, 23393–23400 (2020).
 - [40] Ghorbanzadeh, H., Sheikhahmadi, A., Jalili, M. & Sulaimany, S. A hybrid method of link prediction in directed graphs. *Expert Systems with Applications* **165**, 113896 (2021).

- [41] Zhang, T., Zhang, K., Li, X., Lv, L. & Sun, Q. Semi-supervised link prediction based on non-negative matrix factorization for temporal networks. *Chaos, Solitons & Fractals* **145**, 110769 (2021).
- [42] Kumar, S., Mallik, A. & Panda, B. Link prediction in complex networks using node centrality and light gradient boosting machine. *World Wide Web* 1–27 (2022).
- [43] Zhang, M. & Chen, Y. Link prediction based on graph neural networks. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, 5171–5181 (ACM, 2018).
- [44] García-Pérez, G., Aliakbarisani, R., Ghasemi, A. & Serrano, M. Á. Precision as a measure of predictability of missing links in real networks. *Physical Review E* **101**, 052318 (2020).
- [45] Nasiri, E., Berahmand, K. & Li, Y. A new link prediction in multiplex networks using topologically biased random walks. *Chaos, Solitons & Fractals* **151**, 111230 (2021).
- [46] Broido, A. D. & Clauset, A. Scale-free networks are rare. *Nature Communications* **10**, 1–10 (2019).
- [47] Adamic, L. A. & Adar, E. Friends and neighbors on the web. *Social Networks* **25**, 211–230 (2003).
- [48] Zhou, T., Lü, L. & Zhang, Y.-C. Predicting missing links via local information. *The European Physical Journal B* **71**, 623–630 (2009).
- [49] Leicht, E. A., Holme, P. & Newman, M. E. Vertex similarity in networks. *Physical Review E* **73**, 026120 (2006).
- [50] Lü, L., Jin, C.-H. & Zhou, T. Similarity index based on local paths for link prediction of complex networks. *Physical Review E* **80**, 046122 (2009).
- [51] Papadimitriou, A., Symeonidis, P. & Manolopoulos, Y. Friendlink: link prediction in social networks via bounded local path traversal. In *2011 International Conference on Computational Aspects of Social Networks*, 66–71 (IEEE, 2011).
- [52] Chen, H.-H., Gou, L., Zhang, X. & Giles, C. L. Discovering missing links in networks using vertex similarity measures. In *Proceedings of the 27th annual ACM Symposium on Applied Computing*, 138–143 (ACM, 2012).
- [53] Ran, Y., Liu, T., Jia, T. & Xu, X.-K. A novel similarity measure for mining missing links in long-path networks. *Chinese Physics B* **31**, 068902 (2022).
- [54] Ran, Y., Liu, S.-Y., Yu, X., SHANG, K. & Jia, T. Predicting future links with new nodes in

- temporal academic networks. *Journal of Physics: Complexity* **3**, 015006 (2022).
- [55] Shang, K.-k., Li, T.-c., Small, M., Burton, D. & Wang, Y. Link prediction for tree-like networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **29**, 061103 (2019).
- [56] Muscoloni, A., Abdelhamid, I. & Cannistraci, C. V. Local-community network automata modelling based on length-three-paths for prediction of complex network structures in protein interactomes, food webs and more. *bioRxiv* 346916 (2018).
- [57] Zhou, T., Lee, Y.-L. & Wang, G. Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms. *Physica A: Statistical Mechanics and its Applications* **564**, 125532 (2021).
- [58] Lichtenwalter, R. & Chawla, N. V. Link prediction: fair and effective evaluation. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 376–383 (IEEE, 2012).
- [59] Yang, Y., Lichtenwalter, R. N. & Chawla, N. V. Evaluating link prediction methods. *Knowledge and Information Systems* **45**, 751–782 (2015).
- [60] Mahapatra, S. & Sahu, S. S. Improved prediction of protein-protein interaction using a hybrid of functional-link siamese neural network and gradient boosting machines. *Briefings in Bioinformatics* **22**, bbab255 (2021).
- [61] Shan, N., Li, L., Zhang, Y., Bai, S. & Chen, X. Supervised link prediction in multiplex networks. *Knowledge-Based Systems* **203**, 106168 (2020).
- [62] Feng, X., Zhao, J. & Xu, K. Link prediction in complex networks: a clustering perspective. *The European Physical Journal B* **85**, 1–9 (2012).
- [63] Liu, Y. *et al.* The degree-related clustering coefficient and its application to link prediction. *Physica A: Statistical Mechanics and its Applications* **454**, 24–33 (2016).
- [64] Ma, C., Bao, Z.-K. & Zhang, H.-F. Improving link prediction in complex networks by adaptively exploiting multiple structural features of networks. *Physics Letters A* **381**, 3369–3376 (2017).
- [65] Xie, Y., Gong, M., Wang, S., Liu, W. & Yu, B. Sim2vec: node similarity preserving network embedding. *Information Sciences* **495**, 37–51 (2019).
- [66] Liu, S.-Y., Xiao, J. & Xu, X.-K. Sign prediction by motif naive bayes model in social networks. *Information Sciences* **541**, 316–331 (2020).
- [67] Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *Journal of*

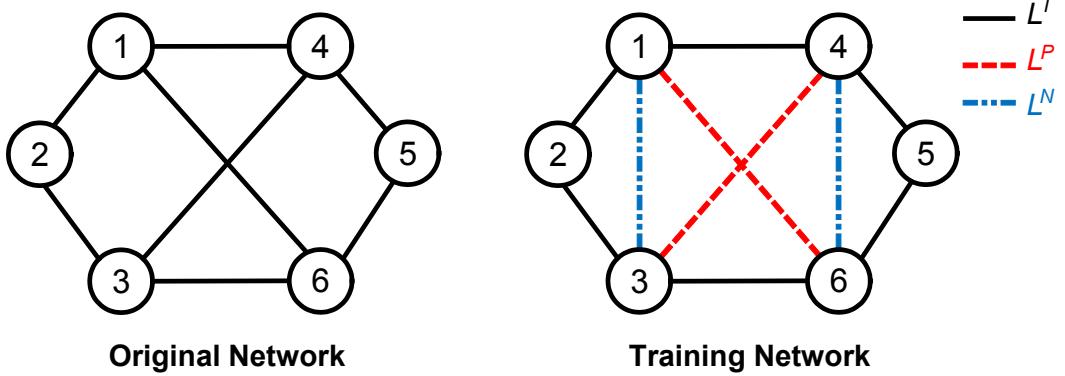
machine learning research **13**, 281–305 (2012).

- [68] Fawcett, T. An introduction to roc analysis. *Pattern recognition letters* **27**, 861–874 (2006).

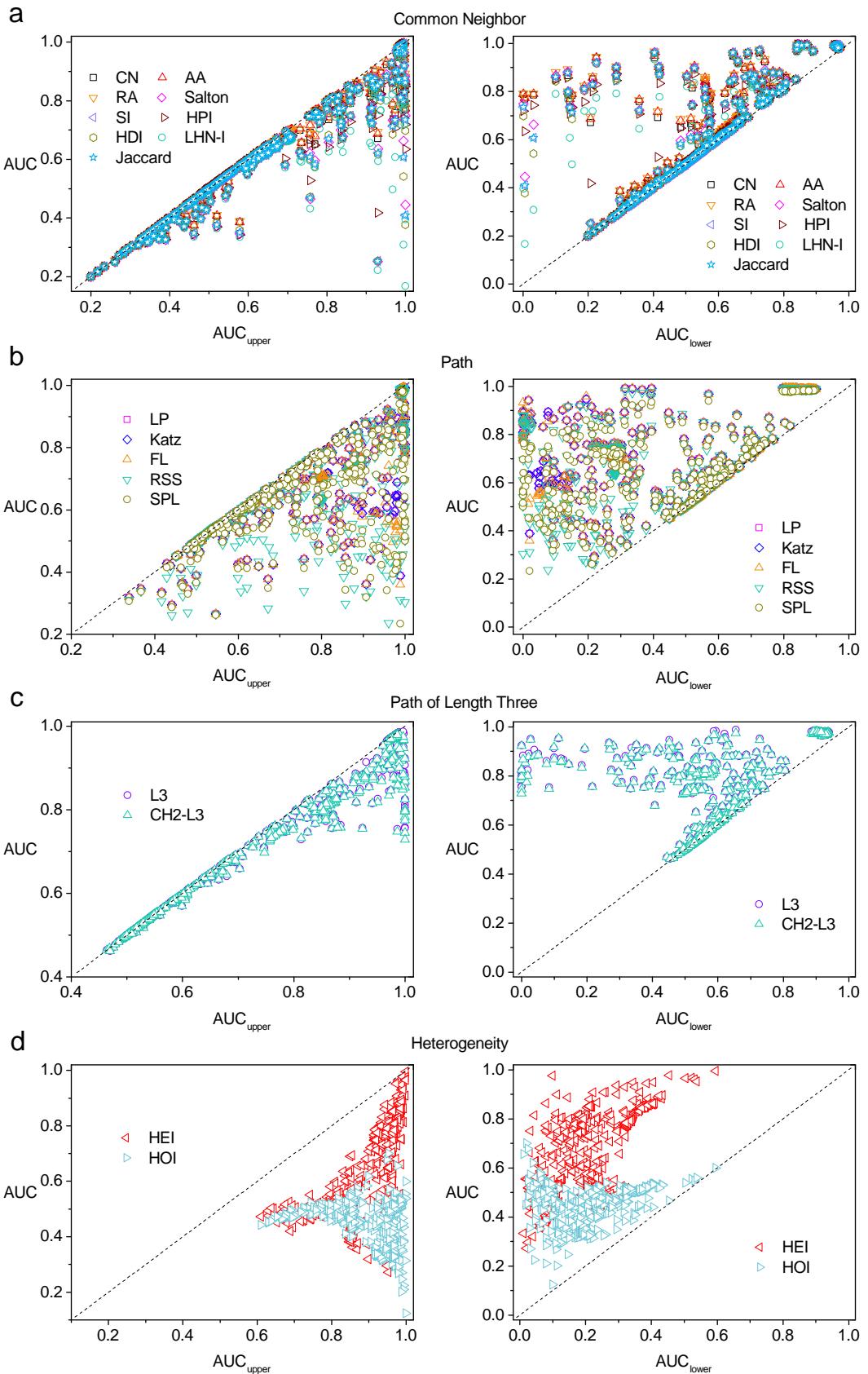
**Supplementary Information: The maximum capability of
a topological feature in link prediction**

Yijun Ran, Xiao-Ke Xu, Tao Jia¹

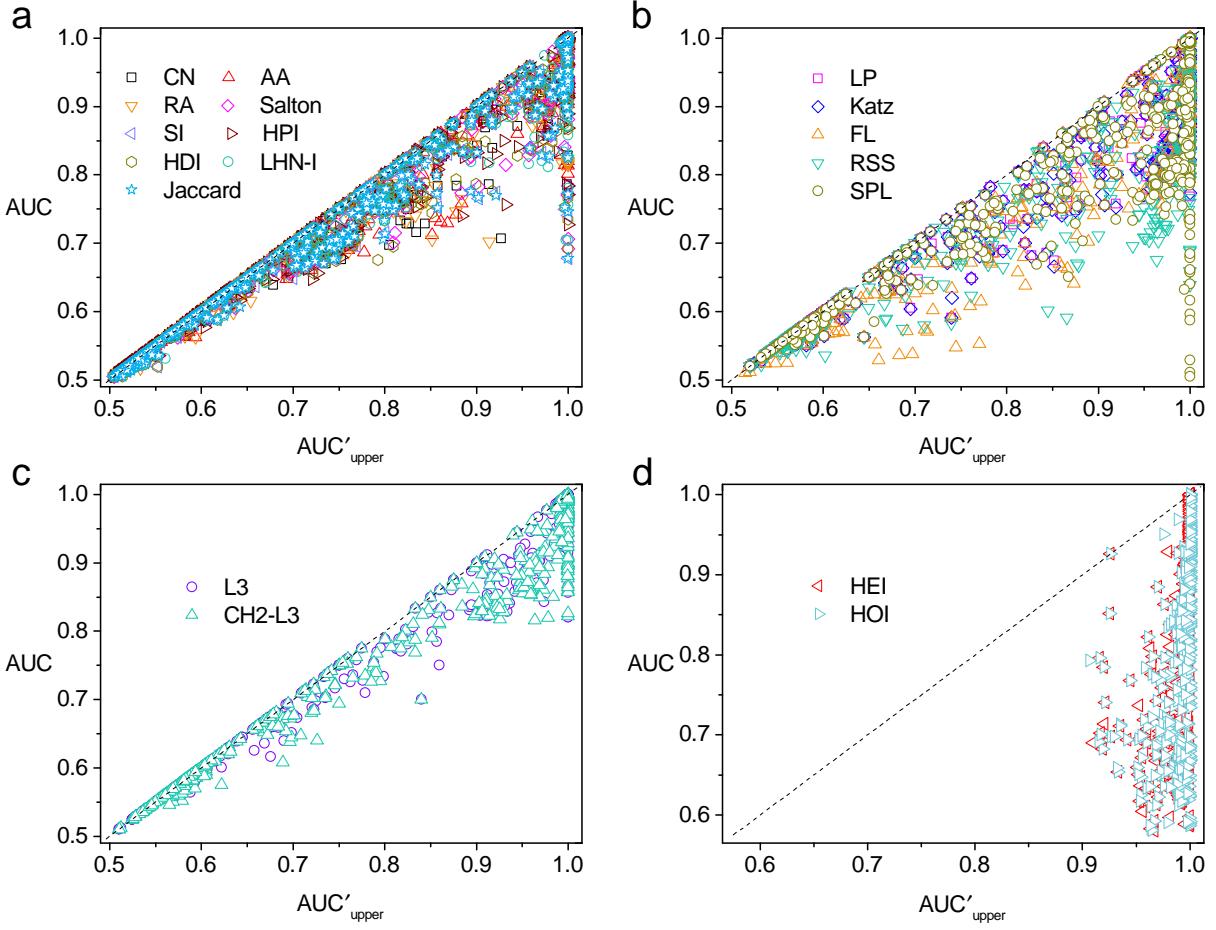
¹ *To whom correspondence should be addressed:* *tjia@swu.edu.cn*



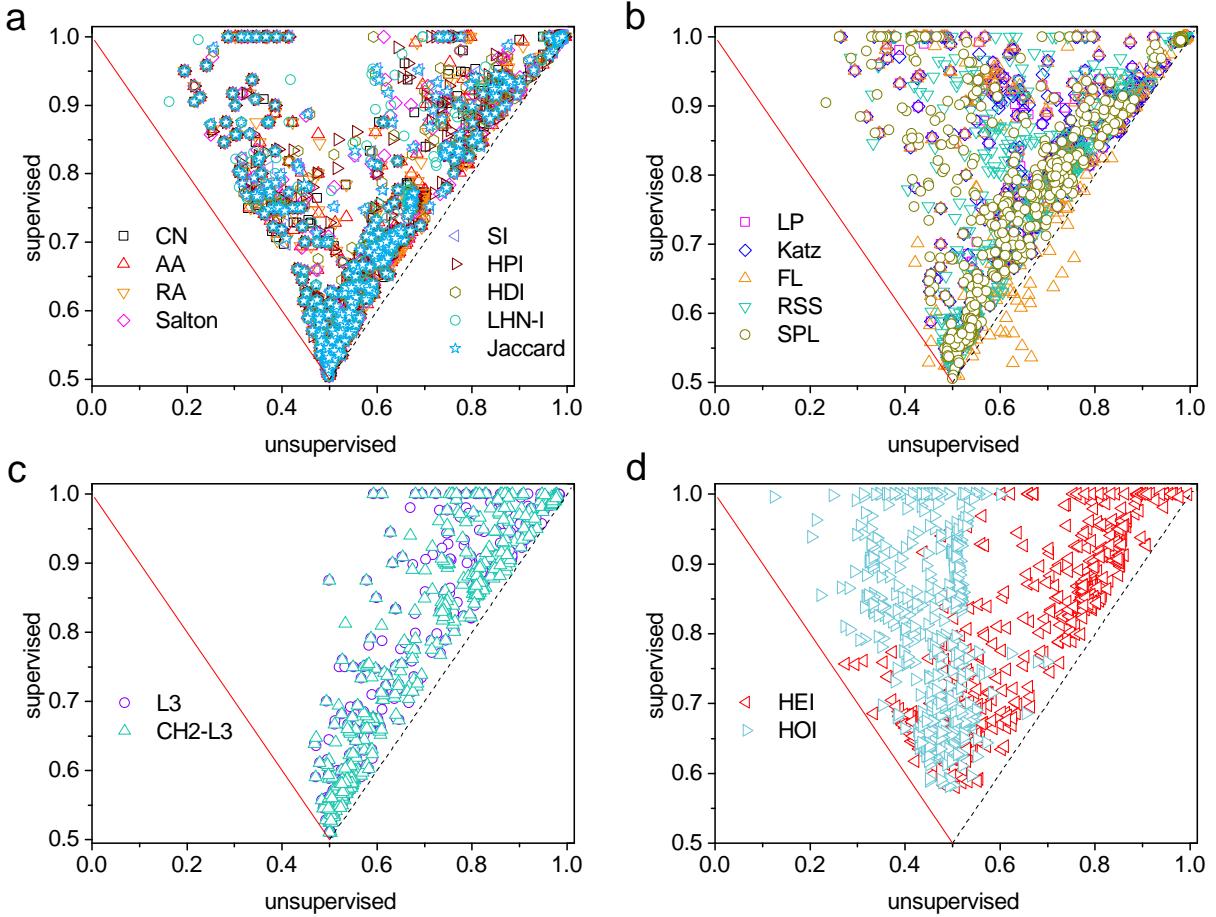
Supplementary Figure S1: An example of the link prediction problem. The common process of link prediction is that a set of existing links is removed randomly from the original network, which is marked as missing links L^P (the red dash lines in the training network). As the control group, a random set of node pairs that are not connected in the original network is selected as nonexistent links L^N (the blue dash lines in the training network). An index considers the topology based on the rest of the links L^T (the black solid lines in the training network) and assigns a value to each node pair in L^P and L^N . In unsupervised prediction, the index values are directly used as the score of samples in L^P and L^N . Assume that according to the index value we have $S_{16} = 0.3$, $S_{34} = 0.58$, $S_{13} = 0.2$, and $S_{46} = 0.58$. The quality of the prediction is measured by how the missing links are ranked ahead of the nonexistent ones. When using AUC to measure the prediction performance, we usually apply the random sampling approach. In each comparison, we randomly draw a node pair from L^P and a node pair from L^N , and compare their scores. Suppose 3 random comparisons are made. Node pairs 1-6 and 1-3, node pairs 1-6 and 4-6, and node pairs 3-4 and 4-6 are selected in each comparison. We have one case where node pair from L^P outscores that from L^N , and one case where node pairs from L^P and L^N have an equal score. According to Eq. (1) of the main text, the AUC can be estimated as $\frac{1+0.5}{3} = 0.5$. When using precision to measure the prediction performance, we rank node pairs according to their values S in descending order. In the example shown, the rank is 3-4, 4-6, 1-6, 1-3. If we select the hyper-parameter $L_k = 2$, the top-two node pairs (3-4 and 4-6) are considered. As node pair 3-4 is the true missing link whereas node pair 4-6 is not, the precision is 0.5.



Supplementary Figure S2: Eqs. (3) and (4) in the main text suggest that different indexes can have different prediction performances, but all indexes associated with one topological feature share the same AUC_{upper} and AUC_{lower} . This is confirmed by 18 indexes related to 4 topological features: common neighbor (**a**), path (**b**), path of length three (**c**), and heterogeneity (**d**). For each network, we randomly generate 200 pairs of L^P and L^N sets. In the figure, we use the average value of 200 samples.

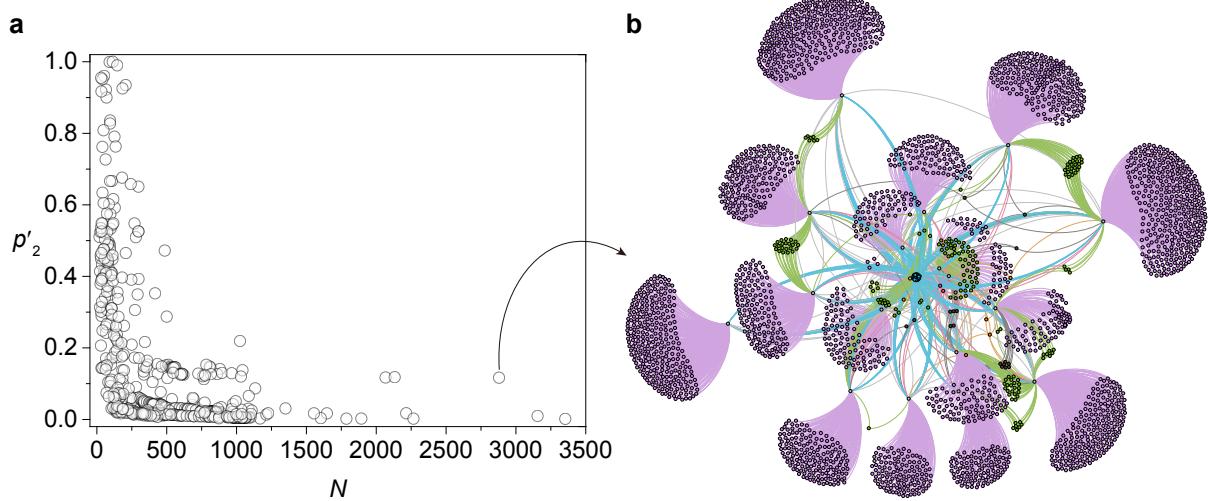


Supplementary Figure S3: Eq. (5) in the main text suggests that AUC'_{upper} sets the upper bound of the supervised prediction. This is confirmed by 18 indexes related to 4 topological features: common neighbor (**a**), path (**b**), path of length three (**c**), and heterogeneity (**d**). For each network, we randomly generate 200 pairs of L^P and L^N sets. In the figure, we choose the highest AUC from 200 samples as the performance of an index.



Supplementary Figure S4: Performance comparison (measured by AUC) between the unsupervised and supervised prediction. The dashed line is $y = x$. Almost all data points are above the line $y = x$, indicating that the supervised prediction in general gives rise to a higher AUC value compared with the unsupervised prediction in the same network based on the same index. In unsupervised prediction, the AUC value can be less than 0.5 (Fig. S2), suggesting that the feature is more prominent in negative samples. A simple fix in this circumstance is to consider the feature as the “negative feature”. When the AUC is below 0.5, the predictor will consider that a smaller index value corresponds to a higher probability that two nodes are truly connected. After this modification is applied, predictions with $AUC < 0.5$ will be lifted. The red solid line is $y = 1 - x$, corresponding to the performance after considering the “negative feature”. All data points are above $y = 1 - x$.

Hence the improvement by the supervised approach is not merely taking advantage of the “negative feature”. We use 18 indexes related to 4 topological features: common neighbor (a), path (b), path of length three (c), and heterogeneity (d). For each network, we randomly generate 200 pairs of L^P and L^N sets. In the figure, we use the average value of 200 samples for unsupervised results. For supervised results, we choose the highest AUC from 200 samples as the performance of an index.



Supplementary Figure S5: (a) In the analytical expression of p_2 (Eq.(7) of the main text), the denominator is dominated by N^2 . Hence, p_2 is expected to decay fast with the network size N . For a sufficiently large network, p_2 is expected to be zero. This is generally held in the empirical analyses of 550 networks. But exceptions are also found. Some large networks have a relatively large p_2 value. Such networks have many “leaves” structures, as illustrated in (b). The leaves will make the number of open triangles proportional to N^2 . Hence, p_2 will not vanish in such networks. The network in (b) is “5886685ba411221d0e7c677e” in the data set.

Features	Common Neighbor	Path of Length Three	Path	Heterogeneity
$p_1 \times p_2$	0.057 ± 0.152	0.155 ± 0.241	0.271 ± 0.315	0.695 ± 0.126

Supplementary Table S1: The value $p_1 \times p_2$ (mean±standard deviation) of each feature on testing set (L^P and L^N) applied to 550 empirical networks. For each network, we randomly generate 200 pairs of L^P and L^N sets. For p_1 and p_2 value of each network, we use the average value of 200 samples.

S1. THE 18 INDEXES USED IN THIS STUDY AND THE CLASSIFICATION OF THESE INDEXES

In this study, we select 18 indexes associated with 4 topological features to validate our quantitative framework presented in the main text. Here, we describe in detail the 18 indexes and how they are classified into 4 families.

A family of indexes based on common neighbor. We consider 9 indexes that gauge the common neighbor feature.

(1) Common Neighbor Index (CN)

The CN directly counts the number of common neighbors two nodes share [33]. It is defined as

$$S_{ab}^{\text{CN}} = |n(a) \cap n(b)|, \quad (\text{S1})$$

where $n(a)$ denotes the set over all neighbors of node a .

(2) Adamic-Adar Index (AA)

Adamic and Adar propose the AA index that computes the similarity between two web pages [47]. The AA emphasizes less-connected common neighbors. It is defined as

$$S_{ab}^{\text{AA}} = \sum_{c \in n(a) \cap n(b)} \frac{1}{\log k(c)}, \quad (\text{S2})$$

where $k(c) = |n(c)|$ is the degree of node c .

(3) Resource Allocation Index (RA)

Motivated by the physical process of resource allocation, Zhou *et al.* propose the RA index that puts penalties to large degree nodes [48]. The RA is defined as

$$S_{ab}^{\text{RA}} = \sum_{c \in n(a) \cap n(b)} \frac{1}{k(c)}. \quad (\text{S3})$$

(4) Salton Index (Salton)

The Salton index is also called cosine similarity [31], which is defined as

$$S_{ab}^{\text{Salton}} = \frac{|n(a) \cap n(b)|}{\sqrt{k(a) \times k(b)}}. \quad (\text{S4})$$

(5) Sørensen Index (SI)

The SI is usually used in ecological science [40], which is defined as

$$S_{ab}^{\text{SI}} = \frac{2 \times |n(a) \cap n(b)|}{k(a) + k(b)}. \quad (\text{S5})$$

(6) Hub Promoted Index (HPI)

The HPI aims to measure the degree of topological overlap between two nodes in metabolic networks [7], which is defined as

$$S_{ab}^{\text{HPI}} = \frac{|n(a) \cap n(b)|}{\min(k(a), k(b))}. \quad (\text{S6})$$

(7) Hub Depressed Index (HDI)

The HDI is similar to the HPI. The difference is that HDI emphasizes the role of nodes with large degrees [48]. It is defined as

$$S_{ab}^{\text{HDI}} = \frac{|n(a) \cap n(b)|}{\max(k(a), k(b))}. \quad (\text{S7})$$

(8) Leicht-Holme-Newman Index (LHN-I)

Leicht *et al.* propose the LHN-I, which assigns a high value to the node pair with many common neighbors [49]. It is defined as

$$S_{ab}^{\text{LHN-I}} = \frac{|n(a) \cap n(b)|}{k(a) \times k(b)}. \quad (\text{S8})$$

(9) Jaccard Index (Jaccard)

The Jaccard directly normalizes the number of common neighbors [33]. It is defined as

$$S_{ab}^{\text{Jaccard}} = \frac{|n(a) \cap n(b)|}{|n(a) \cup n(b)|}. \quad (\text{S9})$$

A family of indexes based on path. We consider 5 indexes gauging the path feature. To reduce the computational complexity and to unify the path feature, we focus on path length less than or equal to 4.

(1) Local Path Index (LP)

Motivated by the CN index, Lü *et al.* propose the LP index [50]. A large number of studies have shown that the predictive ability of the LP index is stronger than that of the CN on many real networks [23, 25, 26]. The definition of LP is

$$S_{ab}^{\text{LP}} = A_{ab}^2 + \beta A_{ab}^3 + \beta^2 A_{ab}^4, \quad (\text{S10})$$

where the A_{ab}^i is the number of paths of length i that links node a and b , and β controls the weight of paths with different lengths. In this study, we use $\beta = 0.02$.

(2) Katz Index (Katz)

The Katz can be regarded as an ensemble method that directly sums the number of all paths [53]. It is defined as

$$S_{ab}^{\text{Katz}} = \beta^2 A_{ab}^2 + \beta^3 A_{ab}^3 + \beta^4 A_{ab}^4, \quad (\text{S11})$$

where β controls the weight of all paths. Katz is similar to LP. The difference between them is that Katz assigns exponentially decaying weights into long paths whereas the decaying weights in LP are slower. In this study, we use $\beta = 0.02$.

(3) FriendLink Index (FL)

The basic idea of the FL index is that two people that have more and shorter paths are more likely to become friends on social networks [51]. It is defined as

$$S_{ab}^{\text{FL}} = \sum_{i=2}^l \frac{1}{i-1} \cdot \frac{|paths_{ab}^i|}{\prod_{j=2}^i (N-j)}, \quad (\text{S12})$$

where N is the number of nodes in a network, l is the length of the longest path between nodes a and b . The $|paths_{ab}^i|$ is the number of paths with length i between nodes a and b .

(4) Relation Strength Similarity Index (RSS)

The RSS proposed by Chen *et al.* is to measure the relative degree of similarity between two nodes [52]. The relation strength is defined as

$$R(ab) = \begin{cases} \frac{\alpha_{ab}}{\sum_{x \in n(a)} \alpha_{ax}} & \text{if } a \text{ and } b \text{ are adjacent} \\ 0 & \text{otherwise,} \end{cases}$$

where α_{ab} is the weight between nodes a and b , which can be any value. Here, the $R(ab)$ is not symmetric, *i.e.* $R(ab) \neq R(ba)$. To make arbitrary nodes available, Chen *et al.* also propose the generalized relation strength as

$$R_{pl}^*(ac) = \prod_{k=1}^{K-1} R(b_k b_{k+1}),$$

where the p_l is a set of paths between nodes a and c . The p_l is formed by b_1, b_2, \dots, b_K in which the b_1 represents node a and the b_K represents node c . To make the calculation tractable, Chen *et al.* control the path length less than r , yielding

$$R_{pl}^*(ac) = \begin{cases} \prod_{k=1}^K R(b_k b_{k+1}) & \text{if } K \leq r \\ 0 & \text{otherwise.} \end{cases}$$

Taken together, if there are P_l paths with length shorter than r from node a to node b , the RSS is defined as

$$S_{ab}^{\text{RSS}} = \sum_{l=1}^{P_l} R_{p_l}^*(ab). \quad (\text{S13})$$

(5) Shortest Path Length Index (SPL)

Ran *et al.* show that the shorter the shortest path length between two unconnected nodes is, the more likely they are to form a link in the long-path networks [53, 54]. The SPL is defined as

$$S_{ab}^{\text{SPL}} = \frac{1}{d_{ab} - 1}, \quad (\text{S14})$$

where the d_{ab} is the length of the shortest path between nodes a and b . As the path length is limited to be not greater than 4, d_{ab} in this study falls within the range [2, 4]. If there is no path within length 4 connecting node a and b , $d_{ab} \rightarrow \infty$ which gives $S_{ab}^{\text{SPL}} = 0$.

A family of indexes based on path of length three. Different from the path feature that includes connections with different lengths, the feature path of length three only considers connection with lengths equal to three. Therefore, the path of length three is different from the path feature. We consider 2 indexes that gauge the feature path of length three.

(1) Paths of length three Index (L3)

Kovacs *et al.* propose a degree-normalized index based on paths of length three (L3). L3 is found to have a remarkable advantage compared with the indexes based on common neighbor in predicting protein-protein interactions [12]. The L3 index is defined as

$$S_{ab}^{\text{L3}} = \sum_{uv} \frac{l_{au} l_{uv} l_{vb}}{\sqrt{k(u)k(v)}}, \quad (\text{S15})$$

where nodes u and v are the intermediate node on a path of length 3 linking nodes a and b . The $l_{au} = 1$ if node a and node u have a link, and $l_{au} = 0$ otherwise.

(2) L3-based Cannistraci-Hebb Index (CH2-L3)

Motivated by the L3 [12], Muscoloni *et al.* adopt the Cannistraci-Hebb network automaton model to extend the local community paradigm to paths of length three [56]. Denote P_{ab} by the set of nodes on all paths of length 3 that connects node a and b . The CH2-L3 is defined as

$$S_{ab}^{\text{CH2-L3}} = \sum_{uv} \frac{l_{au} l_{uv} l_{vb} \sqrt{(1 + i(u))(1 + i(v))}}{\sqrt{(1 + e(u))(1 + e(v))}}, \quad (\text{S16})$$

where $i(u)$ is the number of internal links between node u and nodes except nodes a and b in P_{ab} , $e(u)$ is the number of external links between node u and nodes not in P_{ab} .

A family of indexes based on heterogeneity. We consider 2 indexes gauging the heterogeneity feature.

(1) Heterogeneity Index (HEI)

Shang *et al.* find that the indexes based on common neighbor fail to identify missing links in the tree-like networks [55]. To solve this problem, they take advantage of network heterogeneity and propose the heterogeneity index (HEI). The HEI is defined as

$$S_{ab}^{\text{HEI}} = |k(a) - k(b)|^\beta, \quad (\text{S17})$$

where β is a free heterogeneity exponent. In this work, we set $\beta = 0.02$.

(2) Homogeneity Index (HOI)

Based on the assumption that the network homogeneity plays a major role in homogeneous networks [55], Shang *et al.* define HOI as

$$S_{ab}^{\text{HOI}} = \frac{1}{|k(a) - k(b)|^\beta}. \quad (\text{S18})$$

In this work, we set $\beta = 0.02$.

Index classification. From description of the 18 indexes, we can intuitively link the index with the associated topological feature. To further validate the index classification, we further consider the following property.

If two indexes are associated with the same feature, they will have the same p_1 and p_2 value for the given L^P and L^N .

p_1 is the percentage of samples in L^P that hold the topological feature, and p_2 is the percentage of samples in L^N that hold the topological feature. Technically, p_1 and p_2 can be measured by counting the number of samples whose index value is 0. From the definition of those indexes, it is obvious that for indexes in the family of common neighbor feature, they all give $S_{ab} = 0$ for two nodes that do not share any common neighbor. For indexes in the family of path feature, they all give $S_{ab} = 0$ for two nodes not connected by a path (within length 4). For indexes in the family of path of length three feature, they all give $S_{ab} = 0$ for two nodes not connected by a path of length 3. For indexes in the family of heterogeneity feature, they all give $S_{ab} = 0$ for two nodes with the same degree. We empirically calculate

p_1 and p_2 values by each index and check if they are the same for indexes classified into the four categories. The numerical p_1 and p_2 values confirms the classification.

S2. METHODS AND MATERIALS

Here, we describe in detail the methods for the link prediction problem, especially the sampling strategy of the positive set L^P (missing links) and the negative set L^N (nonexistent links). As real networks are usually very sparse ($L \ll \frac{N(N-1)}{2}$ for an undirected network), link prediction itself is a sample imbalance problem. Hence, different sampling methods for selecting the L^P and L^N are proposed [29, 37, 48, 50, 55, 57].

The unsupervised approach is based on prior or heuristic knowledge that the index value directly gives the likelihood that missing links should be truly linked [29, 48, 50, 55]. The most common process is to randomly remove 10% of L links from the original network, which consequently compose the positive set L^P [29, 37, 55, 57]. The negative set L^N can be constructed by all nonexistent links in the original network. The size of L^N under this choice is $\frac{N(N-1)}{2} - L$. However, for big networks, this choice of L^N is space and computationally costly. Therefore, another choice is to randomly select node pairs from all nonexistent links under the size constraint $|L^P| = |L^N|$ [25, 53, 54, 66].

The supervised approach usually learns the appropriate score from the input index values. The sample imbalance has a huge impact on the learning results when an algorithm learns a function from the given data. Hence, a balance between positive and negative samples is often needed [43, 59]. However, recent studies also use imbalanced samples [38, 39]. The authors randomly remove 20% of L links from the original network and compute the index value from the rest 80% of links [39]. To train the classifier, they take 80% of the removed links (16% of L links) as the positive set L^P and take all nonexistent links as the negative set L^N .

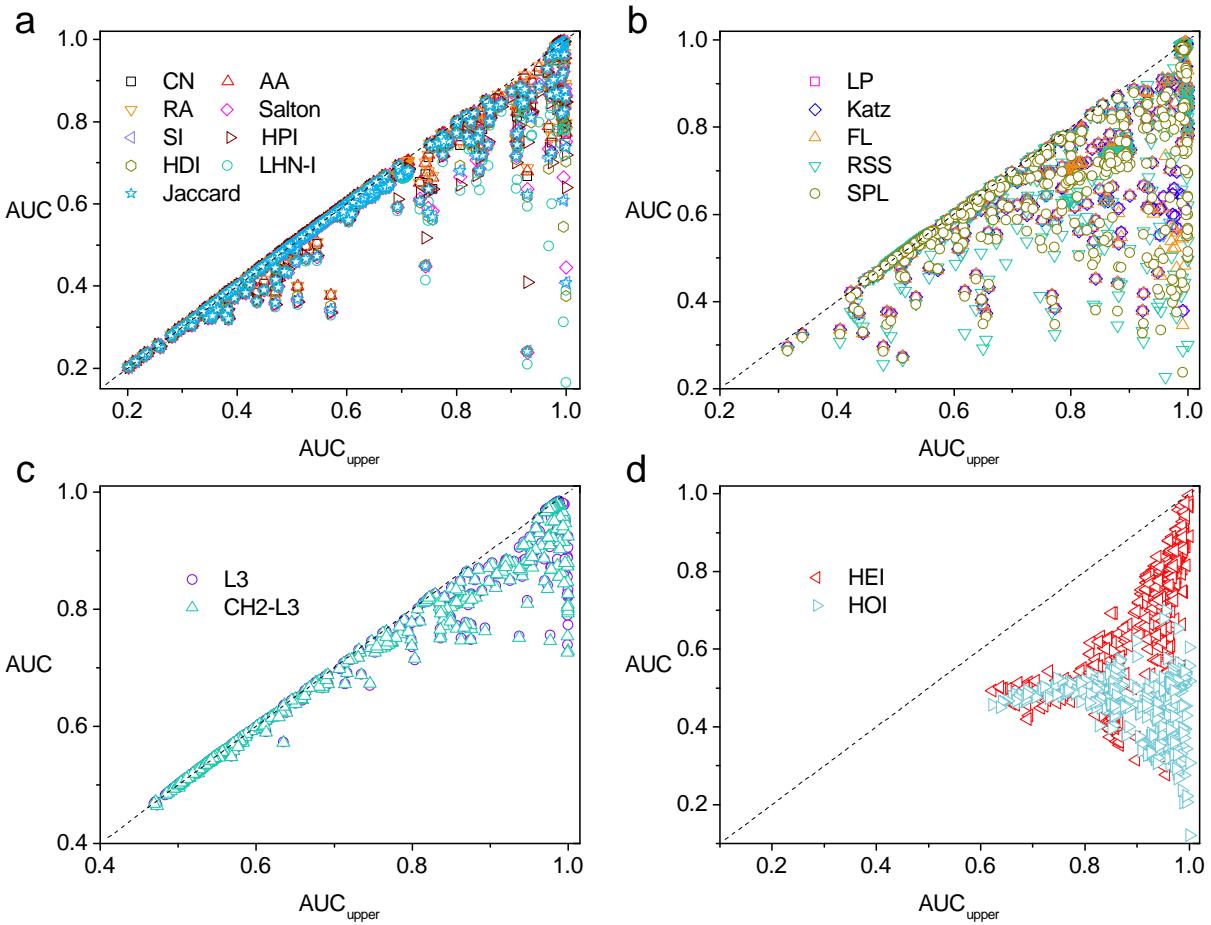
In the main text, we randomly remove 20% of L links from the original network according to the sampling method in Ref [38, 39]. We adopt the $|L^P| = |L^N|$ to balance the positive and negative samples. Moreover, to reasonably compare the performance between the unsupervised and supervised prediction, we use the same testing set. For the supervised approach, to train a classifier, we take 50% of the removed links as the positive set and randomly select node pairs with the same size as the positive set from all nonexistent links as the negative set. The testing set is constructed by using the other 50% of the removed links as the positive set, and another randomly selected nonexistent link set as the negative set. The positive and negative sets always have the same size. Finally, we use the Random

Forest classifier to learn a mapping function. To find the most optimal mapping function, we take the grid search method [67] to optimize the AUC by choosing the best parameters of the Random Forest classifier on each training set.

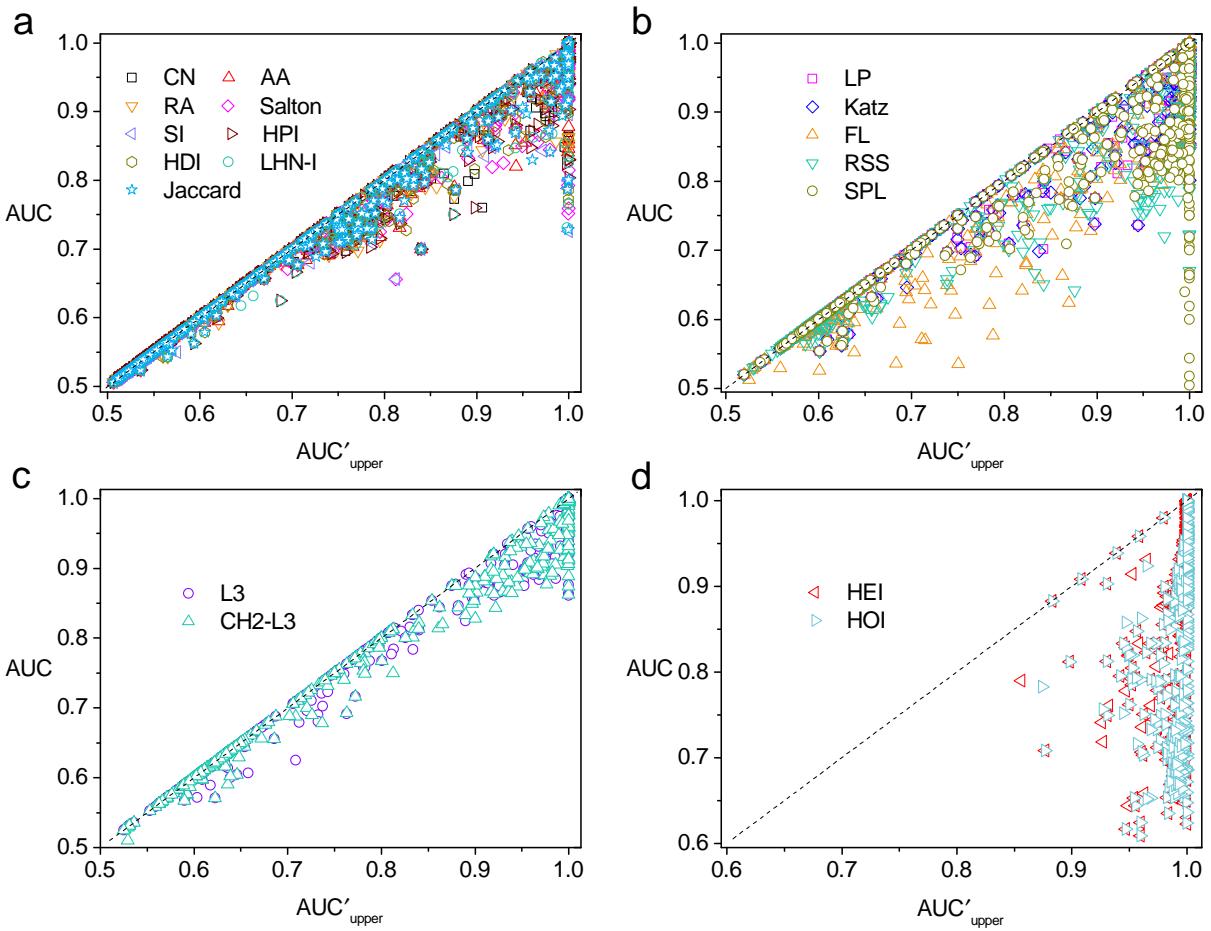
To make the assessment more general, we also evaluate the other two alternative sampling methods. We call the first sampling method *sample1* in the following discussion. *Sample1* uses balanced positive and negative samples ($|L^P| = |L^N|$). It differs from the sampling method in the main text for the size of the training and testing set. In particular, *sample1* randomly removes 20% of L links as the missing links. 80% of the removed links (16% of L links) are used as the positive set in the training and the rest of 20% of the removed links are used as the positive set in the testing. In both training and testing, randomly selected nonexistent links compose the negative set with the same size as the positive set ($|L^P| = |L^N|$). The second sampling method is called *sample2* in the following discussion. *Sample2* uses imbalanced positive and negative samples. Specifically, *sample2* randomly removes 20% of L links as the missing links. In the training step, the L^P is composed of 80% of the removed links (16% of L links), and the L^N is composed of 80% of all nonexistent links. In the testing step, the L^P is composed of the rest 20% of the removed links, and the L^N is composed of the rest 20% of all nonexistent links.

In each of the sampling methods, we generate 200 pairs of L^P and L^N sets by randomly selecting existing and nonexistent links in each network, respectively. Unless otherwise specified, we use the average value of 200 samples for the results of the unsupervised approach in all figures and tables. For the supervised results, we use the highest AUC as the prediction performance of each index.

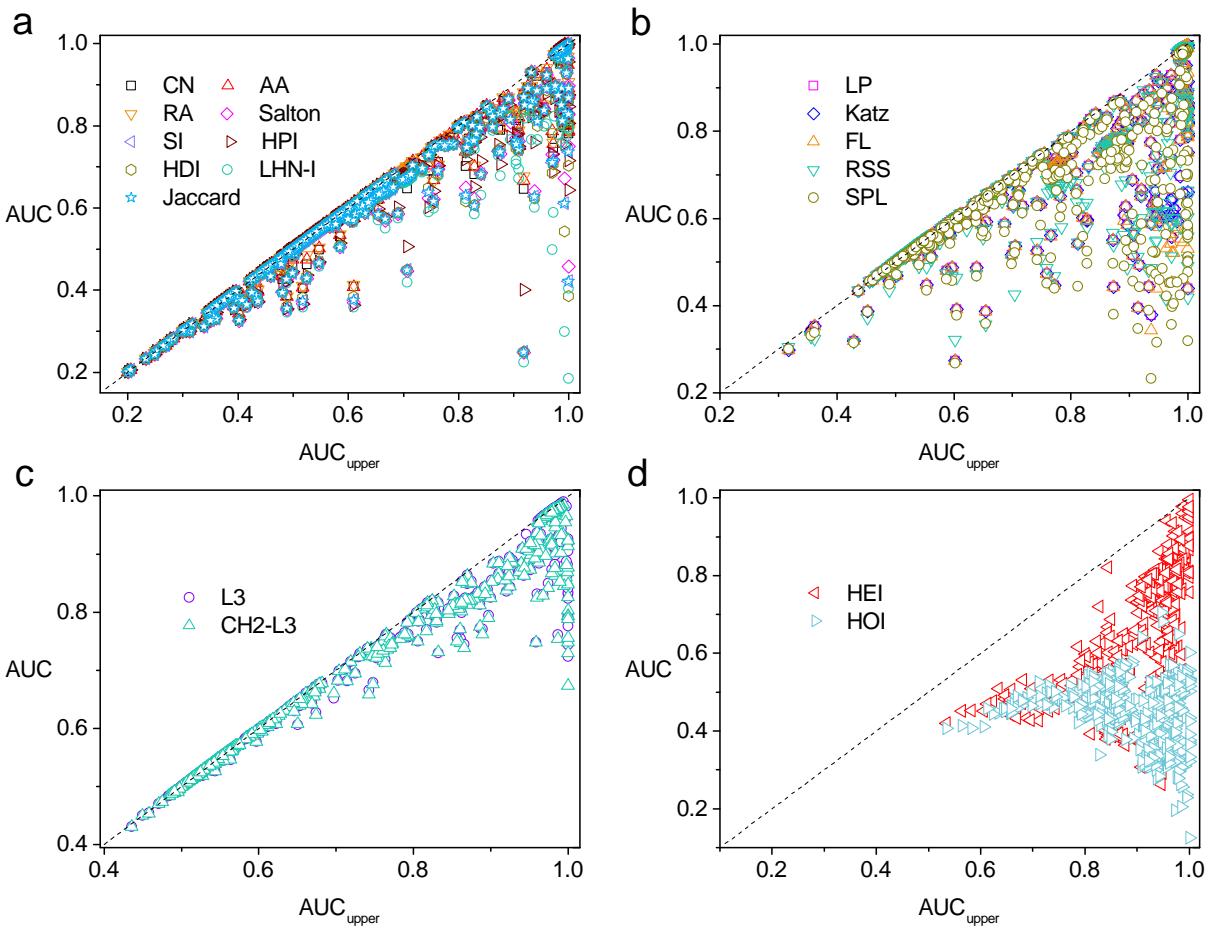
To make sure our results are not affected by a specific sampling method, we perform a **robustness check** by repeating the measurement in the main text using *sample1* (Fig. S6 and Fig. S7) and *sample2* (Fig. S8 and Fig. S9). The same capability applies to different sampling methods, supporting the universality of the conclusion drawn.



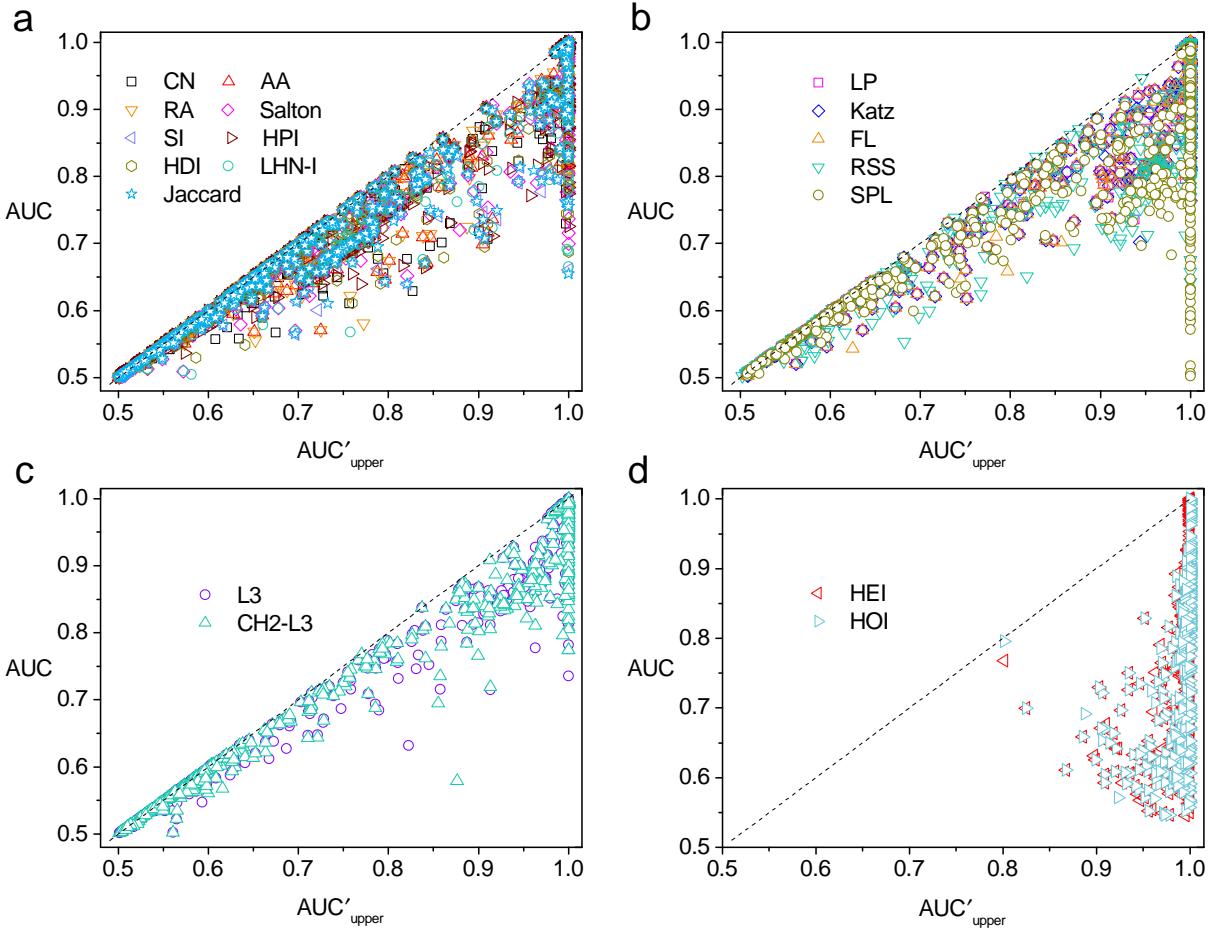
Supplementary Figure S6: Unsupervised prediction based on balanced positive and negative samples by *sample1*. The same quantitative analysis in the Fig. S2 is repeated.



Supplementary Figure S7: Supervised prediction based on balanced positive and negative samples by *sample1*. The same quantitative analysis in the Fig. S3 is repeated.



Supplementary Figure S8: Unsupervised prediction based on imbalanced positive and negative samples by *sample2*. The same quantitative analysis in the Fig. S2 is repeated.



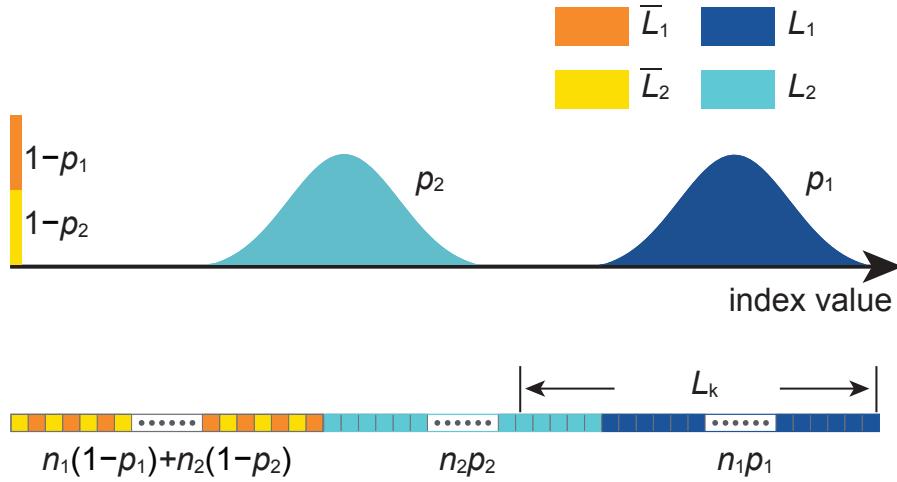
Supplementary Figure S9: Supervised prediction based on imbalanced positive and negative samples by *sample2*. The same quantitative analysis in the Fig. S3 is repeated.

S3. A FEATURE'S MAXIMUM CAPABILITY MEASURED BY PRECISION

Besides AUC, precision can also be used to evaluate the quality of prediction. The precision measures the percentage of the correct prediction (node pairs indeed in L^P) among the top-k predicted candidates [30, 31, 40, 44, 45]. After ranking the node pairs in both L^P and L^N according to their scores in descending order, we select L_k node pairs with the highest score. The precision is given as

$$\text{Precision} = \frac{L_r}{L_k}, \quad (\text{S19})$$

where L_r is the number of selected node pairs that are included in L^P .



Supplementary Figure S10: The maximum capability of a feature measured by precision in the unsupervised approach. In the best index value ranking illustrated in Fig. 1c of the main text, L_1 is ranked ahead of L_2 . Therefore, when ranking node pairs in descending order of their index values, we have three segments in the rank list. The first is $n_1 p_1$ entities of L_1 , followed by $n_2 p_2$ entities of L_2 . The $n_1(1 - p_1) + n_2(1 - p_2)$ entities of $\bar{L}_1 \cup \bar{L}_2$, which all have the same index value, are ranked at the end. Here, $n_1 = |\bar{L}_1 \cup L_1| = |L^P|$ and $n_2 = |\bar{L}_2 \cup L_2| = |L^N|$. Depending on the choice of L_k , the $\text{Precision}_{\text{upper}}$ for the best index value ranking can be derived.

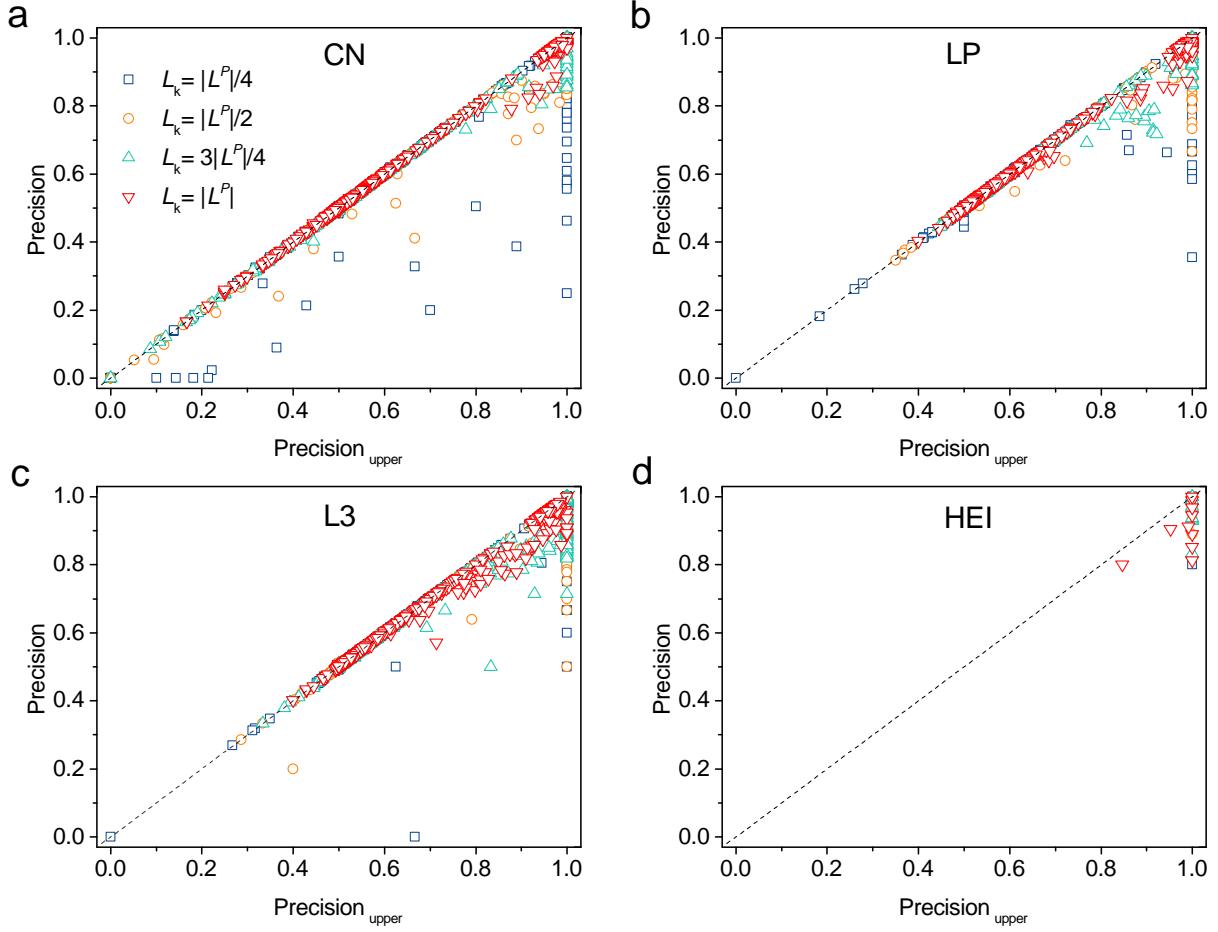
Let us first consider the best index value ranking in the unsupervised approach (Fig. 1c presented in the main text and Fig. S10). Since Eq. (S19), in which the lowest index value of L_1 is greater than the highest index value of L_2 . Assume $n_1 = |\bar{L}_1 \cup L_1| = |L^P|$ and $n_2 = |\bar{L}_2 \cup L_2| = |L^N|$ for the number of node pairs in L^P and L^N . When ranking node pairs in descending order of their index values, we have three segments in the ranking list. The first is $n_1 p_1$ entities of L_1 , followed by $n_2 p_2$ entities of L_2 . The $n_1(1 - p_1) + n_2(1 - p_2)$ entities of $\bar{L}_1 \cup \bar{L}_2$, which all have the same index value, are ranked at the end. Once the

rank list is known, the $\text{Precision}_{\text{upper}}$ for the best index value ranking only depends how L_k cuts the top entities of the list, which can be formulated as

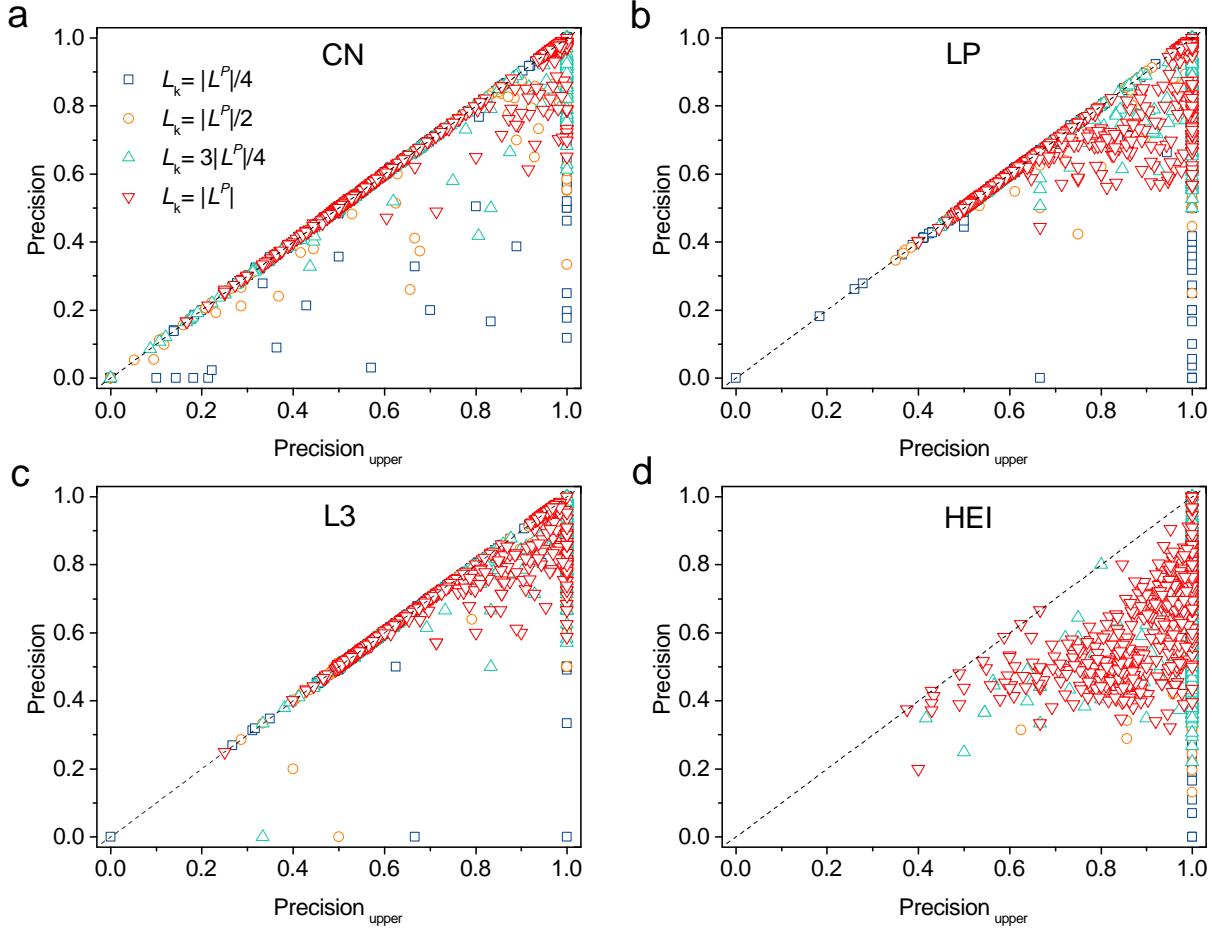
$$\text{Precision}_{\text{upper}} = \begin{cases} 1, & n_1 p_1 \geq L_k, \\ \frac{n_1 p_1}{L_k}, & n_1 p_1 < L_k \leq n_1 p_1 + n_2 p_2, \\ \frac{n_1 p_1 + \frac{n_1(1-p_1)(L_k - n_1 p_1 - n_2 p_2)}{n_1(1-p_1) + n_2(1-p_2)}}{L_k}, & n_1 p_1 + n_2 p_2 < L_k \leq n_1 + n_2. \end{cases} \quad (\text{S20})$$

Note that node pairs in $\bar{L}_1 \cup \bar{L}_2$ have the same index value. The relative position of one node pair among all $n_1(1-p_1) + n_2(1-p_2)$ entities is random. In Eq. (S20), $\frac{n_1(1-p_1)}{n_1(1-p_1) + n_2(1-p_2)}$ corresponds to the probability of finding a missing link in $\bar{L}_1 \cup \bar{L}_2$. Therefore, $\frac{n_1(1-p_1)(L_k - n_1 p_1 - n_2 p_2)}{n_1(1-p_1) + n_2(1-p_2)}$ is the expected number of missing links for a given L_k value.

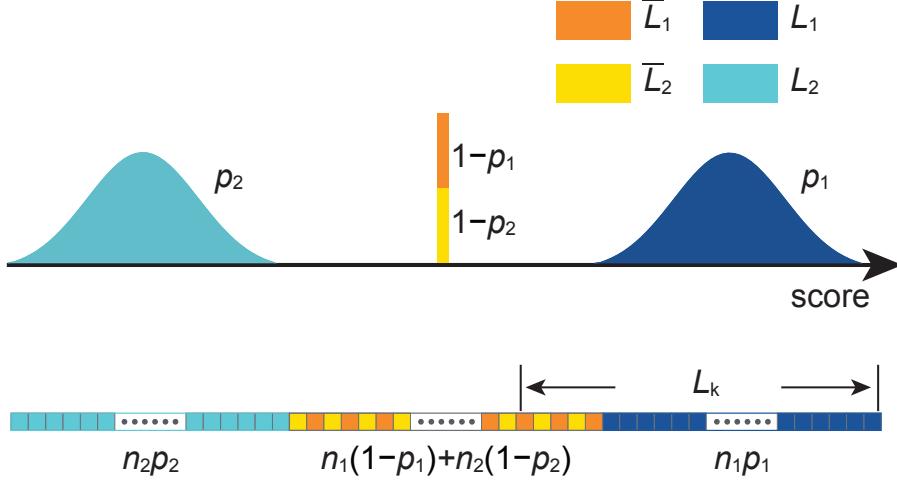
The above deduction suggests that under the best index value ranking, the measured precision should exactly follow $\text{Precision}_{\text{upper}}$ given by Eq. (S20). To test it, we select one index from each family and identify networks in which the unsupervised prediction by this index has an AUC value greater than 95% of $\text{AUC}_{\text{upper}}$. For such networks, we could approximate that the index value ranking is close to the best scenario. We then quantify the performance of the unsupervised prediction using precision. In these networks, the measured precision well follows what Eq. (S20) depicts (Fig. S11). Moreover, when measuring the prediction performance of this index in all 550 networks, the measured precision is below $\text{Precision}_{\text{upper}}$ (Fig. S12), supporting the claim that Eq. (S20) captures the maximum capability of a feature measured by precision in the unsupervised approach.



Supplementary Figure S11: We choose the index CN, LP, L3, and HEI from each of the four families. We select the networks in which the unsupervised prediction by this index is already close to the upper bound measured by AUC (measured AUC is more than 95% of AUC_{upper}). For such networks, it is expected that the performance of this index measured by precision should follow Eq. (S20). Indeed, for different choices of L_k ($L_k = |L^P|/4, |L^P|/2, 3|L^P|/4, |L^P|$), the precision measured is almost on the line $y = x$, supporting the theoretical prediction.



Supplementary Figure S12: We choose the index CN, LP, L3, and HEI from each of the four families and measure the performance of the unsupervised prediction by these indexes in all 550 networks. For different choices of L_k ($L_k = |L^P|/4, |L^P|/2, 3|L^P|/4, |L^P|$), the measured precision is equal to or below Precision_{upper}, supporting the claim that Precision_{upper} gives the maximum capability of a feature measured by precision.



Supplementary Figure S13: The maximum capability of a feature measured by precision in the supervised approach. The optimal score ranking after the mapping function is to have $\bar{L}_1 \cup \bar{L}_2$ ranked in between L_1 and L_2 , as illustrated in Fig. 1d of the main text. When ranking node pairs in descending order of their index values, we have three segments in the rank list. The first is n_1p_1 entities of L_1 , followed by $n_1(1-p_1) + n_2(1-p_2)$ entities of $\bar{L}_1 \cup \bar{L}_2$, which all have the same index value. n_2p_2 entities of L_2 are ranked at the end. Here, $n_1 = |\bar{L}_1 \cup L_1| = |L^P|$ and $n_2 = |\bar{L}_2 \cup L_2| = |L^N|$. Depending on the choice of L_k , the $\text{Precision}'_{\text{upper}}$ in the supervised approach can be derived.

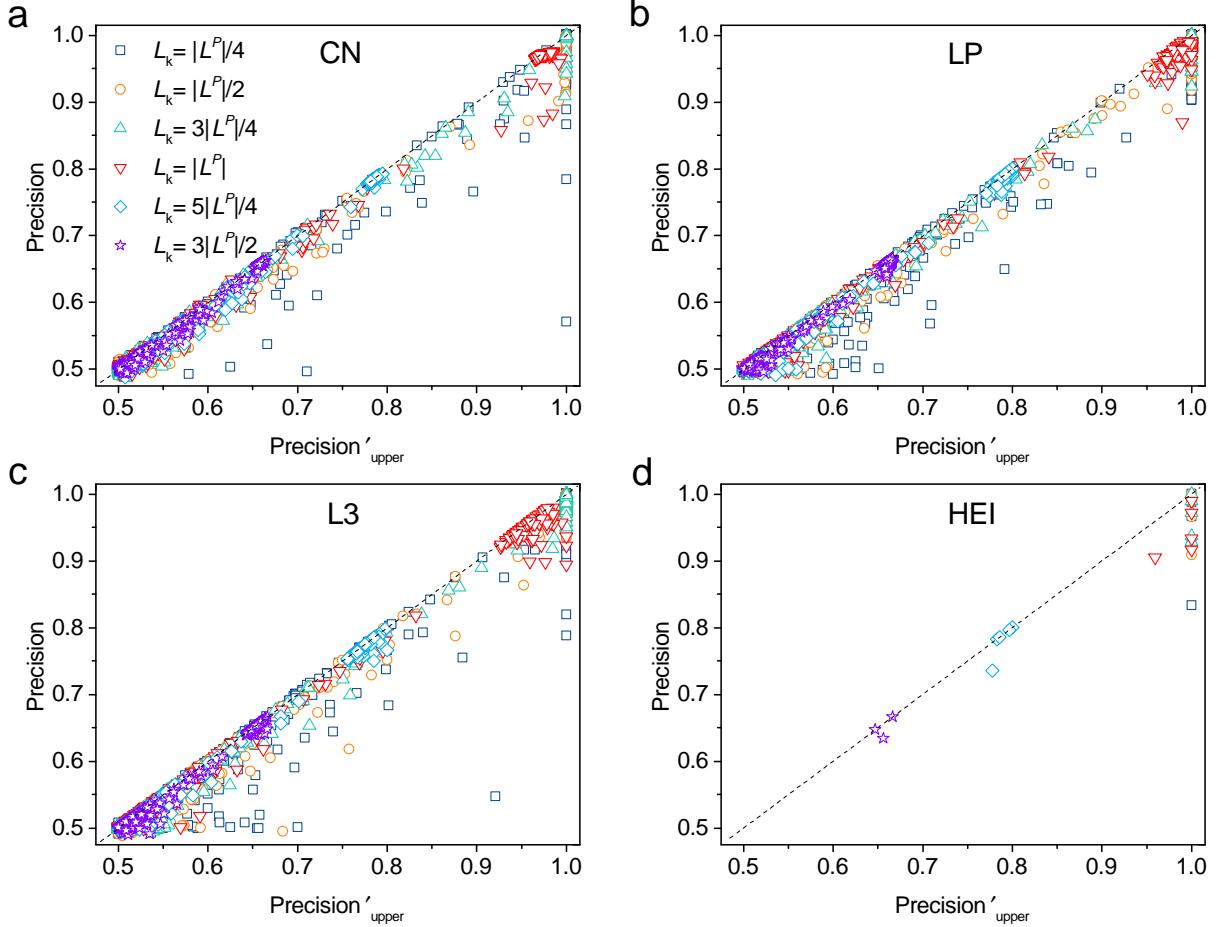
In the supervised approach, the mapping function can further optimize the score ranking from the index value ranking. As discussed in the main text, the optimal score ranking is such that L_1 ranks ahead, followed by $\bar{L}_1 \cup \bar{L}_2$, and L_2 ranks at the end (Fig. S13). Similar to Eq. (S20), we can derive the upper bound of supervised prediction measured by precision as

$$\text{Precision}'_{\text{upper}} = \begin{cases} 1, & n_1p_1 \geq L_k, \\ \frac{n_1p_1 + \frac{n_1(1-p_1)(L_k - n_1p_1)}{n_1(1-p_1) + n_2(1-p_2)}}{L_k}, & n_1p_1 < L_k \leq n_1 + n_2 - n_2p_2, \\ \frac{n_1}{L_k}, & n_1 + n_2 - n_2p_2 < L_k \leq n_1 + n_2, \end{cases} \quad (\text{S21})$$

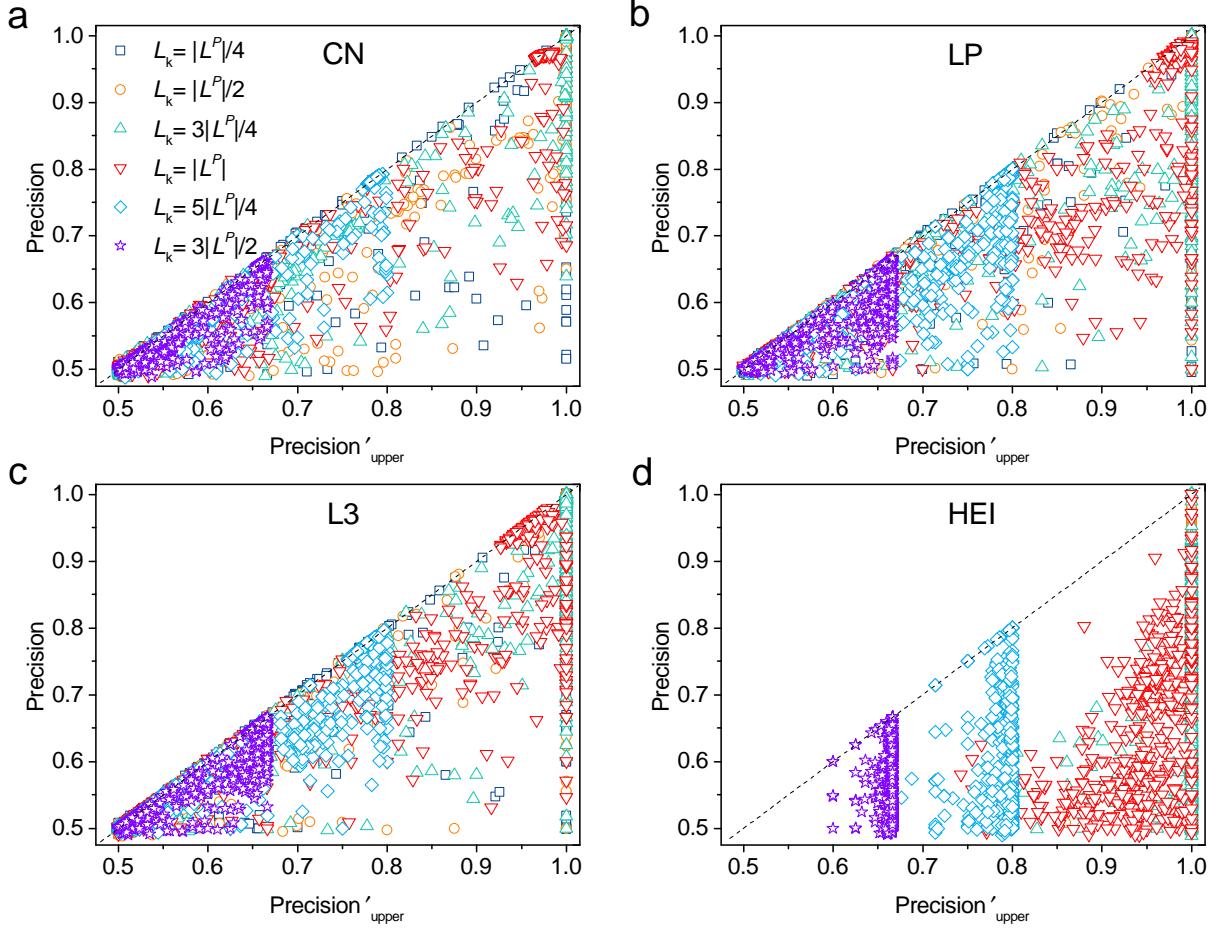
where $\frac{n_1(1-p_1)}{n_1(1-p_1) + n_2(1-p_2)}$ corresponds to the probability of finding a missing link in $\bar{L}_1 \cup \bar{L}_2$. It is noteworthy that the Eq. (S20) and Eq. (S21) still hold when the $L_k = |L^P|$ (where the precision is equivalent to the recall [29]).

We perform the same test for Eq. (S21) as for Eq. (S20). For networks in which an index gives a supervised prediction already close to the upper bound $\text{AUC}'_{\text{upper}}$, the prediction measured by precision is expected to follow Eq. (S21). This is confirmed in Fig. (S14). For all networks, the prediction measured by precision should be below what Eq. (S21) yields.

This is also confirmed in Fig. (S15).

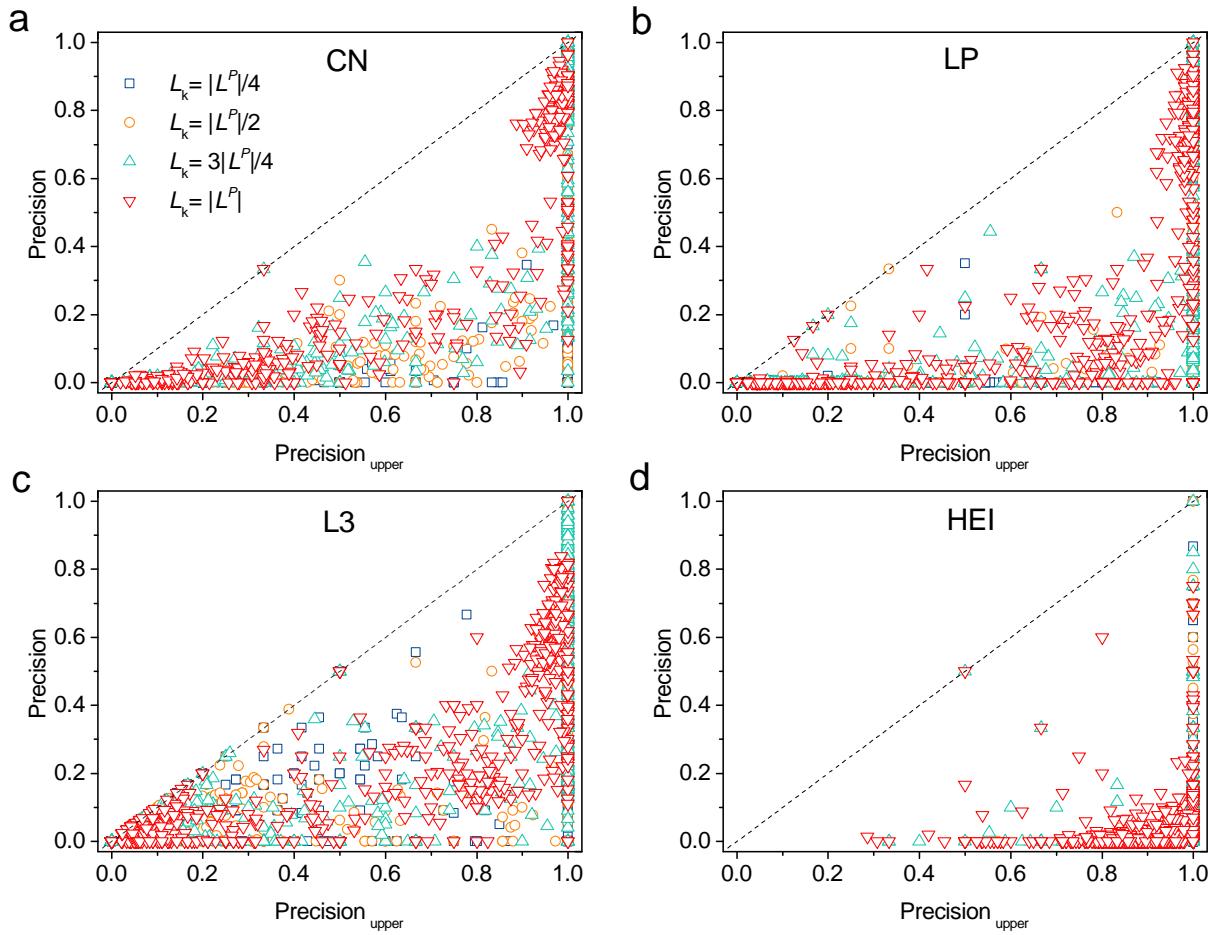


Supplementary Figure S14: We choose the index CN, LP, L3, and HEI from each of the four families. We select the networks in which the unsupervised prediction by this index is already close to the upper bound measured by AUC (measured AUC is more than 95% of AUC'_{upper}). For such networks, it is expected that the performance of this index measured by precision should follow Eq. (S21). Indeed, for different choices of L_k ($L_k = |L^P|/4, |L^P|/2, 3|L^P|/4, |L^P|$), the precision measured is almost on the line $y = x$, supporting the theoretical prediction.

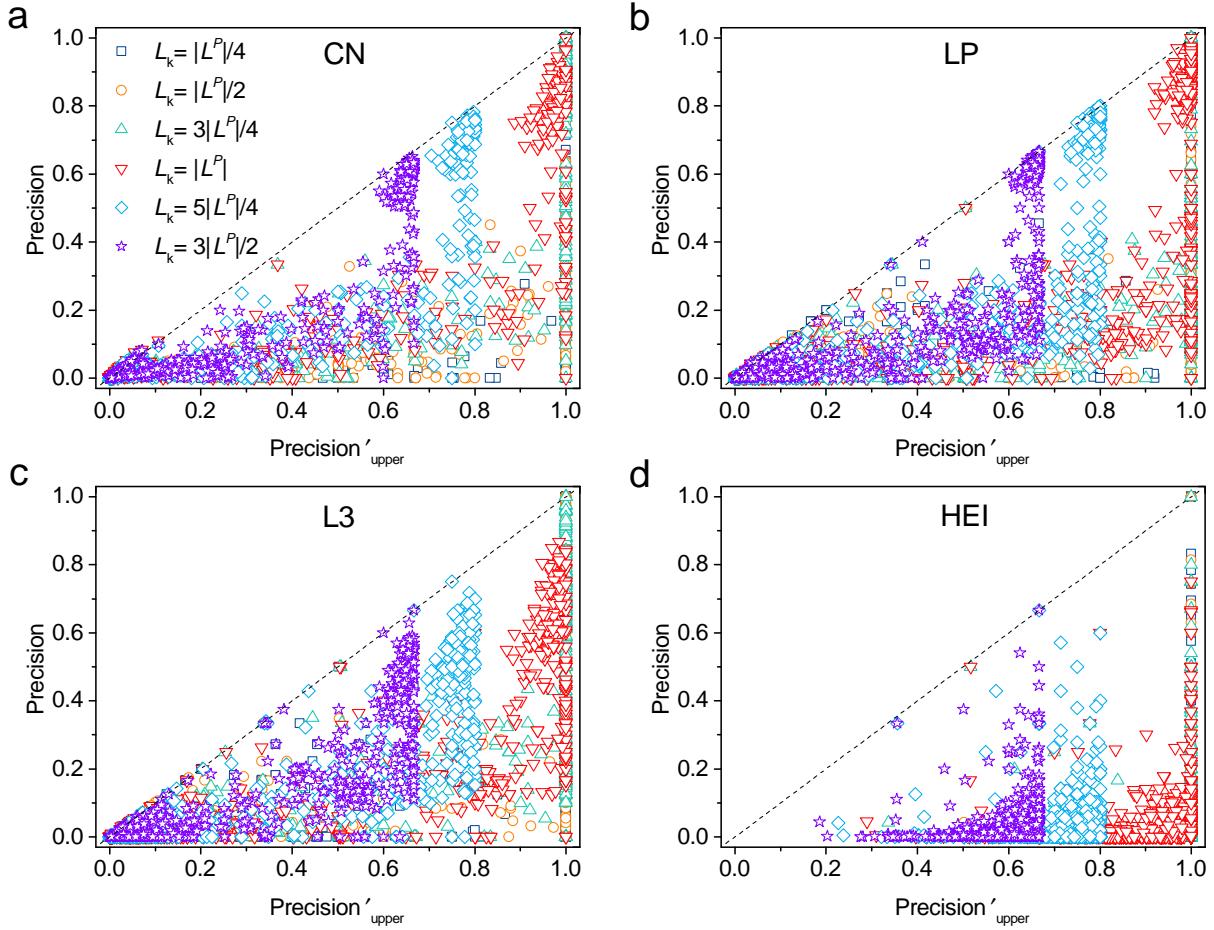


Supplementary Figure S15: We choose the index CN, LP, L3, and HEI from each of the four families and measure the performance of the supervised prediction by these indexes in all 550 networks. For different choices of L_k ($L_k = |L^P|/4, |L^P|/2, 3|L^P|/4, |L^P|$), the measured precision is equal to or below $\text{Precision}'_{\text{upper}}$, supporting the theoretical results for the maximum capability of a feature measured by precision.

Finally, some studies argue that the imbalanced samples extremely affect the results measured by precision [58, 59, 68]. For this reason, we test Eq. (S20) and Eq. (S21) under the imbalanced sampling *sample2*. The imbalanced sample indeed reduces the performance by precision (Fig. S16 and Fig. S17). But Eq. (S20) and Eq. (S21) still capture the upper bound of the performance, which correctly give the maximum capability of a topological feature. It is worth noting that the Precision'\$_{\text{upper}}\$ for supervised prediction will be lower than 0.5 when the positive set \$L^P\$ and the negative set \$L^N\$ are imbalanced (Fig. S17).



Supplementary Figure S16: Unsupervised prediction based on imbalanced positive and negative samples by *sample2*. The same quantitative analysis in the Fig. S12 is repeated.



Supplementary Figure S17: Supervised prediction based on imbalanced positive and negative samples by *sample2*. The same quantitative analysis in the Fig. S15 is repeated.

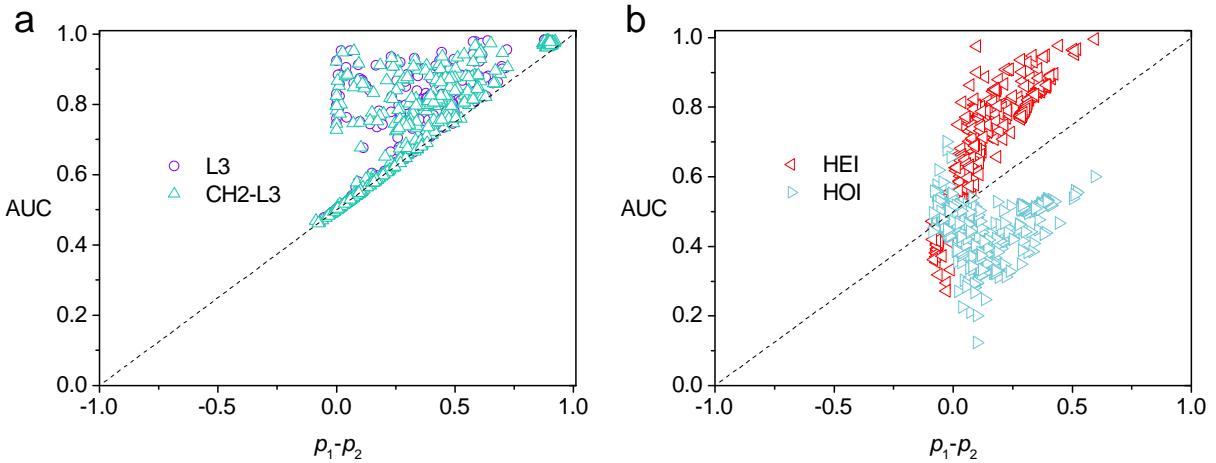
S4. EXTENDED DISCUSSION ON THE LOWEST INDEX VALUE

An index for a topological feature is designed to quantify the expression of this feature. Hence, the logic behind the index is that its value increases monotonically with the extent that the feature is expressed. For entities that do not hold the feature at all, they should have the same and the lowest value. If an entity that does not hold the feature has a higher index value than that does hold the feature, the index must be incorrectly designed. Likewise, if two entities that do not hold the feature have different index values, the index is also incorrectly designed.

Currently, there is no unified rule on what value should be the lowest. In the main text, we consider the lowest value to be zero. Indeed, all 18 indexes in this study assign the value 0 to entities that do not hold the feature they are designed to quantify. There could be exceptions. For example, one can design an index equal to the CN value plus 0.5. This is a valid index. But it is easy to see that our theoretical framework still holds when the lowest value is not zero. Indeed, the validity of based on the fact that entities that do not hold the feature are assigned the same and the lowest value, which is guaranteed by the design of the index.

S5. EXTENDED DISCUSSION ON THE SCALING OF THE UNSUPERVISED PREDICTION PERFORMANCE

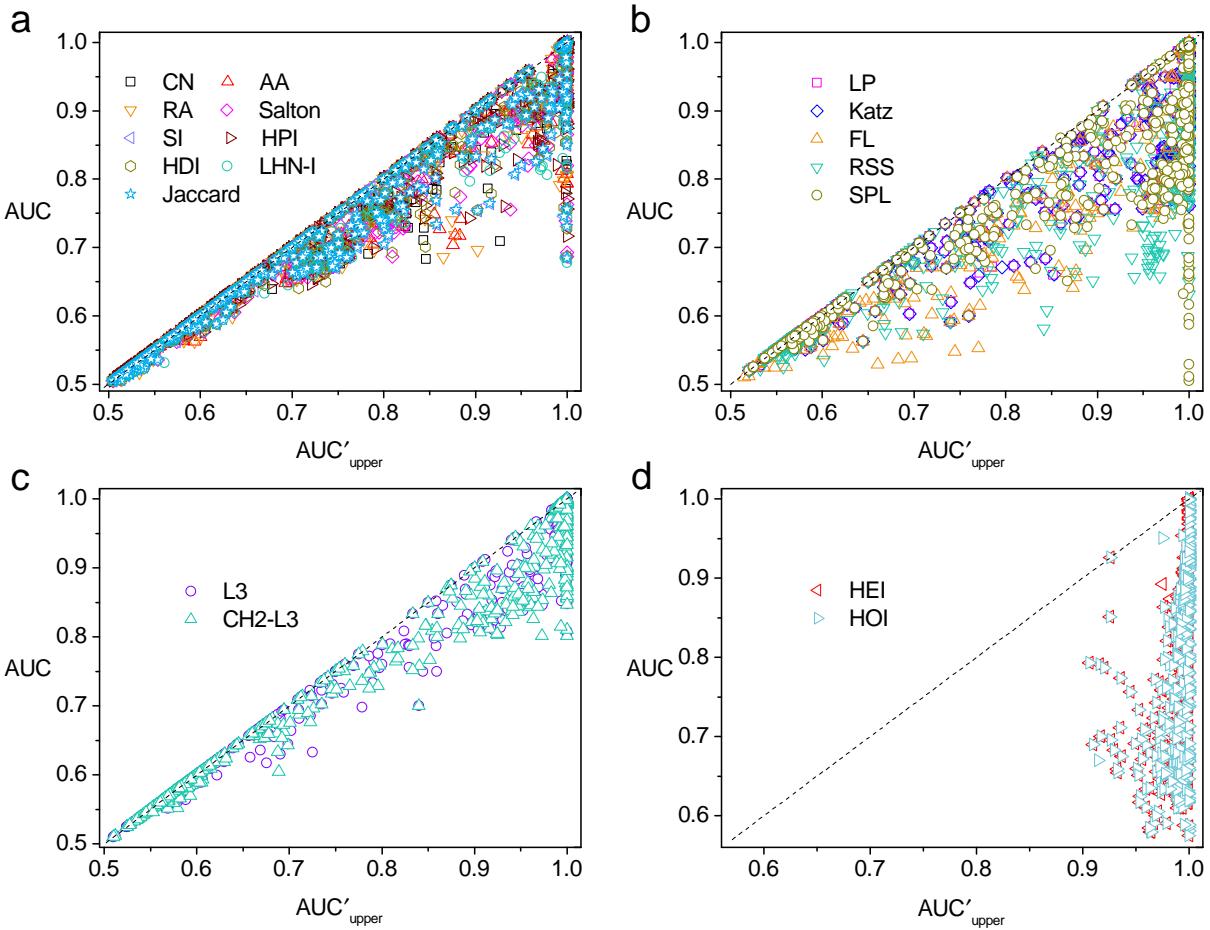
As illustrated in Eq. (3) and Eq. (4) of the main text, the gap between the upper and lower bound of the unsupervised prediction is $p_1 \times p_2$. Therefore, if the average $p_1 \times p_2$ is relatively small for a given feature, the performance of different indexes associated with it should roughly scale as $p_1 - p_2$. But the final scaling depends on how even the measured AUC data points distribute within the gap. In the main text, we show the scaling behavior of the common neighbor feature and path feature. For the path of length three feature, the average $p_1 \times p_2$ is smaller than that of path feature (Table S1). The scaling also holds (Fig. S18a). But because the AUC for the path of length three feature is closer to the upper bound compared with the path feature (Figs. S2b and c), the data points are systematically located in the upper corner. For the heterogeneity feature, the average $p_1 \times p_2$ is much larger (Table S1). Hence, the AUC does not follow $p_1 - p_2$.



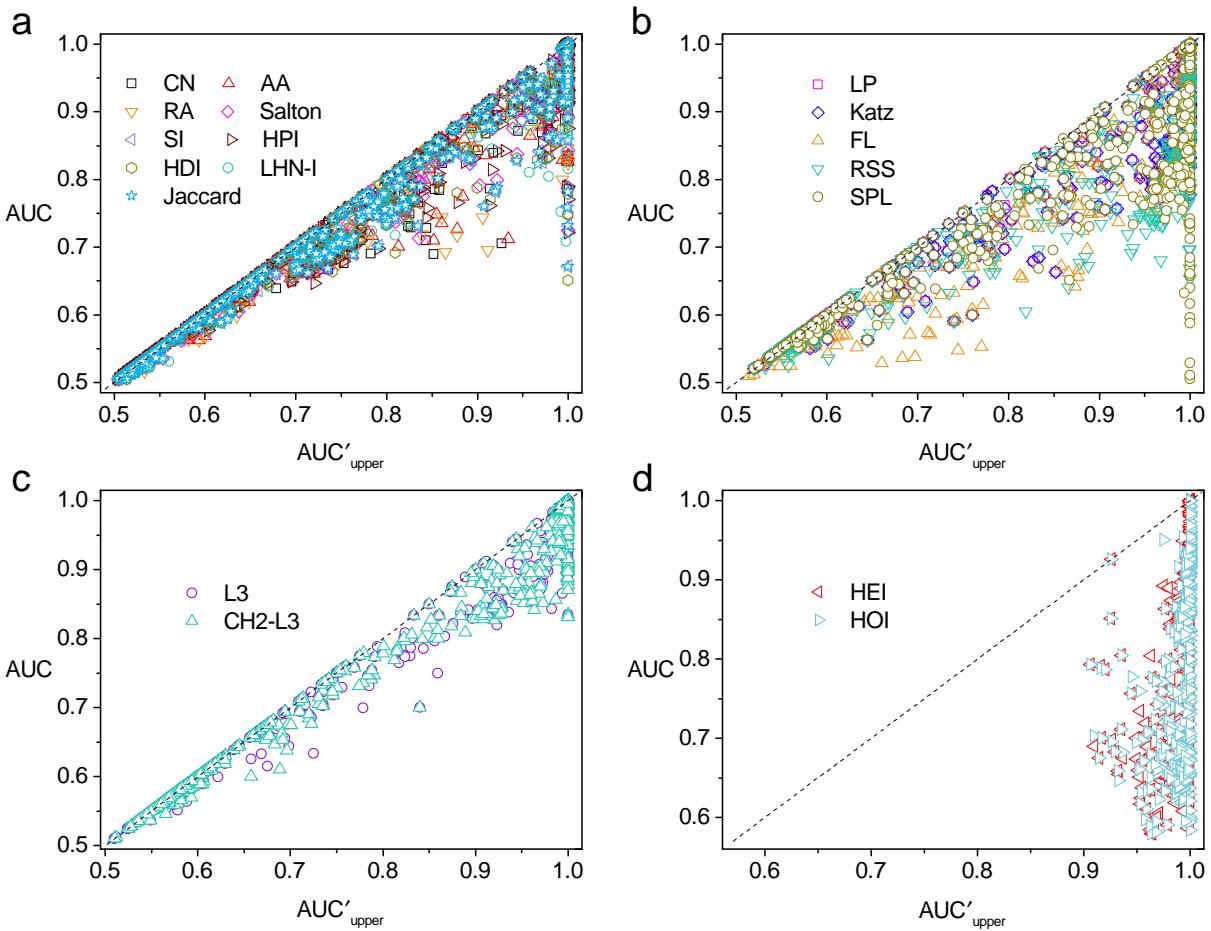
Supplementary Figure S18: The unsupervised prediction for indexes associated with a given feature and the $p_1 - p_2$ for the path of length three feature (a) and the heterogeneity feature (b). The path of length three feature has a relatively small average $p_1 \times p_2$ value. The data points roughly scales as $p_1 - p_2$. For the heterogeneity feature, the average $p_1 \times p_2$ value is large. Data points do not collapse to the line $y = x$.

S6. TEST ON OTHER MACHINE LEARNING ALGORITHMS

In the main text, we use the Random Forest classifier to analyze the capability of a feature. Here, we also consider Gradient Boosting and AdaBoost classifiers to display similar results. The sampling method keeps the same as that of the main text. The results from Figs. S19 and S20 show that our quantitative framework is not affected by the machine learning algorithm. Hence, this further indicates that the machine learning algorithm can find the optimal mapping function and improve the prediction compared with that of the unsupervised prediction.

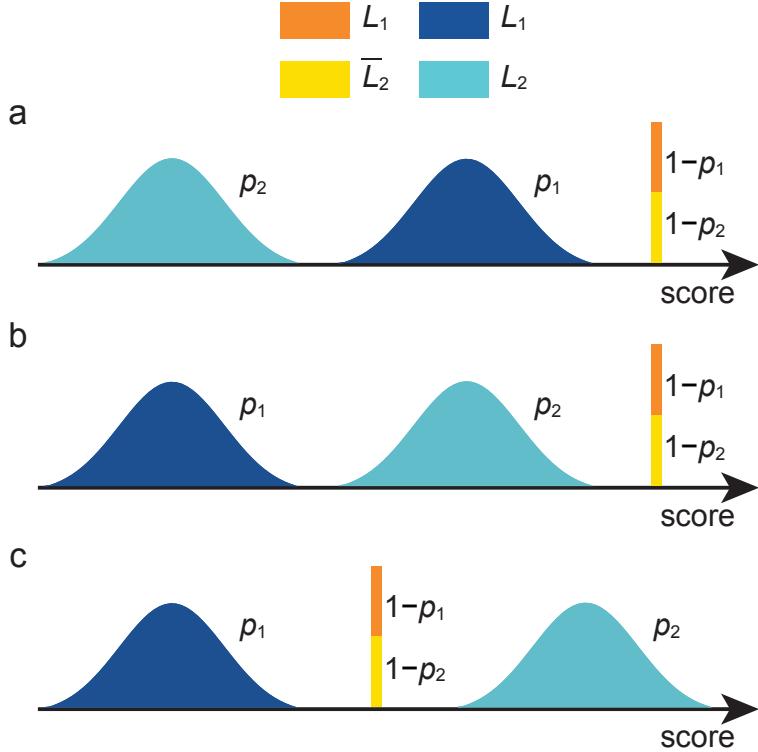


Supplementary Figure S19: Supervised prediction based on the Gradient Boosting classifier. The same quantitative analysis in the Fig. S3 is repeated.



Supplementary Figure S20: Supervised prediction based on the AdaBoost classifier. The same quantitative analysis in the Fig. S3 is repeated.

S7. THE OPTIMAL SCORE RANKING



Supplementary Figure S21: An illustration of different link prediction performance. It is used as a supplement to prove the most optimal score ranking. The variables applied here are the same as those in Fig. 1 of the main text: p_1 and p_2 represent the fraction of missing links and nonexistent links in which each node pair has an index value greater than 0, respectively.

The link prediction performance relies on the relative rank of the three sets: L_1 , L_2 and $\bar{L}_1 \cup \bar{L}_2$. Totally there will be 6 different rankings. In Figs. 1b, 1c and 1d of the main text, we show the results based on 3 rankings (in descending order of the scores/values): $L_2 > L_1 > \bar{L}_1 \cup \bar{L}_2$, $L_1 > L_2 > \bar{L}_1 \cup \bar{L}_2$, and $L_1 > \bar{L}_1 \cup \bar{L}_2 > L_2$. Here we show the other 3 different permutations in Fig. S21 and calculate the AUC value in these ranking scenarios.

For the ranking in Fig. S21a, we have $\bar{L}_1 \cup \bar{L}_2 > L_1 > L_2$. A positive sample outscores a negative sample only when one node pair is from $L_1 \cup \bar{L}_1$ (the whole set L^P), and the other is from L_2 . Hence, this gives

$$\text{AUC}_{s1} = \frac{n'}{n} + \frac{1}{2} \frac{n''}{n} = p_2 + \frac{1}{2}(1 - p_1)(1 - p_2). \quad (\text{S22})$$

For the ranking in Fig. S21b, we have $\bar{L}_1 \cup \bar{L}_2 > L_2 > L_1$. A positive sample outscores a negative sample only when one node pair is from \bar{L}_1 , and the other is from L_2 (Fig. S21b).

Correspondingly, we have

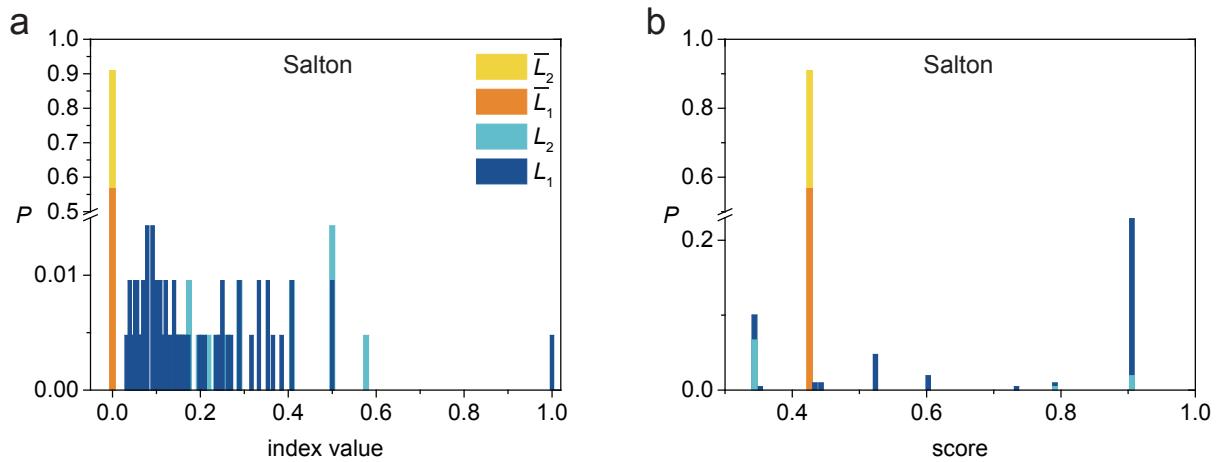
$$\text{AUC}_{s2} = \frac{n'}{n} + \frac{1}{2} \frac{n''}{n} = (1 - p_1)p_2 + \frac{1}{2}(1 - p_1)(1 - p_2). \quad (\text{S23})$$

For the ranking in Fig. **S21c**, we have $L_2 > \bar{L}_1 \cup \bar{L}_2 > L_1$. Node pairs in L_2 outscore all positive samples (the whole set L^P). Hence, this gives

$$\text{AUC}_{s3} = \frac{n'}{n} + \frac{1}{2} \frac{n''}{n} = \frac{1}{2}(1 - p_1)(1 - p_2). \quad (\text{S24})$$

By comparing the 6 equations (Eq. (S22), Eq. (S23), Eq. (S24), $\text{AUC}_{\text{lower}} = p_1(1 - p_2) + \frac{1}{2}(1 - p_1)(1 - p_2)$, $\text{AUC}_{\text{upper}} = p_1 + \frac{1}{2}(1 - p_1)(1 - p_2)$, and $\text{AUC}'_{\text{upper}} = p_1 + (1 - p_1)p_2 + \frac{1}{2}(1 - p_1)(1 - p_2)$), we can find that the $\text{AUC}'_{\text{upper}}$ gives the highest AUC value. Hence, Fig 1d of the main text gives the optimal rankings of the three sets. Indeed, this is in line with the theory of machine learning, as the score of the positive sample should be greater than that of the negative one if we assume the positive one is expected to be predicted during the learning process.

To further test if a machine learning algorithm practically re-arranges the ranking, we show an example below. The Salton index is used. The distribution of index values in different sets is shown in Fig. **S22a**. After applying the Random Forest algorithm, the classifier finds the mapping function to transform the index value into the score. The corresponding distribution of the score is presented in Fig. **S22b**. The set $\bar{L}_1 \cup \bar{L}_2$ indeed moves to the middle, giving rise to a ranking similar to the optimal ranking derived theoretically.



Supplementary Figure S22: A real case that has achieved the mapping from the index value to the score. **(a)** The distribution of index values obtained. **(b)** The distribution of scores obtained by using the Random Forest classifier in the supervised approach. This network is “56e9e0d7a6d70217090cdffa” in the data set.

S8. ANOTHER EXAMPLE OF FEATURE AND INDEX SELECTION IN LINK PREDICTION

In the main text, taking the common neighbor feature as an example, we show the theoretical finding can help us to determine the feature and index selection. To further validate the preceding analysis, we here conduct a second analysis using the path feature (Table S2). When using the index SPL to make unsupervised predictions in two real networks, we obtain $AUC = 0.567$ in both networks. The prediction results are relatively low, so it is difficult for us to decide whether the path feature is suitable for these two real networks. Under this case, we have to try one by one for the other 4 indexes on the two networks when the p_1 and p_2 values are unknown. However, calculating the maximum capability of the path feature (AUC_{upper} presented in the main text) shows that we should switch to a new feature for prediction in the network E and should try the other indexes based on the path feature in the network F (Table S2). This analysis further shows that the theoretical finding presented in the main text can be applied to optimize the feature and index selection.

Moreover, as the theoretical expression can confirm whether an index is superior to the other indexes, our theoretical finding can provide strong support for experimental validation such as the interaction between genes, and protein-protein interactions. More importantly, the theoretical expression can also help to estimate how close to the upper bound for observed performance in the same topological feature.

	SPL	p_1	p_2	AUC_{upper}	LP	Katz	FL	RSS
Network E	0.567	0.158	0.026	0.568	0.567	0.567	0.567	0.567
Network F	0.567	1.0	1.0	1.0	0.867	0.867	0.869	0.841

Supplementary Table S2: The AUC performance of the unsupervised prediction using SPL is the same for both networks E and F. However, using the p_1 and p_2 value, the capability of the path feature can be estimated. The path feature is not suitable for network E but has potential in network F. The network E is “56e98770a6d70217090cde08”, and the network F is “Cat_cerebral_hemisphere_cortex_only” in the data set.

S9. THE THEORETICAL EXPRESSION OF p_1 AND p_2

To give more insights into the structural characteristics that make a topological feature effective in link prediction, we deduct the theoretical expression of p_1 and p_2 . Here, we take the common neighbor feature as an example. The idea behind the feature is that two unconnected nodes that share the same neighborhood nodes are likely to become a link. Hence, the indexes based on common neighbor assign the value greater than 0 to a node pair only if this node pair belongs to a link of the closed triangle. Since p_1 is the fraction of missing links with an index value greater than 0, p_1 can be quantified as the probability that a randomly picked node pair from all existing links of a network is exactly from one link in a closed triangle. Therefore, the theoretical expression of p_1 is defined as

$$p'_1 = \frac{3 * N_{\Delta} - S_{\Delta}}{L}, \quad (\text{S25})$$

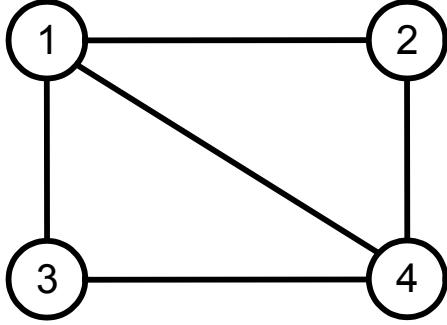
where the N_{Δ} is the number of closed triangles in a network. The S_{Δ} is the number of times that a link is shared by multiple triangles (Fig. S23). Because a link can belong to multiple triangles, the $3 * N_{\Delta}$ would over-count the number of links belonging to a triangle and has to subtract the number of times a link appears in other triangles.

Likewise, As p_2 is the fraction of nonexistent links with an index value greater than 0, p_2 can be quantified as the probability that a randomly picked node pair from all nonexistent links of a network shares a common neighbor (this node pair and an open triangle constitute a closed triangle). Hence, the theoretical expression of p_2 is formulated as

$$p'_2 = \frac{N_{\wedge} - S_{\wedge}}{\frac{N(N-1)}{2} - L}, \quad (\text{S26})$$

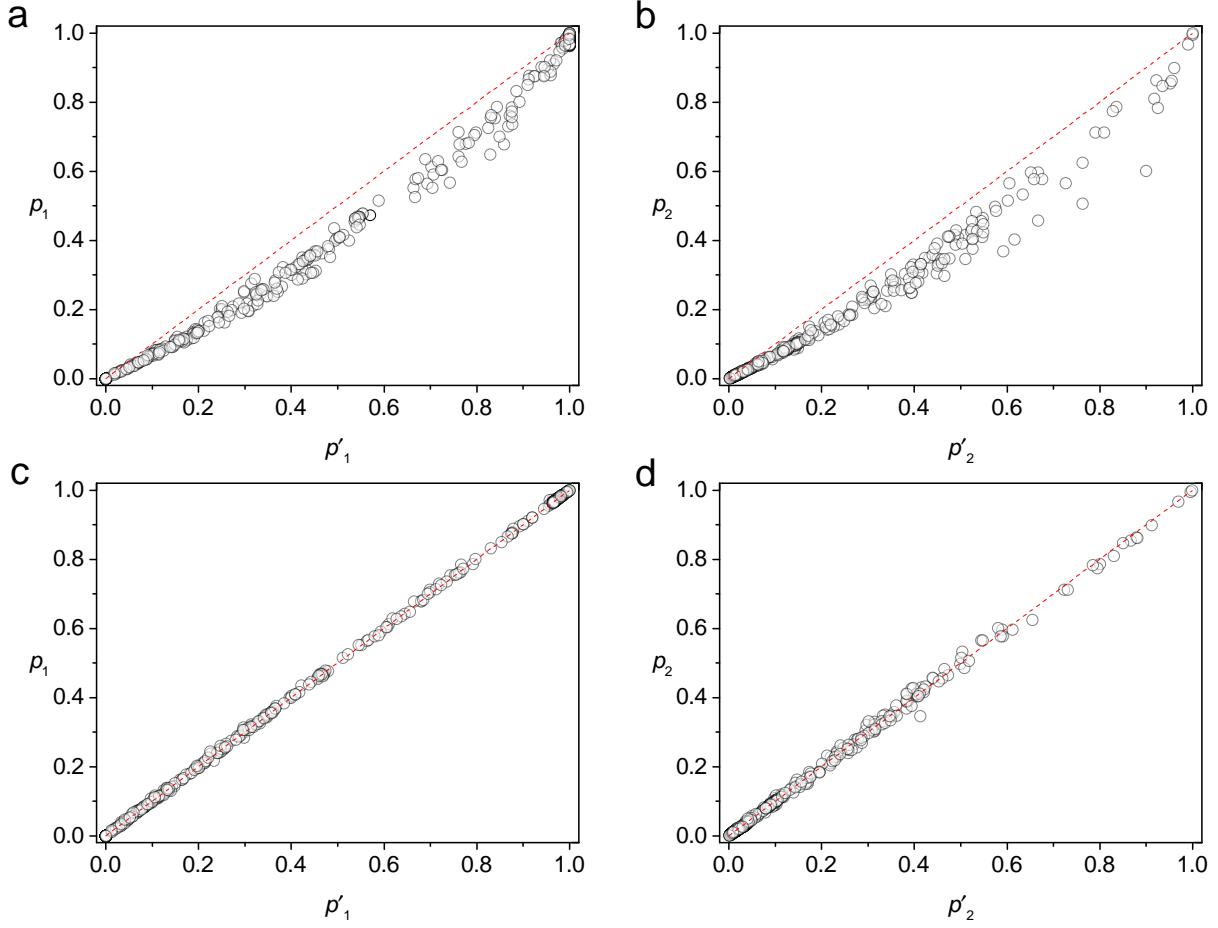
where the N_{\wedge} is the number of open triangles in a network (the triad that two nodes that are not directly connected but both connect to the third node). The S_{\wedge} is the number of times that an unconnected node pair is shared by other open triangles, which can cause an over count if it is not subtracted (Fig. S23). The $\frac{N(N-1)}{2} - L$ corresponds to the total number of unconnected node pairs.

To validate Eqs. (S25) and (S26), we take 550 empirical networks to directly calculate the theoretical values p'_1 and p'_2 . Shading multiple links will change the existing topology of the original network which makes the estimate of N_{Δ} or N_{\wedge} incorrect. Still, we observe an overall nice agreement between the theoretical and empirical values even when 20% of links



Supplementary Figure S23: An illustration of explaining parameters in Eqs. (S25) and (S26). Since the two closed triangles (\triangle_{134} and \triangle_{124}) share the same link (1-4), $N_\Delta = 2$ and $S_\Delta = 1$. Similarly, as the two open triangles (\wedge_{123} and \wedge_{243}) share the unconnected node pair (2-3), $N_\wedge = 2$ and $S_\wedge = 1$.

are temporally removed (Figs. S24a, b). Moreover, to give more reasonable evidence of Eqs. (S25) and (S26), we here also consider the situation that computing the p'_1 and p'_2 from the rest 80% of links. The results from Fig. S24c and Fig. S24d show that the theoretical values (p'_1 and p'_2) are line with the empirical values (p_1 and p_2), demonstrating a perfect agreement. The theoretical expressions explain why C itself is insufficient to characterize the capability of common neighbor feature. More importantly, the Eqs. (S25) and (S26) help us understand network characteristics associated with the utilization of a topological feature in link prediction.



Supplementary Figure S24: Evidence of the theoretical values p'_1 and p'_2 applied to 550 real networks. **(a, b)** The theoretical values p'_1 and p'_2 are given by Eq. (S25) and Eq. (S26) from the original network, respectively. **(c, d)** The theoretical values p'_1 and p'_2 are given by Eq. (S25) and Eq. (S26) from the rest 80% of links, respectively. In each network, we generate 200 independent pairs of L^P and L^N sets based on the random sampling (removed 20% of links). In the figure, we use the average value of them.

S10. EXTENDED DISCUSSION ON THE PREDICTION PERFORMANCE MEASURED BY PRECISION

The precision measure involves a hyper-parameter L_k for the cutoff of the top-k node pairs. Therefore, we might not well measure the performance of an index when only using precision. Here, we show the link prediction performance in two networks (Table **S3**). When using CN index to make unsupervised predictions in network G, we obtain $AUC = 0.533$, suggesting that the CN index overall has limited potential ($AUC_{upper} = 0.533$). But when measuring the performance by precision, the obtained measure can vary significantly with different choices of L_k . When choosing $L_k < p_1|L^P|$, the predicted upper bound is 1. In this case, we also obtain a high precision value $Precision = 0.973$. But because p_1 is very small for network G, a high precision value only suggests that the prediction is correct for the very top candidates. When L_k becomes larger, the precision drops drastically ($Precision = 0.550$ for $L_k = 121$).

Likewise, in network H, the performance of unsupervised prediction by LP index yields $AUC = 0.826$. The LP index also has a high potential in terms of the AUC measure ($AUC_{upper} = 0.977$). But if measured by precision, we obtain a low value ($Precision = 0.381$ for $L_k < p_1|L^P|$). Hence, the prediction accuracy is low for the top candidates. But if the candidates increases, the precision goes up again. When $L_k > p_1|L^P|$, we obtain $Precision = 0.711$.

The two networks in Table **S3** demonstrate a vivid example on the complexity in interpreting the precision measure. One has to take into the p_1 into consideration in order to explain the number of populations the precision is measured from.

	AUC	p_1	$ L^P $	$p_1 L^P $	Precision	Precision
Network G	0.533	0.067	241	16	0.973 ($L_k = 10$)	0.550 ($L_k = 121$)
Network H	0.826	0.976	83	81	0.381 ($L_k = 21$)	0.711 ($L_k = 83$)

Supplementary Table S3: The precision gives different performance when choosing different L_k . The network G is “Water.Distribution.Network.EXNET”, and the network H is “Freshwater_stream_webs_Stony” in the data set.