A Transformative Topological Representation for Link Modeling, Prediction and Cross-domain Network Analysis

Kai Zhang, Junchen Shen, Gaoqi He, Yu Sun, Haibin Ling, Hongyuan Zha, Honglin Li, Jie Zhang

***Abstract*—Many complex social, biological, or physical systems are characterized as networks, and recovering the missing links of a network could shed important lights on its structure and dynamics. A good topological representation is crucial to accurate link modeling and prediction, yet how to account for the kaleidoscopic changes in link formation patterns remains a challenge, especially for analysis in cross-domain studies. We propose a new link representation scheme by projecting the local environment of a link into a "dipole plane", where neighboring nodes of the link are positioned via their relative proximity to the two anchors of the link, like a dipole. By doing this, complex and discrete topology arising from link formation is turned to differentiable point-cloud distribution, opening up new possibilities for topological feature-engineering with desired expressiveness, interpretability and generalization. Our approach has comparative or even superior results against state-of-the-art GNNs, meanwhile with a model up to hundreds of times smaller and running much faster. Furthermore, it provides a universal platform to systematically profile, study, and compare link-patterns from miscellaneous real-world networks. This allows building a global link-pattern atlas, based on which we have uncovered interesting common patterns of link formation, i.e., the bridge-style, the radiation-style, and the community-style across a wide collection of networks with highly different nature.**

***Index Terms*—Link Prediction, Topological Representation, Complex Networks**

I. Introduction

MANY complex social, biological, or physical systems are characterized as networks, where vertices represent individual agents and links signify their interactions [1]–[9]. Due to the cost and uncertainties of data acquisition, networked data are often incomplete with missing links. As a result, estimating the likelihood that an unobserved link actually exists based on the observed portion of the network, commonly known as link prediction, thus becomes a fundamental problem in network and information sciences [10]–[16]. Accurate link prediction is not only a practical goal in physics, social networks and recommender systems, but also provides valuable insights into scientific discoveries related to network structure, dynamics, and organizing principles.

Link prediction is a statistically hard problem and no single model has shown to be superior for all networks unless by stacking different models together as an ensemble model [16]. The predictability is related to the intrinsic structural regularity of the network [15]. Early work study rules of link formation and design various heuristics to evaluate the proximity between two nodes for link prediction [17]–[21]. Probabilistic models [22]–[24] and maximum likelihood approaches [25], [26], on the other hand, estimate the probability of a link conditioned on the network structure or node attributes. Link prediction can also be solved by using latent node representations through graph embedding techniques [27]–[31].

In recent years, learning-based algorithms that predict the missing links through a classifier [32] began to draw more attention. The key advantage of learning based algorithms nowadays lies in their ability to automatically craft features for the predictive task through end-to-end optimization [33] by leveraging a specific inductive bias. In fact, the grand success of deep neural networks is largely attributed to their power of learning good representations. This philosophy has inspired a surge of interest in applying graph neural networks (GNNs) [29], [31], [34]–[37] to extract useful topological features for link prediction. The pioneering idea of GNN-based link prediction is due to Zhang *et al.* [38], which extracts the "*local enclosing subgraph*" for a target link to capture the key topological information for a link to be formed. By doing this, link prediction is converted to subgraph classification, on top of which powerful GNNs can be readily introduced to generate significantly improved results over previous methodologies.

Despite recent progresses, substantial challenges persist in link representation learning. This is because the links of a network are often enclosed in local subgraphs involving an arbitrary number of nodes and kaleidoscopic topological variation, which is notoriously hard to align or profile. Therefore GNN models may have to use heuristics such as truncation or padding to obtain constant-sized features [39], which inevitably alters graph topology; the convenient choice of graph pooling, which collapses all the nodes into one, may incur information loss and become the bottleneck as noted

Kai Zhang, Junchen Shen and Gaoqi He are with the East China Normal University, 200062 Shanghai, China. E-mail: {kzhang@cs, 51215901048@stu, gqhe@cs}.ecnu.edu.cn.

Yu Sun is with the Indeed Inc., 94086 Sunnyvale, United States. E-mail: sunyu9910@gmail.com.

Haibin Ling is with the Stone Brook University, 11794 New York, United States. E-mail: hling@cs.stonybrook.edu.

Hongyuan Zha is with the Chinese University of Hong Kong, Shenzhen, 518172 Shenzhen, China. E-mail: zhahy@cuhk.edu.cn.

Honglin Li is with the East China Normal University, 200062 Shanghai, China. E-mail: hlli@ecnu.edu.cn.

Jie Zhang is with the Fudan University, 200438 Shanghai, China. E-mail: jzhang080@gmail.com.

This work was supported partially by National Key Research and Development Program of China with grants number 2022YFC3400500 and 2022YFC3400501, and National Natural Science Foundation of China with grants number 62276099 (Corresponding author: Jie Zhang, Honglin Li).

by [40], [41]. Moreover, graph convolutions can be difficult to interpret when using topological features for message passing. As to graph embedding techniques [29], [31], [36], the low-dimensional node embedding vectors are defined only for each individual network separately, which limits their utility in cross-domain studies involving multiple networks. Overall, the lack of a good link representation hampers not only link prediction accuracy, but also knowledge discovery from links of miscellaneous real-world networks.

In this paper, we propose a transformative link representation that characterizes complex link patterns in the network with desired interpretability, generalization, and cross-domain modeling capacity. The key idea is to project the local environment (or enclosing-subgraph) of a link into a two-dimensional "*dipole plane*". In this plane, the neighboring nodes of the link are positioned via their relative proximity to the two anchors of the link through random-walks, like a dipole. By doing this, complex and discrete topology arising from link formation is turned to continuous and differentiable distribution of a point-cloud, opening up new possibilities for topological feature-engineering in link modeling and prediction.

The presented link representation offers significant advantages for link modeling and prediction. Theoretically, it possesses the ability to discern between link patterns with different topological characteristics, which is critical for training discriminative models. The density-based profile is also robust against link perturbations due to inherent kernel smoothing, thus greatly contributing to the generalization performance. Empirically, the proposed representation yields comparable or even superior results in link prediction compared to the best-performing Graph Neural Networks (GNNs), yet with a compact model up to hundreds of times smaller and running much faster. Finally, the proposed link representation is physically interpretable and with naturally aligned dimensions, therefore it provides a universal platform to study, compare and explore link-patterns from networks across different domains simultaneously. This allows building a global link-pattern-atlas, based on which we have uncovered interesting general themes of link formation, as well as network similarities that may otherwise be hidden from a collection of scientific, social, biological and technological networks of highly different nature.

The subsequent sections of this paper are structured as follows. Section II provides a review of related work. Section III introduces the proposed link representation scheme, while Section IV contains the theoretical analysis. Experimental evaluations and cross-domain case studies are presented in Section V. The concluding remarks and identification of future directions are outlined in the last section.

II. Related Work

A large body of link prediction methods has been devised and can be broadly grouped into similarity-based approaches, probabilistic models, embedding techniques, and learning-based models, as outlined in relevant surveys [42], [43].

Early work of link prediction focus on the design of various heuristics to quantify the structural relations between the nodes. Well-known examples include the common neighbors [17], Jaccard similarity [18] and Adamic–Adar index [19] based on first-order neighbors. Later, high-order heuristics are proposed to characterize the relation between the nodes in a more global context. For example, Rooted PageRank [20] measures the probability of reaching a node from a predefined root node through random-walks, implicitly taking into account all possible paths in the network. Katz score [21] considers paths of varying lengths and assigns higher weights to shorter paths. SimRank [44] evaluates node similarity by recursively looking into their neighbors. These heuristics are easy to compute, and can well recover the missing links if the underlying assumption of homophily is satisfied.

Another big family of algorithms relies on latent node embeddings as fundamental features for link prediction. Such low-dimensional representations can be obtained through traditional matrix factorization technique [27] and stochastic block models [28]. In recent years, the development of distributed representation learning frameworks such as the skip-gram model in word2vec [45] lays a solid foundation for network embedding (or node embedding), with prominent examples including DeepWalk [29], LINE [30] and node2vec [31], and a unified matrix factorization view can be found in [36]. The high-quality network embedding allows faithfully capturing the proximity relation between the nodes of a network, and hence sophisticated classifiers can be built upon pairs of such node embeddings to perform link prediction. See a comprehensive review for such methods in [42].

The concept of using GNN models for link prediction is pioneered by Zhang *et al.* in their seminal work of "SEAL" [38], which introduces the notion of "*local enclosing subgraphs*". Given a graph G and any pair of nodes x and y representing a target link, the local enclosing subgraph G_{xy} is defined as a subgraph that is composed of the union of x and y's neighbors up to l hops. The enclosing subgraph delineates the "l-hop surrounding environment" centered around the target link (x, y), encapsulating the crucial topological information necessary for link formation by theoretically approximating a wide range of high-order heuristics. It allows link prediction to be converted faithfully to subgraph classification. Automatic graph feature engineering can then be performed on top of the enclosing subgraphs with GNNs, leading to promising results that significantly outperform previous algorithms.

Inspired by SEAL [38], various innovations have been made toward extracting features from the enclosing subgraph (or the entire training graph). For example, Walk-Pooling [41] integrates node representations and graph topology into random-walk based transition probabilities, and uses the difference of such probabilities before and after removing the focal link of the enclosing subgraph as the high-order structural information for link prediction. Line-Graph Network [40] proposes to convert the enclosing subgraph to a line graph where each node corresponds to a unique link in the original graph. The features of the link can then be learned directly through the line graph representation, turning link prediction problems to node classification with significantly improved performance. Distance-enhanced GNN [46] combines the pairwise node distances with the GNN model, and obtains promising results in the problems of drug-drug-interaction and protein-protein-

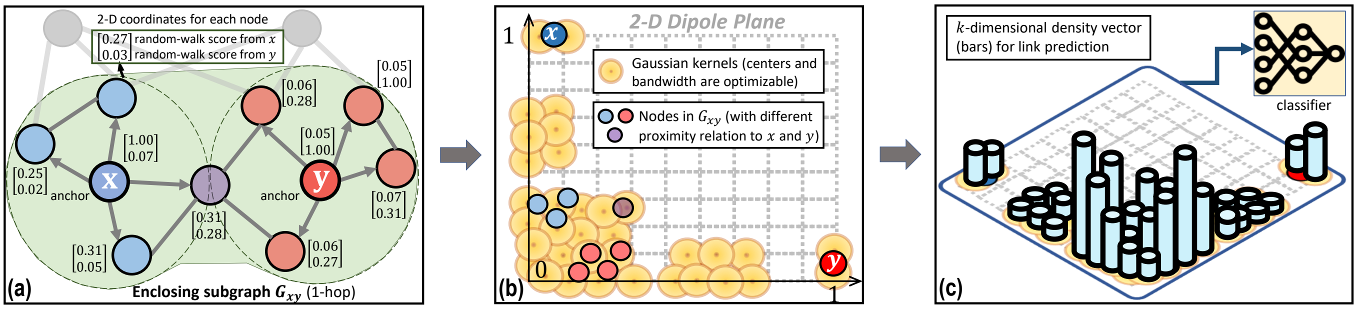

2-D coordinates for each node
$\begin{bmatrix}0.27\\0.03\end{bmatrix}$ random-walk score from x / random-walk score from y
$\begin{bmatrix}0.05\\1.00\end{bmatrix}$
$\begin{bmatrix}0.06\\0.28\end{bmatrix}$
$\begin{bmatrix}1.00\\0.07\end{bmatrix}$
$\begin{bmatrix}0.05\\1.00\end{bmatrix}$
$\begin{bmatrix}0.25\\0.02\end{bmatrix}$
x
anchor
y
anchor
$\begin{bmatrix}0.07\\0.31\end{bmatrix}$
$\begin{bmatrix}0.31\\0.28\end{bmatrix}$
$\begin{bmatrix}0.31\\0.05\end{bmatrix}$
$\begin{bmatrix}0.06\\0.27\end{bmatrix}$
(a)
Enclosing subgraph G_{xy} (1-hop)
2-D Dipole Plane
1
0
1
x
y
Gaussian kernels (centers and bandwidth are optimizable)
Nodes in G_{xy} (with different proximity relation to x and y)
(b)
k-dimensional density vector (bars) for link prediction
classifier
(c)

Fig. 1. The proposed Dipole Space Density Network (DSDN) that transforms a link to a continuous point-cloud distribution through an end-to-end learning architecture. (a) For a focal link between two anchor nodes x and y, find the l-hop (here $l = 1$) enclosing-subgraph G_{xy}. Then use two random-walks starting from x and y to measure the proximity between each node in G_{xy} and the two anchors. (b) Project the nodes in G_{xy} to the "dipole plane", where the 2-D coordinates are exactly the scores generated by the two random-walks starting from x and y. Then estimate the density of the 2-D point-cloud using k Gaussian kernels whose centers and bandwidth are both optimizable. (c) Use the estimated density as input features to a classifier for link prediction.

association. The distance calculations are based on a set of random anchors to improve the computational efficiency.

III. A New Representation for Link Prediction

Let $G_0 = (\mathbb{V}_0, \mathbb{E}_0)$ be a "complete" network with edge set $\mathbb{E}_0$ and node set $\mathbb{V}_0$. In practice, only a partial version $G = (\mathbb{V}, \mathbb{E})$ is observed with $\mathbb{E} \in \mathbb{E}_0$, $\mathbb{V} \in \mathbb{V}_0$. The goal of link prediction is then to predict whether a link indeed exists between a pair of unconnected nodes, $(x, y) \in \{\mathbb{V} \times \mathbb{V} - \mathbb{E}\}$, based on the observed graph G.

A focal link between any two nodes, x and y, is characterized by its *enclosing subgraph* G_{xy} as in Fig. 1(a), i.e., a subgraph composed of the l-hop neighbors around x and y. In constructing the enclosing subgraph, l is typically chosen as a small integer like one or two, and the truly observed focal link between x and y will be removed from the subgraph. Next, we show how to obtain a desired representation of the link between x and y based on its enclosing subgraph.

The proposed approach is named as "Dipole Space Density Network (DSDN)" and illustrated in Figure 1. First, we use two random-walks that start from the two anchors of the link, x and y, to evaluate the proximity between the neighboring nodes and the two anchors. Then the "local environment" of the target link G_{xy} can be projected as a point-cloud onto a 2-D "dipole plane", whose distribution closely reflects local topological organizations of the target link. Finally, an adaptive kernel estimator is used to profile the density distribution as a compact and informative representation for link prediction. In the following, we introduce the algorithm details.

A. From Enclosing-subgraph to Point-cloud

We first use two random-walks starting from the two anchors of the target link, x and y, to evaluate the proximity between the nodes in G_{xy} and the two anchors. More specifically, we resort to Random-walk With Restart (RWR) to quantify such relation [47]. It starts from one anchor node x and iteratively visits neighbors encountered, each step having a probability $1 - c$ to jump back to the start node x, as

$$\mathbf{p}_x^{(t+1)} = c \cdot \tilde{\mathbf{A}}_{xy} \mathbf{p}_x^{(t)} + (1 - c) \cdot \mathbf{e}_x. \tag{1}$$

Here t is step, $\mathbf{p}_x \in \mathbb{R}^{n_{xy} \times 1}$ is node-wise random-walk scores, n_{xy} is the number of nodes in G_{xy}, $\tilde{\mathbf{A}}_{xy}$ is the transition probability matrix of G_{xy}, $1 - c$ is the restart probability, and $\mathbf{e}_x$ is a one-hot vector indicating the start node x. The converged distribution for $t \to \infty$ has closed form [47],

$$\mathbf{p}_x = (\mathbf{I} - c \cdot \tilde{\mathbf{A}}_{xy})^{-1} \mathbf{e}_x. \tag{2}$$

The random-walk score $\mathbf{p}_x$ reflects the proximity of each node to anchor x. If $c = 0$, the re-start probability $1 - c$ is 1, namely the random-walk will be frozen at x with $\mathbf{p}_x$ being a one-hot vector; if $c \to 1$, $\mathbf{p}_x$ becomes the stationary distribution of a Markov random-walk that freely explores G_{xy} but never jumps back to x; in other words, the start node is forgotten. In practice, we use $c \in [0.5, 0.8]$ to explore the subgraph G_{xy} thoroughly while still remembering where the starting node is. We also prefer the closed-form random-walk distribution (2) because it is more robust than the discrete version (1) and saves the effort of determining the number of iterations.

We use two separate random-walks from x and y as in Fig. 1(a), to map G_{xy}'s nodes to a new space with n_{xy} pairs of coordinates $\mathbf{P}_{xy} = [\mathbf{p}_x \; \mathbf{p}_y] \in \mathbb{R}^{n_{xy} \times 2}$, and normalize them by the maximum score, as

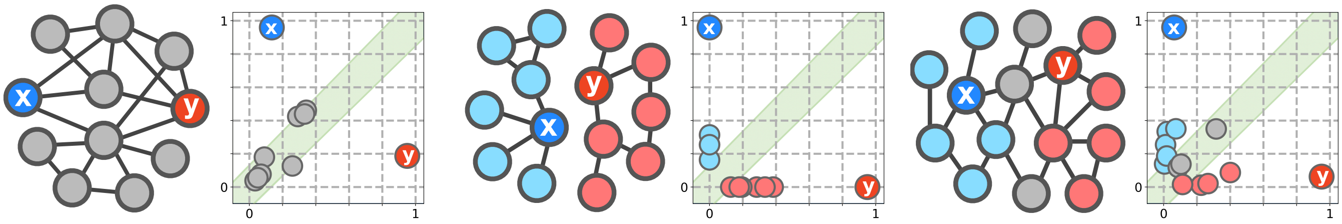
$$\mathbf{P}_{xy} = \left[\frac{\mathbf{p}_x}{\max(\mathbf{p}_x)} \quad \frac{\mathbf{p}_y}{\max(\mathbf{p}_y)} \right]. \tag{3}$$

By normalizing the random-walk score vectors $\mathbf{p}_x$ and $\mathbf{p}_y$ with their maximum entries, the resultant point-clouds will always reside in a unit square $[0, 1] \times [0, 1]$. We call it a "*dipole plane*" as in Fig. 1(b), because it is induced by the two anchors of the target link, like a dipole. It serves a universal "coordinate system" quantifying how the neighboring nodes of a link gather around its two anchors[1].

B. From Point-cloud to Density-profile

The "dipole plane" allows any link of interest to be projected and turned to a 2-D point-cloud. *How is subgraph topology translated to point-cloud distribution*? Since the 2-D point coordinates quantify the proximity of a node to the two anchors of the link, x and y, some simple observations can be made: (i) nodes far away from x and y must have lower

[1] For each link, the two anchors can be either labeled as (x, y) or (y, x), so we can generate two point-clouds that are symmetric with regard to the diagonal of the dipole plane, with the same label.



(a) An observed link between x and y (2-hop).

(b) A NULL link between x and y (2-hop).

(c) An observed link between x and y (2-hop).

Fig. 2. Three link examples, each with their enclosing-subgraphs G_{xy} on the left, and resultant point-clouds on the right. The focal link between x and y will be removed if it is truly observed. The 2-D coordinates of the point-cloud are proximities of each node to x and y. Here, blue and red mark the nodes closer to anchor x and y, respectively, while gray nodes along the diagonal are about equally distant away from x and y. (a) nodes in G_{xy} are roughly equal-distant away from x and y, so the point-cloud is distributed along $y = x$; (b) nodes in G_{xy} breaks into two components, so the point-cloud forms two arms along the two axes of the dipole plane; (c) nodes in G_{xy} fall into three zones: closer to anchor x (blue), closer to anchor y (red), or in the middle (gray).

proximity scores and hence be mapped around the origin, otherwise they will be pushed away from it; (ii) nodes equally distant away from x and y will be mapped around $y = x$. In fact, the symmetry, grouping, and shape of the point-cloud are closely related to the topological characteristics of the link.

Fig. 2 gives some examples: (a) is an observed link, and most nodes in G_{xy} are equally distant from x and y, so the point-cloud falls in the diagonal band; (b) is a NULL link whose G_{xy} breaks into two components, so the point-cloud has two arms along the two axes; (c) is an observed link, where blue and red nodes are close to x and y, respectively, while gray nodes lie in the middle; so the point-cloud has a mixed pattern of distribution. In all cases, anchor x and y are around corners. Sometimes the points aggregate into clusters in the dipole plane, meaning that the corresponding nodes in one cluster have similar proximity patterns to x and y.

Next we show how to model these highly diversified point-cloud distributions such that a compact, informative representation can be obtained for link prediction. Here, a key observation is that not all the locations in the dipole plane have useful density values: some locations could be rarely populated by the point-cloud, while some locations may always have similar density values for true links and NULL links. Therefore, we are interested in those locations in the dipole plane whose densities are useful for discriminating the true links against the NULL ones.

To achieve this, we borrow ideas from Parzen window density estimator [48], and extend it to a discriminative version. We place a number of k Gaussian kernels in the dipole plane that are marked as yellow circles in Fig. 1(b), each with center $\boldsymbol{\mu}_j \in \mathbb{R}^{1\times 2}$'s for $j = 1, 2, ..., k$, and a common bandwidth $h \in \mathbb{R}_+$. These kernel parameters are then optimized in an end-to-end fashion by minimizing the link-prediction error (defined in Sec III.C), so that a faithful density landscape can be recovered for link prediction tasks.

Specifically, we compute the affinity between the n_{xy} points in (3) and the k kernel centers $\boldsymbol{\mu}_j$'s as a $\mathbb{R}^{n_{xy}\times k}$ matrix

$$\mathbf{W}_{ij} = \exp\left(-\frac{\left\|\mathbf{P}_{[i,:]} - \boldsymbol{\mu}_j\right\|^2}{2h^2}\right), \tag{4}$$

where $\mathbf{P}_{[i,:]}$ is the i-th row (point) in $\mathbf{P}_{xy}$ (sub-index xy omitted for convenience). Namely, if the i-th point $\mathbf{P}_{[i,:]}$ falls in the receptive domain of the j-th Gaussian kernel, it will trigger the kernel with strength $\mathbf{W}_{ij}$. After normalizing each row of $\mathbf{W}$ to probabilities that sum up to 1, so that its ij-th entry then signifies the probability that the i-th point triggers the j-th kernel, we then sum up all the rows of $\mathbf{W}$ to get the accumulated density at each kernel location $\boldsymbol{\mu}_j$, as

$$f(\boldsymbol{\mu}_j) = \sum_{i=1}^{n_{xy}} \frac{\mathbf{W}_{ij}}{\sum_{j'=1}^{k} \mathbf{W}_{ij'}}. \tag{5}$$

The point-cloud distribution can then be nicely encoded as the k-dimensional density-profile

$$\mathbf{F}_{xy} = \left[f(\boldsymbol{\mu}_1), \quad f(\boldsymbol{\mu}_2), \quad ..., \quad f(\boldsymbol{\mu}_k)\right]. \tag{6}$$

The representation $\mathbf{F}_{xy}$ has several advantages: (i) it is easy to compute and can be very compact; (ii) it is invariant to the order of the nodes in subgraph G_{xy}; (iii) it has k dimensions consistently defined across different subgraphs of a network, even if they have a varying number of nodes. In fact, by sharing the Gaussian centers $\{\boldsymbol{\mu}_j's\}$ in the dipole plane, $\mathbf{F}_{xy}$ can be used as an intrinsic, well-aligned link feature even across different networks for cross-domain studies; (iv) it is interpretable. The k dimensions of $\mathbf{F}_{xy}$ sum up to n_{xy}, the number of nodes in G_{xy} due to normalization in (5)[2]. So the j-th dimension $f(\boldsymbol{\mu}_j)$ can be deemed as the "amount" of nodes in G_{xy} whose relative proximity to the two anchors are close to $\boldsymbol{\mu}_j = [\boldsymbol{\mu}_{j1} \ \boldsymbol{\mu}_{j2}]$, with $\boldsymbol{\mu}_j$ the proximity pattern specified by the j-th Gaussian center. With such physical meaning, our network model is no longer a black-box, but instead allows human interpretation of the prediction results by looking into dominantly weighted Gaussian kernels in the classifier, each of which represents a specific proximity or topological pattern towards link formation in the enclosing subgraph.

C. Link Prediction

The k-dimensional density profile $\mathbf{F}_{xy}$ in (6) is used as features for a multilayer perceptron (MLP) for link prediction, with the cross entropy loss function. This then leads to a complete end-to-end architecture, in which the Gaussian

[2] In this sense, $\mathbf{F}_{xy}$ is not strictly a density because it sums up to n_{xy}, the subgraph size. We believe n_{xy} carries useful information in link prediction so we do not normalize $\mathbf{F}_{xy}$ to sum to 1.

kernels are placed adaptively to generate compact, discriminative representation. Empirically, $k \approx 100$ suffices for most datasets. This gives a parsimonious model that is not only computationally efficient, but also less prone to overfitting.

In case there are extra node attributes available (such as bags of words feature for a paper in citation networks), the dipole plane provides an innovative platform for heterogeneous message passing, i.e., the cross-talk between the points (nodes) and the "Gaussian sensors". We can use the pairwise relation between the n_{xy} points and the k sensors $\mathbf{W}$ as an "adjacency" matrix to aggregate node attributes to the nearby sensors, as

$$\overline{\mathbf{Z}}_{xy} \leftarrow \sigma\left(\mathbf{D}^{-1}\mathbf{W}^{\top}\mathbf{Z}_{xy}\cdot\mathbf{T}\right).$$

Here, $\mathbf{D}$ is column-wise degree matrix of $\mathbf{W}$, $\mathbf{Z}_{xy} \in \mathbb{R}^{n_{xy}\times d}$ is attributes matrix for nodes in subgraph G_{xy}, $\mathbf{T} \in \mathbb{R}^{d\times d'}$ is a linear transform, and $\sigma(\cdot)$ is nonlinear activation. The resultant sensor-wise feature matrix $\overline{\mathbf{Z}}_{xy} \in \mathbb{R}^{k\times d'}$ is of a fixed dimension regardless of the size of the enclosing subgraph G_{xy}, and carries both featural and topological information of the enclosing subgraph. We will feed it into the FC-layers for the final link prediction task.

IV. Properties of the Proposed Link Representation

In this section we study properties of the proposed link representation. It transforms a target link by two steps, i.e., the enclosing subgraphs of the target link are first transformed to a 2-D point-cloud, and then to a fixed-dimensional density vector. In fact, these two steps have interesting properties that are complementary to each other. On the one hand, transforming an enclosing subgraph to a point-cloud is a mapping that effectively captures topological differences of the enclosing subgraphs. On top of such sensitivity, the kernel density estimator can further improve the stability of the learned feature, which is beneficial to link prediction tasks.

A. Discriminative Properties of the Point-cloud

A natural question on the representation power of the point-cloud is under what condition it can distinguish between two different enclosing subgraphs, so that accurate link prediction can be made based on the topological structure of subgraphs.

For simplicity, we consider one anchor x in each subgraph with point-cloud coordinate p_x as in (2). This is because the choice of the two anchors is independent and so the extension from one anchor to two anchors is trivial. Second, the anchor is always the first node in a subgraph, followed by its 1st-order neighbors, and then the 2nd-order neighbors, and so on; this is a natural partial ordering due to the "central status" of the anchor, under which we shall compare the point-clouds from different subgraphs. Under this partial order, we provide sufficient condition for point clouds of two subgraphs to be different. We restrict ourselves to the case of equal-sized subgraphs, because two subgraphs with different numbers of nodes must have different point-clouds and density profiles.

Theorem 1. *Consider unweighted, undirected subgraphs with no self-loops. Let $\mathbf{W}_1$ and $\mathbf{W}_2$ be the adjacency matrix of two equal-sized subgraphs, and w.l.o.g. assume the anchor node is the first node, and the random-walk scores starting from the anchor node are $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively, for the two subgraphs. Let $\tilde{\mathbf{A}}_1$ and $\tilde{\mathbf{A}}_2$ be the transition matrix of the two subgraphs, and $\mathbf{B}_1^*$ and $\mathbf{B}_2^*$ be the adjugate matrix of $\mathbf{I}-c\tilde{\mathbf{A}}_1$ and $\mathbf{I}-c\tilde{\mathbf{A}}_2$, respectively. Then we have:*

1) *if $\frac{|\mathbf{B}_1^*[1,1]|}{\mathbf{B}_1^*[1,j]} \neq \frac{|\mathbf{B}_2^*[1,1]|}{\mathbf{B}_2^*[1,j]}$ for some $j \neq 1$, then $\mathbf{p}_1 \neq \mathbf{p}_2$;*
2) *if $\frac{|\mathbf{B}_1^*[1,1]|}{\mathbf{B}_1^*[1,j]} = \frac{|\mathbf{B}_2^*[1,1]|}{\mathbf{B}_2^*[1,j]}$ for all j, then $\mathbf{p}_1 = \mathbf{p}_2$;*

here j is an integer in $[1,n]$.

Proof is in the Supplementary Materials. Theorem 1 shows that as long as the ratio between the first and the j-th entry in the first column of the adjugate matrix of $\mathbf{I} - c\tilde{A}_1^{-1}$ and $\mathbf{I} - c\tilde{A}_2^{-1}$ are different for at least one j ($2 \leq j \leq n$), then the point-clouds of the two enclosing subgraphs must be different. In order for the two point-clouds to be the same, the ratios have to be the same for all $2 \leq j \leq n$ across the two matrices. In fact, we speculate that as long as the two enclosing subgraphs are non-isomorphic, then their point-clouds will be different. We have empirically verified this conjecture on a large number of subgraph pairs. However, a strict proof involves the general problem of graph isomorphism, which can be quite challenging and is being pursued as a future research topic.

B. Smoothing Effect of the Kernel Density Estimator

In this subsection, we study how the bandwidth h in (4) affects the distance between two point-cloud density-profiles as measured by (6), thereby potentially improving the stability of the representation. The distance between two point-cloud distributions is affected by many factors, e.g., locations of the points, choice of the Gaussian kernel centers, and the bandwidth. To focus on the smoothing effect of the bandwidth alone and obtain explicit bounds, we restrict ourselves to a simple scenario, in which there is only one non-overlapping pair of points across two point-clouds. Then such discrepancy will be captured by the k Gaussian kernels located in the dipole plane, as quantified below.

Theorem 2. *Let there be only one pair of non-overlapping points, $\mathbf{u}$ and $\mathbf{v}$, from two equal-sized point-clouds. The distance between $\mathbf{u}$ (or $\mathbf{v}$) and the k Gaussian centers $\boldsymbol{\mu}_j$'s are $\mathbf{d}_j^u = \|\mathbf{u}-\boldsymbol{\mu}_j\|^2$ (or $\mathbf{d}_j^v = \|\mathbf{v}-\boldsymbol{\mu}_j\|^2$) for $j = 1, 2, ..., k$. Let $\mathbf{d}_j^{min} = \min\left(\mathbf{d}_j^u, \mathbf{d}_j^v\right)$, $\mathbf{d}_j^{max} = \max\left(\mathbf{d}_j^u, \mathbf{d}_j^v\right)$, $\mathbf{d}_j^{uv} = \left|\mathbf{d}_j^u - \mathbf{d}_j^v\right|$. The distance between the density profile $\mathbf{F}_1$ and $\mathbf{F}_2$ as defined in (6) from the two point-clouds is bounded by*

$$\|\mathbf{F}_1-\mathbf{F}_2\|^2 \geq \frac{1}{2h^2}\frac{\sum_j \exp\left(-\frac{\mathbf{d}_j^{max}}{h^2}\right)}{\max\left(\sum_j \exp\left(-\frac{\mathbf{d}_j^u}{h^2}\right), \sum_j \exp\left(-\frac{\mathbf{d}_j^v}{h^2}\right)\right)},$$

$$\|\mathbf{F}_1-\mathbf{F}_2\|^2 \leq \frac{1}{2h^2}\frac{\sum_j \exp\left(-\frac{\mathbf{d}_j^{min}}{h^2}\right)}{\min\left(\sum_j \exp\left(-\frac{\mathbf{d}_j^v}{h^2}\right), \sum_j \exp\left(-\frac{\mathbf{d}_j^u}{h^2}\right)\right)}.$$

Proof is in the Supplementary Materials. Theorem 2 shows that the distance between two point-cloud densities, i.e., $\|\mathbf{F}_1 - \mathbf{F}_2\|$, is strongly modulated by the bandwidth h: a small h highlights their difference, while a large one trivializes it (due to the $\frac{1}{2h^2}$ term). An empirical verification can be found in the

TABLE I

Average link-prediction AUC on synthetic networks. In WS-model, p the is rewiring probability, and the three values chosen are for regular, small-world, and more random networks, respectively; in BA-model (scale-free networks), m_0 and m are the initial # nodes and # attached nodes in each step.

Data	*WS-network*			*BA-network* $(m = m_0)$		
Alg\Para	$p = 0$	$p = 0.5$	$p = 0.8$	1	3	5
CN	99.51±0.01	61.46±0.56	51.16±0.58	49.54±0.01	53.96±0.46	57.17±0.63
Jaccard	99.58±0.01	61.58±0.57	51.17±0.58	49.02±0.27	53.67±0.42	55.90±0.55
PA	13.25±0.10	35.48±0.24	37.27±1.45	33.57±3.01	66.66±2.38	69.12±1.67
Katz	99.79±0.01	62.75±0.27	43.52±0.84	29.55±7.25	59.95±1.68	63.19±2.16
RWR	99.90±0.01	64.92±0.29	45.61±0.58	29.54±7.27	62.89±1.47	64.93±1.58
AA	99.51±0.01	61.54±0.55	51.15±0.56	49.54±0.01	54.04±0.46	57.17±0.71
SEAL	99.94±0.03	70.73±0.39	57.07±3.49	89.58±2.85	78.56±1.73	72.07±1.71
WP	99.83±0.09	74.87±0.50	59.81±1.51	95.59±0.52	81.25±0.56	72.56±1.72
LG	99.79±0.21	75.30±0.81	61.97±1.41	95.93±0.71	79.68±1.31	74.47±1.64
Ours	**99.99±0.01**	**75.58±0.19**	**62.18±1.55**	**96.30±1.08**	**85.13±0.85**	**76.21±1.93**

Supplementary Materials, in which we show that the curves of the derived bounds and the actual distance both drop with h for a data example satisfying the conditions in Theorem 2. In fact, even for more complex cases in which the two point-clouds have more than one pair of non-overlapping points, the difference between their density profile still decays with the bandwidth, as illustrated in the Supplementary Materials.

The relation between the distance $\|\mathbf{F}_1 - \mathbf{F}_2\|$ and the bandwidth is due to the smoothing effect of Gaussian kernels [49], since large bandwidth flattens distributions. More specifically, note that the j-th density feature $f(\boldsymbol{\mu}_j)$ is a statistical average on the "amount" of nodes in G_{xy} whose proximity to the two anchors is close to $\boldsymbol{\mu}_j = [\boldsymbol{\mu}_{j1}, \boldsymbol{\mu}_{j2}]$, and sum up to $|G_{xy}| = n_{xy}$. Therefore, if the links in a subgraph are perturbed (insertion or removal) but with nodes unchanged, the density $\mathbf{F}_{xy}$ is then a re-assignment of n_{xy} "density quota" among the k kernels, and will be relatively stable if smoothed by a Gaussian whose bandwidth exceeds the level of perturbation. This is desirable because real world graphs are always incomplete, so a reasonable amount of smoothing of the distances between different link-patterns can prevent the model from remembering (overfitting) the training graph too rigidly to generalize to unseen links. Note that the degree of smoothing can be determined adaptively through end-to-end optimization of the bandwidth parameter h.

V. Experimental Results

A. Link Prediction

We evaluate link-prediction algorithms by (i) AUC of link prediction, a highly interpretable and widely used metric in the area; (ii) scalability; and (iii) model size. We use artificial networks generated with the BA-model [50] and the WS-model [51] (with 1000 nodes and different parameters), and 13 real-world benchmark networks widely used in link prediction community ([15], [38], [40], [41]), including scientific, social, biological and technological networks of diverse topological and statistical properties. See more details of the artificial and the real networks in the Supplementary Materials.

Experimental results from nine competing methods are reported, including common neighbors [52], Jaccard [18], preferential attachment [53], Katz score [21], random-walk with restart [47], Adamic-Adar [19], and state-of-the-art GNNs like SEAL [38], Walk-Pooling [41], and Line-Graph [40].

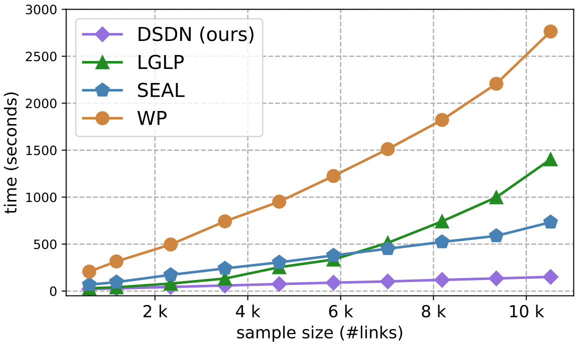


Fig. 3. Time consumption of GNN based link-prediction algorithms and our method versus the number of training links. Our approach scales linearly with sample size and is computationally more efficient.

1) Experimental setting: The training and testing data are generated with the protocol widely used in link-prediction community ([15], [38], [40], [41]). For an input network, we first randomly remove 10% observed links as the positive testing data, and randomly sample the same number of unconnected node pairs as negative testing data. The remaining 90% observed links and the same number of additionally sampled negative links are used as the positive and negative training data. We randomly generate 10 such splits for each network. All competing methods are evaluated on the same 10 data splits for each network, and the mean and the standard deviation of the AUCs are reported.

For our approach, we use 2-hop enclosing subgraphs, and the Gaussian centers $\{\boldsymbol{\mu}_j' s\}$ are initialized as follows: we first collect the enclosing subgraphs from the training links and project them onto the 2-D dipole plane $[0, 1] \times [0, 1]$ with 0.05×0.05 grids, and then initialize $\boldsymbol{\mu}_j$'s as the centers of those grids with high point-cloud densities. One can also use k-means clustering centers for initialization with pre-defined k ($k \approx 100$). The bandwidth is initialized at $h = 0.025$. The re-start probability c is chosen from $\{0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8\}$ using 5% of the training

TABLE II
AVERAGE LINK PREDICTION AUC FOR 10 LINK PREDICTION ALGORITHMS ON 13 POPULAR BENCHMARK NETWORKS; BEST AUCS ARE IN BOLD FONTS.

Domain	*Social*		*Science Collaboration*		*Citation*		
Alg\Data	Facebook	Email	NetSci	GrQ	Citeseer	Pubmed	Cora
CN	99.27±0.03	84.97±0.72	93.12±0.91	91.96±0.25	67.21±0.46	65.42±0.26	73.64±0.69
Jaccard	99.08±0.02	84.57±0.71	90.38±1.12	92.25±0.35	63.75±0.36	65.24±0.28	73.10±0.33
PA	83.15±0.22	77.79±1.45	66.54±1.75	73.61±0.84	58.51±0.93	72.00±0.62	63.65±0.50
Katz	61.18±0.36	89.75±0.48	93.83±1.01	91.27±0.47	75.35±0.79	75.54±0.60	83.36±1.22
RWR	99.31±0.02	90.40±0.16	93.87±1.01	91.39±0.49	75.48±0.83	76.49±0.63	85.07±1.22
AA	99.38±0.02	85.00±0.72	85.25±0.99	91.99±0.25	65.18±0.57	64.87±0.22	75.51±0.80
SEAL	99.36±0.03	91.72±0.58	98.87±0.43	97.89±0.17	89.74±1.13	**97.15**±0.31	92.00±1.56
WP	99.65±0.02	93.01±0.12	98.49±0.70	98.46±0.01	89.48±0.02	**97.16**±0.01	92.62±0.01
LGLP	99.47±0.02	92.80±0.67	99.14±0.08	98.39±0.30	91.88±0.05	**97.15**±0.04	92.44±0.13
DSDN	**99.70 ±0.02**	**93.33±0.33**	**99.20±0.31**	**98.56±0.12**	**92.01±0.49**	**97.17±0.25**	**94.23±0.66**

Domain	*Biology*			*Infrastructure & Technology*		
Alg\Data	Yeast	PPI	HPD	USAir	Power	Router
CN	88.85±0.05	85.90±0.57	73.82±0.44	93.08±1.37	59.23±0.58	56.16±0.35
Jaccard	88.75±0.58	83.61±0.47	73.38±0.45	90.06±0.91	58.97±0.56	55.19±0.37
PA	82.31±0.87	88.47±0.35	82.19±0.40	88.67±1.57	44.50±0.73	48.01±1.17
Katz	91.98±0.77	89.75±0.40	86.36±0.35	92.39±1.61	64.77±1.48	38.69±0.97
RWR	92.61±0.75	91.14±0.36	87.64±0.38	94.44±1.55	65.17±1.53	38.75±0.96
AA	88.21±0.59	85.47±0.59	73.86±0.44	92.09±1.20	57.69±0.46	56.12±0.34
SEAL	97.53±0.18	**92.70±0.20**	93.59±0.27	96.08±0.58	87.77±0.87	96.37±0.74
WP	97.87±0.44	92.43±0.49	93.76±0.08	97.15±1.09	90.53±0.89	97.02±0.48
LGLP	97.62±0.03	92.40±0.08	93.16±0.30	97.09±0.14	85.40±0.22	95.66±0.09
DSDN	**98.06±0.27**	92.18±0.46	**93.87±0.09**	**97.65±0.75**	**92.52±0.47**	**97.19±0.52**

data as validation[3]. In other words, we encourage the random-walk to explore larger neighbors when starting from the anchor x or y. Three hidden FC-layers are used with dimension $(k, 64)$, $(64, 16)$, and $(16, 2)$. The model is optimized with Adam of an initial learning rate 0.01 and decay factor 0.001.

For all the methods, the focal link is removed from the enclosing subgraph of those positive link instances in training. The averaged model size for different models are listed in Table III. The detailed calculations can be found in the Supplementary Materials.

TABLE III
AVERAGE MODEL SIZE OF OUR METHOD AND GNN-BASED ONES.

Methods	Ours	SEAL [38]	WP [41]	LGLP [40]
Number of parameters	7.6k	56k	3,278k	21k
feature generation	0.2k	11k	12k	8k
fully-connected layers	7.4k	45k	3,266k	13k

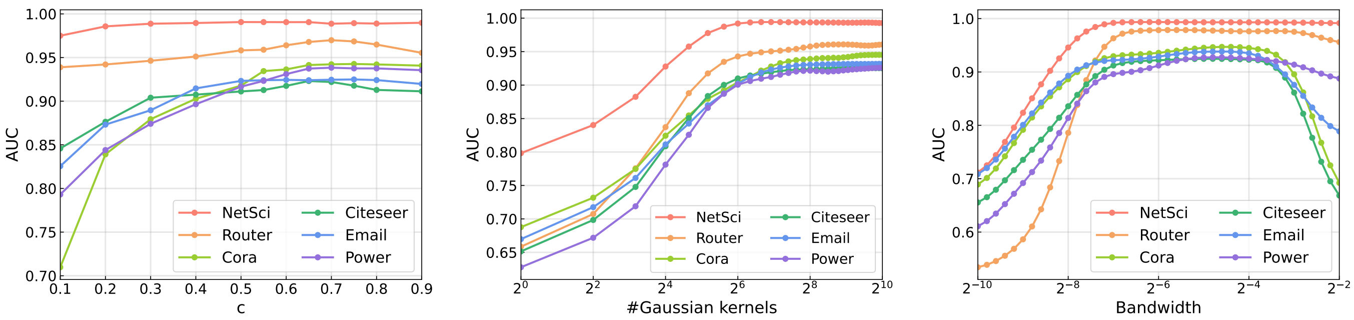
2) Comparative Results: Table I and II report average AUCs with standard deviation over 10 random training/test splits for synthetic and real networks, respectively. On synthetic networks, our approach has the highest AUC across all the different settings of model parameters, including regular, small-world, random and scale-free networks. On real-world networks, our approach achieves the highest AUC on 12 out of 13 networks, and the improvement over GNN-based models ranges in 2-5% for difficult networks (e.g., Cora and Power). *It is worth mentioning that our model is 400 times smaller than Walk-Pooling [41]*, the best GNN-based algorithm compared in this work (averagely) according to Table III. So the performance gains of our approach are considered encouraging, and clearly verify the effectiveness of the proposed link representation. Along with the compact model is the superior efficiency: as shown in Fig. 3, our method is 5-20 times faster than GNNs. Experiments were also conducted on networks with node attributes, and the corresponding results can be found in the Supplementary Materials.

3) Impact of the Hyper-parameters: In this section, we examine the impact of the hyper-parameters. Figure 4(a) shows the impact of the jump-back probability $(1-c)$ in the random-walk. As can be seen, when c grows larger and larger from 0.5 the performance of link prediction steadily improves and reaches a plateau. In fact, the optimal choice is around $c = 0.7$ for the networks investigated here, corresponding to a jump-back probability $1 - c = 0.3$. This means that on the one hand, the random-walk somehow remembers where it gets started (i.e., the anchor node x or y); on the other hand, it has the freedom to explore the whole subgraph, i.e., how the neighboring nodes of the link gather around the two anchors.

In Figure 4(b), we report the AUC versus the number of Gaussian kernels k. As can be seen, the AUC is relatively insensitive to the choice of k if it is large enough. In the experiments, we typically choose k around 100.

Figure 4(c) shows the AUC of link prediction over the bandwidth. The bandwidth controls the degree of smoothing, which is beneficial in avoiding overfitting. Here the AUC is averaged over 5 random initialization of parameters. As can be seen, when the bandwidth increases from a small value to 0.05, the predictive performance keeps improving, validating the usefulness of smoothing. When the bandwidth is too large, distinct point-cloud distributions will not be discriminated and so the performance will drop as expected. In practice, the

[3] Note that the validation set can also be used to choose other hyper-parameters, such as the number of hops, the initial bandwidth, and uniform-sampling interval used to initialize the Gaussian centers.



(a) The AUC of link prediction versus the choice of c that controls the re-start probability $(1-c)$.

(b) The AUC of link prediction versus the choice of the number of Gaussian kernels k.

(c) The AUC of link prediction versus the choice of the bandwidth parameter h.

Fig. 4. Impact of the jump-back probability $(1-c)$, the number of kernels k and the bandwidth parameter h on the performance of link prediction.

bandwidth parameter is optimized end-to-end.

B. Cross-Domain Network Analysis

Cross-domain studies can be valuable in uncovering diversified link formation patterns from miscellaneous real-world networks and finding the underlying commonalities. However, a universal representation for links from different networks is still challenging. On the one hand, GNN-based approaches such as SEAL [38] uses node2vec to extract topological features for each node, which can only be defined for each network independently; therefore the resultant link representation is also network-specific and cannot be compared across multiple networks. On the other hand, learning based approaches are more suitable for differentiating between positive and NULL links, while we are more interested in profiling the distribution of positive links across different networks.

Our link representation provides a universal platform to study and compare links from diverse fields simultaneously. First, the density profile (6) serves as a fixed-dimensional, well aligned representation that bears consistent physical meaning even across networks (Section III-B), by choosing the same set of Gaussian kernel centers $\{\boldsymbol{\mu}_j' s\}$ in the dipole plane across networks. Second, the link representation can be obtained in an unsupervised manner due to the strong inductive-bias in link modeling. Specifically, the $\boldsymbol{\mu}_j$'s in the dipole plane can be chosen as the clustering centers of the aggregated point-clouds without using class labels. By doing this, the resultant link representation faithfully preserves the distribution of diverse patterns of positive links. It allows building a global link-pattern atlas, based on which we have uncovered interesting common patterns of link formation, as well as hidden network similarities beyond their original domains.

Here we report a preliminary study. We select 10 networks covering 5 different domains from Table II, and randomly sample 3000 true links from each network. By using 2-hop enclosing subgraphs, each link is represented as k-dimensional density profile as in (6) with $k = 64$, and further normalized by the average subgraph size from each network. Based on this representation, we then visualize the 30,000 links through their tSNE embedding, thus presenting a global link-pattern landscape in Fig. 5 (panel-I). Based on this embedding, we can see that the link patterns from the 10 networks form three visual clusters (though with overlaps), discussed below.

Cluster-A (top): NetSci (co-author, dark red), Power (red), Citeseer (citation, brown), Cora (citation, orange) and GrQ (co-author, orange); the link distributions form a few zones (with overlaps) that are stacked layerwise in this cluster.

Cluster-B (middle): Email (dark green), protein networks Yeast (green) and HPD (green), and USAir (light green); the link distributions form curved manifold in this cluster.

Cluster-C (bottom): Facebook (blue); link distribution has many micro-clusters around the bottom of the landscape.

For visual clarity, links in each network are also plotted separately in Fig. 5 (panel-II). An intriguing observation is that networks from very different domains could exhibit strong similarity of link patterns and group together. What is the underlying "subject" of each cluster? In fact, networks in the same cluster share general commonalities in the formation and characteristics of link patterns. Next, we elaborate on these clusters and look into some representative link examples.

Bridge-style (Cluster-A). Many links serve as a bridge to connect two nodes that are otherwise many hops away, indicating decentralized connections across distant subnetworks. For example, in Power network, typically a limited number of transmission lines are built to cover as many stations as needed under efficiency and cost considerations. Citation networks like Cora and Citeseer include many different research areas and so the links also spread out. Similarly, links in co-author networks like NetSci and GrQ are distributed, too[4]. The point-clouds may look L-shaped or break into a few small groups, see examples A_1-A_6 in Fig. 5 (panel-III).

Radiation-style (Cluster-B). Links are mostly between a hub and a non-hub node, or central position of a star-shaped subnetwork to its periphery, like B_2, B_3, B_5 and B_6. This coincides well with the organizing principles of the networks falling in this cluster. For example, in protein networks (Yeast and HPD), it is widely acknowledged that a small number of hub nodes play a central role in the global network organization. In the USAir network, air flights are mostly between one

[4] Intriguingly, not all coauthor networks are dominated by decentralized link connections; instead, some coauthor networks may have more centralized collaboration similar to link patterns in Cluster-B, see the larger-scale case studies reported at the end of this section. We speculate the latter is due to the existence of a fraction of highly influential researchers in the area.

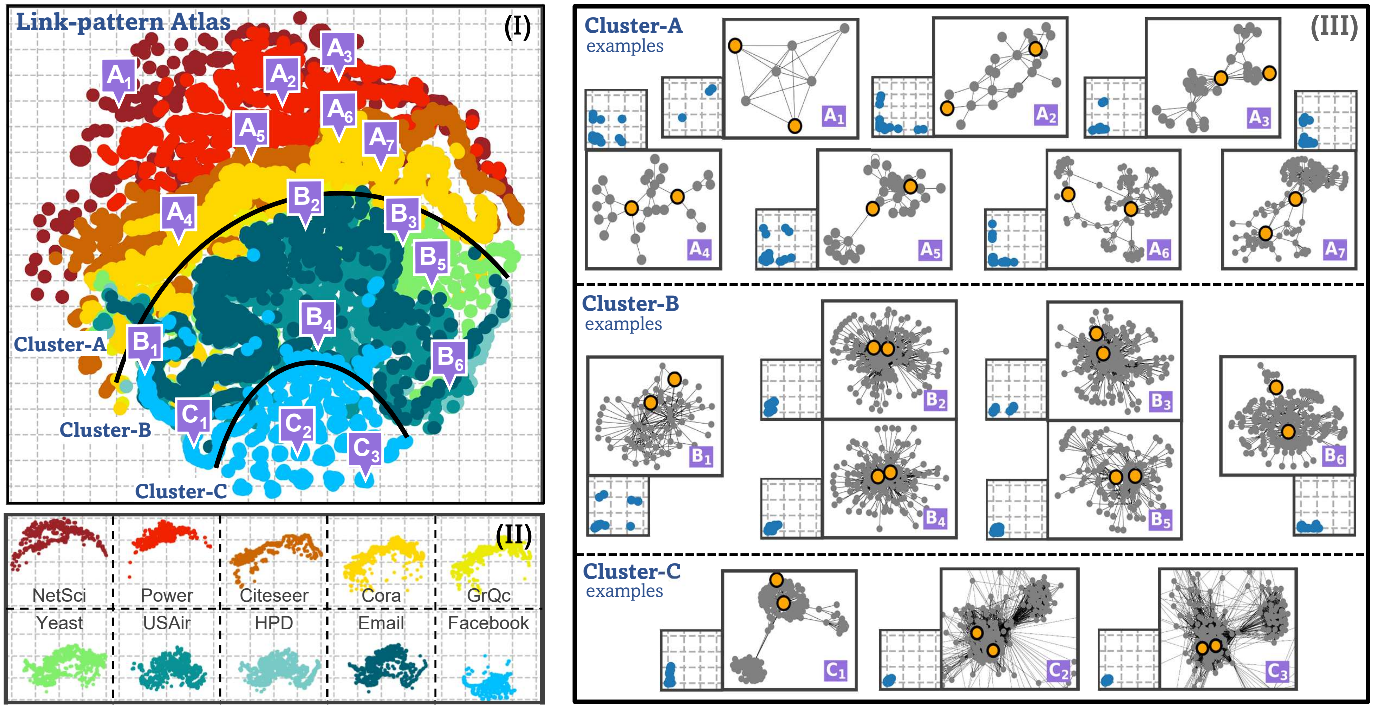


Fig. 5. A Cross-domain study of network links collected from a number of different science and engineering fields. Panel-I: link-pattern atlas including 30,000 links drawn randomly from 10 networks, embedded as 30,000 points by tSNE with one color for each network. Three main clusters are identified: **cluster-A: bridge style**, **cluster-B: radiation style**, and **cluster-C: community style**, each with a few representative examples that are marked as A_1-A_7, B_1-B_6, C_1-C_3 for case study. Panel-II: the tSNE embedding in Panel-I is plotted separately for each network to better visualize the distribution of the links in each network, allowing a quick examination of the similarity between the networks (in terms of link pattern distribution). Panel-III: enclosing-subgraph and point-cloud for 16 representative link examples; each subgraph is with gray nodes and two orange anchors; each point-cloud is with blue points and plotted in $[0, 0.3] \times [0, 0.3]$ of the dipole plane to better visualize details. Note that we have used 2-hop neighbors from each anchor node, so the resultant enclosing subgraph would allow a maximum pairwise distance up to 5-hops between nodes. Section V provides detailed discussions.

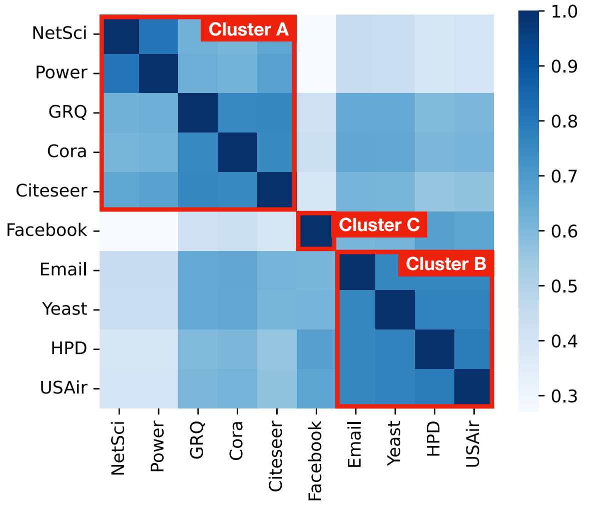


Fig. 6. Pairwise similarities of the 10 networks selected from Table I across 5 domains. The similarities are based on the distribution of the link-patterns from each network as measured through the proposed link representation.

of the transfer stations and a terminal station. For the Email network (the email communications from the users of Univ. Rovira i Virgili, Spain), many messages are between a central unit (office, dept. head, group lead) to individuals (faculty, staff, student), in a hierarchical, self-similar manner [6]. The point-clouds typically lean to one axis, i.e., the structural roles of the two anchor nodes are not symmetric to each other.

Community-style (Cluster-C). Notably, links in Facebook are found inside communities, like C_1, C_2 and C_3. Among them, link C_1 is between the central node and a boundary node of the community, i.e., its two anchors are structurally asymmetric; for link C_2 and C_3, the two anchors are structurally more symmetric, and so the point-cloud spreads along $y = x$ in the dipole plane. Another observation is that the enclosing subgraph typically involves multiple communities, indicating local communities in social networks that are inter-connected within a small number of hops.

Besides these representative cases, some transitional link patterns around cluster boundaries can also be found. For example, link B_1 is between cluster-B and cluster-C. Here, the two anchors of B_1 are structurally symmetric like cluster C, while the local enclosing subgraph only involves one community like cluster-B. For Link C_1, its two anchors are found inside a community where one is the bub node and the other is the boundary node, which is the key characteristic of cluster-B. On the other hand, the enclosing subgraph involves two communities, which is like cluster-C.

In Figure 6 we report the network-level spectral clustering based on the link patterns from each network. Detailed procedures can be found in the Supplementary Materials. As can be seen, the similarity matrix clearly reveals three diagonal blocks, corresponding to the three clusters discussed above.

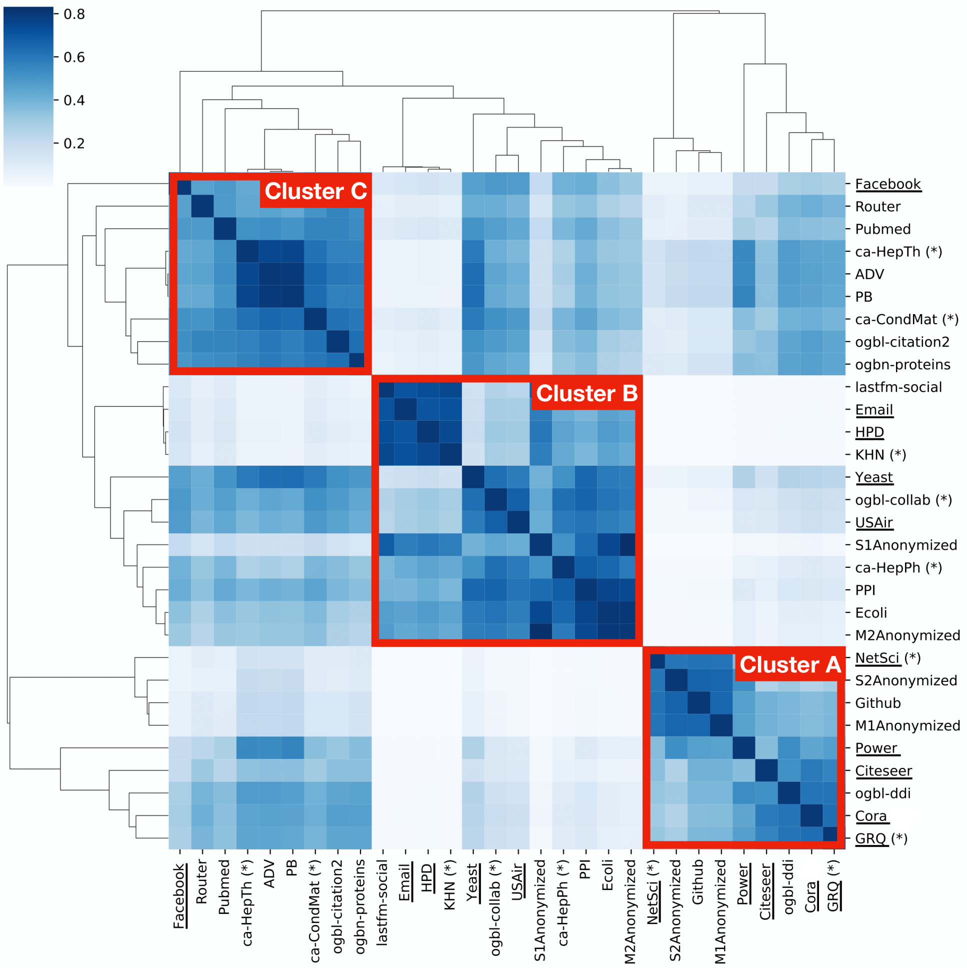


Fig. 7. Hierarchical clustering of 30 networks collected from a wide spectrum of disciplines. Three big clusters are identified and correspond to the three clusters A, B and C, as obtained in the Figure 4 of the main text. The 10 networks with underlines are those used in the comparative studies in Section V-B. Networks marked with (*) are scientific networks that encode collaborative relations among the researchers.

The cross-domain analysis based on the proposed link representation fosters a global understanding of how the link patterns from diversified application fields are distributed. It reveals intriguing network similarities that may otherwise be hidden. This universal platform could help promote new findings in network knowledge discovery, especially from the perspective of link-pattern characteristics or distributions. In Figure 7 we use 30 networks from a wider variety of areas and examine their network-level grouping based on the link-pattern distributions. The details of these networks can be found in the Supplementary Materials. A hierarchical clustering based on pairwise network similarities shows three dominant clusters, which are in accordance with the three clusters from the smaller-scale studies in Figure 6.

Here we take a closer look at the 6 scientific collaboration networks: GRQ, NetSci, KHN, ogbl-collab, ca-ConcMat, and ca-HepTh. An interesting observation is that these 6 collaboration networks are not in the same cluster. For example, GRQ and NetSci are in Cluster A (bridge style), KHN and ogbl-collab are in cluster B (radiation-style), and ca-HepTh and ca-CondMat are in cluster C (community-style). This reflects the diversity of the link-pattern distributions of networks even for those from the same domain, due to the difference in the organization principles. We speculate that the diversity of the collaboration networks originates from the different ways of collaborations in different scientific communities.

VI. Conclusion

We have introduced a novel topological representation for link modelling, prediction and cross-domain link pattern analysis. It transforms the local enclosing subgraph associated with each link into a fixed dimensional density distribution, which possesses desired expressiveness, interpretability, and cross-domain modelling power, opening up new possibilities for topological feature engineering in the task of link prediction and network analysis. Our experimental findings demonstrate that our method achieves a performance comparable or superior to that of GNN-based models, while simultaneously offering accelerated training speed and reduced model size. Moreover, our representation method establishes a versatile research platform for investigating cross-network topological information, which unravels interesting commonalities of link formation patterns across networks of very different nature.

A number of interesting directions will be pursued in our future studies. For example, we will extend the proposed link representation to more general scenarios like (sub)graph representation learning. We will also apply it to large-scale case studies to facilitate domain-specific knowledge discovery like brain functional networks and protein interaction networks. Finally, we plan to study how the proposed link representation is related to network control, especially in building accurate predictive models that can map the distribution of the link patterns of a network to its global behaviour or functionality.

References

[1] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, “Classes of small-world networks,” *Proceedings of National Academy of Sciences*, vol. 97, no. 21, pp. 11 149–11 152, 2000.

[2] M. Newman, “The structure of scientific collaboration networks,” *Proceedings of the National Acedemy of Sciences*, vol. 98, no. 2, pp. 404–409, 2001.

[3] A.-L. Barabási, *Network Science*. Cambridge University Press, 2016.

[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang, “Complex networks: Structure and dynamics,” *Physics Reports*, vol. 424, pp. 175 – 308, 2006.

[5] P. Holme and J. Saramäki, “Temporal networks,” *Physics reports*, vol. 519, no. 3, pp. 97–125, 2012.

[6] R. Guimerá, L. Danon, A. Díaz-Guilera, F. Giralt, and A. A., “Self-similar community structure in a network of human interactions,” *Physical Review E*, vol. 68, no. 3, p. 065103, 2006.

[7] S. Laughlin and T. Sejnowski, “Communication in neuronal networks,” *Science*, vol. 301, no. 5641, pp. 1870–1874, 2003.

[8] E. Hossain, Z. Han, and H. Poor, *Smart grid communications and networking*. Cambridge University Press, 2012.

[9] D. Mishra, T. Bepler, B. Teague, B. Berger, J. Broach, and R. Weiss, “An engineered protein-phosphorylation toggle network with implications for endogenous network discovery,” *Science*, vol. 373, pp. 623–630, 2021.

[10] L. Lü and T. Zhou, “Link prediction in complex networks: A survey,” *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, p. 1150–1170, Mar 2011.

[11] A. Clauset, C. Moore, and M. Newman, “Hierarchical structure and the prediction of missing links in networks,” *Nature*, vol. 453, pp. 98–101, 2008.

[12] R. R. Guimerá and S.-P. Marta, “Missing and spurious interactions and the reconstruction of complex networks,” *Proceedings of National Academy of Sciences*, vol. 106, no. 22073, 2009.

[13] D. Liben-Nowell and J. Kleinberg, “The link prediction problem for social networks,” in *International Conference on Information and Knowledge Management*, 2003, p. 556–559.

[14] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, “Networks formed from interdependent networks,” *Nature Physics*, vol. 8, pp. 40–48, 2012.

[15] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, and H. E. Stanleya, “Toward link predictability of complex networks,” *Proceedings of National Academy of Sciences*, vol. 112, no. 8, pp. 2325–2330, 2019.

[16] A. Ghasemian, H. Hosseinmardi, A. Galstyan, E. M. Airoldi, and A. Clauset, “Stacking models for nearly optimal link prediction in complex networks,” *Proceedings of National Academy of Sciences*, vol. 117, no. 38, p. 23393–23400, 2020.

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

[18] P. Jaccard, “Étude comparative de la distribution florale dans une portion des alpes et des jura,” *Bulletin del la Société Vaudoise des Sciences Naturelles*, vol. 37, no. 142, pp. 547–579, 1901.

[19] L. A. Adamic and E. Adar, “Friends and neighbors on the web,” *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[20] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 56, no. 18, pp. 3825–3833, 2012.

[21] W. R. Arney, “A refined status index for sociometric data,” *Sociological Methods & Research*, vol. 1, no. 3, pp. 329–346, 1973.

[22] C. Wang, V. Satuluri, and S. Parthasarathy, “Local probabilistic models for link prediction,” in *Seventh IEEE International Conference on Data Mining*, 2007, pp. 322–331.

[23] J. Neville, “Statistical models and analysis techniques for learning in relational data,” *Computer Science*, 2006.

[24] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu, “Stochastic relational models for discriminative link prediction,” *Advances in neural information processing systems*, vol. 19, 2006.

[25] A. Clauset, C. Moore, and M. E. J. Newman, “Hierarchical structure and the prediction of missing links in networks,” *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

[26] R. Guimerà and M. Sales-Pardo, “Missing and spurious interactions and the reconstruction of complex networks,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 52, pp. 22 073–22 078, 2009.

[27] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computers*, vol. 8, pp. 30–37, 2009.

[28] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps.” *Social Networks*, vol. 5, pp. 109–137, 1983.

[29] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[30] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[31] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 855–864.

[32] J. Leskovec, D. Huttenlocher, and J. Kleinberg, “Predicting positive and negative links in online social networks,” in *International Conference on World Wide Web*, 2010, pp. 641–650.

[33] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2009.

[34] M. Bronstein, J. Bruna, Y. LeCun, and A. Szlam, “Geometric deep learning: going beyond euclidean data,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[35] W. W. L. Hamilton, R. Ying, and J. Leskovec, “Representation learning on graphs: Methods and applications,” in *Proceedings of the Neural Information Processing Systems*, 2017, p. 1024–1034.

[36] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, “Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec,” in *ACM International Conference on Web Search and Data Mining*, 2018, p. 459–467.

[37] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” in *Proceedings of the Fifth International Conference on Learning Representations*, 2017.

[38] M. Zhang and Y. Chen, “Link prediction based on graph neural networks,” in *Proceedings of the 32nd Conference on Advances in Neural Information Processing Systems*, 2018.

[39] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, “An end-to-end deep learning architecture for graph classification,” in *AAAI Conference on Artificial Intelligence*, 2018, pp. 4438–4445.

[40] L. Cai, J. Li, J. Wang, and S. Ji, “Line graph neural networks for link prediction.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5103–5113, 2022.

[41] L. Pan, C. Shi, and I. Dokmanić, “Neural link prediction with walk pooling,” in *Proceedings of the Tenth International Conference on Learning Representations*, 2022.

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

[43] V. Martínez, F. Berzal, and J. Cubero, “A survey of link prediction in complex networks,” *ACM Comput. Surv.*, vol. 49, no. 4, dec 2016.

[44] G. Jeh and J. Widom, “Simrank: a measure of structural-context similarity,” in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[45] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Neural and Information Processing System*, 2013.

[46] B. Li, Y. Xia, S. Xie, L. Wu, and T. Qin, “Distance-enhanced graph neural network for link prediction,” in *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[47] H. Tong, C. Faloutsos, and J. Pan, “Fast random walk with restart and its applications,” in *Proceedings of the Sixth International Conference on Data Mining*, 2006, pp. 613–622.

[48] E. Parzen, “On estimation of a probability density function and mode,” *Applied Mathematics Letters*, vol. 33, p. 1065–1076, 1962.

[49] M. Wand, *Kernel Smoothing*. Chapman and Hall/CRC, 1995.

[50] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Reviews of Modern Physics*, vol. 74, no. 7, pp. 47–97, 2002.

[51] D. J. Watts and S. H. Strogatz, “Collective dynamics of ’small-world’ networks,” *Nature*, vol. 393, no. 6684, p. 440–442, 1998.

[52] G. Kossinets, “Effects of missing data in social networks,” *Soc. Networks*, vol. 28, pp. 247–268, 2006.

[53] A. L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[54] J. McAuley and J. Leskovec, “Learning to discover social circles in ego networks,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 2012, p. 539–547.

[55] V. Batagelj and A. Mrvar, “Pajek datasets,” URL: http://vlado.fmf.uni-lj.si/pub/networks/data/, 2006.

[56] M. E. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[57] J. Leskovec, J. M. Kleinberg, and C. Faloutsos, “Graph evolution: Densification and shrinking diameters,” *ACM Trans. Knowl. Discov. Data*, vol. 1, p. 2, 2006.

[58] C. Mering, R. Krause, B. Snel, M. Cornell, S. Oliver, S. Fields, and P. Bork, “Comparative assessment of large-scale data sets of protein–protein interactions,” *Nature*, vol. 417, no. 6887, p. 399–403, 2002.

[59] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, “Measuring isp topologies with rocketfuel.” *IEEE/ACM Transactions on networking*, vol. 12, no. 1, pp. 2–16, 2004.

[60] Y. Qi, Z. Bar-Joseph, and J. Klein-Seetharaman, “Evaluation of different biological data and computational classification methods for use in protein interaction prediction,” *Proteins: Structure, Function, and Bioinformatics*, vol. 63, no. 3, pp. 490–500, 2006.

[61] D. J. Watts and S. H. Strogatz., “Collective dynamics of ‘small-world’networks,” *Nature*, vol. 393, no. 6684, p. 440–442, 1998.

[62] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” *AI Magazine*, vol. 29, no. 3, p. 93, 2008.

[63] S. Peri, J. Navarro, R. Amanchy, T. Kristiansen, C. Jonnalagadda, V. Surendranath, V. Niranjan, B. Muthusamy, T. Gandhi, and M. e. a. Gronborg, “Development of human protein reference database as an initial platform for approaching systems biology in humans,” *Genome Research*, vol. 13, pp. 2363–2371, 2003.

[64] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graphs over time: Densification laws, shrinking diameters and possible explanations.” Association for Computing Machinery, 2005, p. 177–187.

[65] P. Massa, M. Salvetti, and D. Tomasoni, “Bowling alone and trust decline in social network sites,” in *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*. IEEE, 2009, pp. 658–663.

[66] R. Ackland, “Mapping the u.s. political blogosphere: Are conservative bloggers more prominent?” *BlogTalk*, 2005.

[67] K. Wang, Z. Shen, C. Huang, C.-H. Wu, Y. Dong, and A. Kanakia, “Microsoft Academic Graph: When experts are not enough,” *Quantitative Science Studies*, vol. 1, no. 1, pp. 396–413.

[68] J. C. Paetzold, J. McGinnis, S. Shit, I. Ezhov, P. Büschl, C. Prabhakar, A. Sekuboyina, M. Todorov, G. Kaissis, A. Ertürk, S. Günnemann, and B. Menze, “Whole brain vessel graphs: A dataset and benchmark for graph learning and neuroscience,” Tech. Rep., 2021.

[69] D. Szklarczyk, A. Gable, D. Lyon, A. Junge, S. Wyder, J. Huerta-Cepas, M. Simonovic, N. Doncheva, J. Morris, P. Bork, L. Jensen, and C. von Mering, “String v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets,” *Nucleic acids research*, vol. 47, pp. 607–613, 2018.

[70] B. Rozemberczki and R. Sarkar, “Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models,” in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM ’20)*. ACM, 2020, p. 1325–1334.

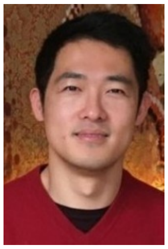
[71] M. Fire, R. Puzis, and Y. Elovici, “Organization mining using online social networks,” *Networks and Spatial Economics*, vol. 16, pp. 545–578, 2013.

[72] B. Rozemberczki, C. Allen, R. Sarkar, and x. Thilo Gross, “Multi-scale attributed node embedding,” *Journal of Complex Networks*, vol. 9, no. 1, pp. 1–22, 2021.

[73] D. Wishart, Y. Djoumbou, A. C. Guo, E. Lo, A. Marcu, J. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, N. Assempour, I. Iynkkaran, Y. Liu, A. Maciejewski, N. Gale, A. Wilson, L. Chin, R. Cummings, D. Le, and M. Wilson, “Drugbank 5.0: A major update to the drugbank database for 2018,” *Nucleic acids research*, vol. 46, 11 2017.

[74] M. Zhang, Z. Cui, S. Jiang, and Y. Chen, “Beyond link prediction: Predicting hyperlinks in adjacency space,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 04 2018.

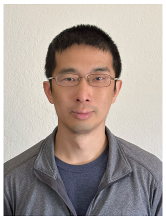
VII. Biography Section



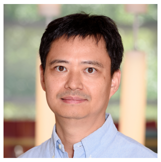
Kai Zhang Kai Zhang received his Ph.D. degree in Computer Science in 2008 from the Hong Kong University of Science and Technology. He is currently a professor in the School of Computer Science and Technology at East China Normal University. His research focuses on neural networks, machine learning and their applications in bioinformatics, brain science, time series analysis and complex networks. He is the recipient of NEC Labs America 2016 Business Contribution Award, and the KDD 2016 Best Paper Runner-up Award.

Junchen Shen Junchen Shen received his Bachelor’s degree in applied physics at Nanjing University of Posts and Telecommunications in 2021. He is currently pursuing his Master’s degree in the School of Computer Science and Technology at East China Normal University. His research interests include machine learning, graph neural networks, time series analysis and recommendation systems.

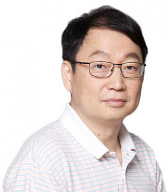
Gaoqi He Gaoqi He is currently a Professor in the School of Computer Science and Technology at East China Normal University. He received his PhD dearee from the State Key Laboratory of CAD & CG, Zhejiang University, China,in 2007. His research interests include computer graphics, computer vision and machine learning. His work aims to develop efficient and practical scene understanding and visual analysis algorithms with applications including video processing, crowd simulation, virtual reality and augmented reality.



Yu Sun Yu Sun received his Ph.D degree in Mechanical Engineering from University of Illinois, Urbana-Champaign in 2011, and M.S. Degree in Electrical Engineering from University of Notre Dame in 2006. His research interests include machine learning, distributed optimization, game theory, reinforcement learning and their applications.



Haibin Ling Haibin Ling received the B.S. and M.S. degrees from Peking University in 1997 and 2000, respectively, and the Ph.D. degree from the University of Maryland, College Park, in 2006. From 2000 to 2001, he was an assistant researcher at Microsoft Research Asia. From 2006 to 2007, he worked as a postdoctoral scientist at the University of California Los Angeles. In 2007, he joined Siemens Corporate Research as a research scientist; then, from 2008 to 2019, he worked as a faculty member at Temple University. In 2019, he joined Stony Brook University as a SUNY Empire Innovation Professor in the Department of Computer Science. His research interests include computer vision, augmented reality, medical image analysis, machine learning, and human computer interaction. He received Best Student Paper Award at ACM UIST (2003), Best Journal Paper Award at IEEE VR (2021), NSF CAREER Award (2014), Yahoo Faculty Research and Engagement Award (2019), and Amazon Machine Learning Research Award (2019). He serves or served as Associate Editors for IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), IEEE Trans. on Visualization and Computer Graphics (TVCG), Computer Vision and Image Understanding (CVIU), and Pattern Recognition (PR), and as Area Chairs for CVPR, ICCV, ECCV and WACV.



Hongyuan Zha Hongyuan Zha received his B.S. degree in Mathematics from Fudan University in 1984, and his Ph.D. in Scientific Computing from Stanford University in 1993. He was a faculty member of College of Computing at Georgia Institute of Technology from 2006 to 2020, and the Department of Computer Science and Engineering at Pennsylvania State University from 1992 to 2006. His current research interest lies in machine learning and its applications. He has won many prominent academic awards including Leslie Fox Prize (second prize,1991) awarded by the Institute of Mathematics and Applications (IMA), the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011) Best Student Paper Award (as advising Professor), the 26th NeurIPS Outstanding Paper Award (2013). Prof. Hongyuan Zha is a X.Q. Deng Presidential Chair Professor of The Chinese University of Hong Kong, Shenzhen and the Executive Dean of the School of Data Science.

Honglin Li Honglin Li received his bachelor degree in Chemical Engineering, and PhD degree in engineering mechanics at Dalian University of Technology in 2000 and 2005, respectively. As a post-doctor, he studied at Shanghai Institute of Materia Medica from July 2005 to May 2008. Since March 2008, he worked at School of Pharmacy in East China University of Science and Technology as a Professor of Medicinal Chemistry and Computational Chemistry. He has received a number of academic awards, including: the 13th Science and Technology Award for Chinese Youth (2013), National Youth Science and technology innovation leader (2015), Ten Thousand Talent Program Leading scientist (2017), National Science Fund for Distinguished Young Scholars (2018). His is interested in novel computational tools to rationalize and improve the efficiency, speed and cost-effectiveness of the drug discovery process. He has made some innovative achievements by combining dry/wet research approaches in drug design, drug interaction mechanism investigation and target identification.

Jie Zhang Jie Zhang received his Ph.D. from the Hong Kong Polytechnic University in 2008 and subsequently held a post-doctoral fellowship at the same institution from 2008 to 2010. Currently a professor at the Institute of Brain-inspired Intelligence at Fudan University, his research focuses on complex and dynamic systems, computational neuroscience, and brain disorders. His work is dedicated to developing interdisciplinary computational approaches to shed light on the principles of brain information processing and functional network organizations underlying various psychiatric disorders. He was a finalist for the 2007 Young Scientist Award in the field of physical/mathematical sciences.

Appendix A
Proof of Theorem

We will provide a sufficient theoretic condition for the point-clouds of two enclosing-subgraphs to be different. Here, we only consider equal-sized subgraphs, since if two subgraphs have different numbers of nodes, their layer-wise degree sequences must be different from each other, and so their point-clouds are also easily distinguishable. In this case, the proof of Theorem 1 becomes trivial and so will be skipped.

Given two enclosing subgraphs, G_1 and G_2, each with node set V_1 and V_2, unweighted (1/0), zero diagonal adjacency matrix $\mathbf{A}_1 \in \mathbb{R}^{n\times n}$ and $\mathbf{A}_2 \in \mathbb{R}^{n\times n}$, and anchor node x_1 and x_2, respectively. In each subgraph, the anchor will be the first node, followed by its 1st-order neighbors, and then 2nd-order neighbors, and so on. This is a natural partial ordering given the central status of the anchor. Note that, only when the point-clouds of the two subgraphs are different under all the feasible partial orderings, the two point-clouds will be different. The random-walk starting from each anchor can be defined as

$$\begin{aligned}\mathbf{p}_1 &= (\mathbf{I} - c\tilde{\mathbf{A}}_1)^{-1}\mathbf{e}\\ \mathbf{p}_2 &= (\mathbf{I} - c\tilde{\mathbf{A}}_2)^{-1}\mathbf{e},\end{aligned}$$

with $\mathbf{e}$ an n-dimensional one hot indicator vector whose first entry is 1 (corresponding to the anchor), and $\tilde{\mathbf{A}}_1$ and $\tilde{\mathbf{A}}_2$ the transition probability matrix by normalizing the columns of $\mathbf{A}_1$ and $\mathbf{A}_2$, respectively, so that each column sums up to 1.

Define $\mathbf{B}_1 = \mathbf{I} - c\tilde{\mathbf{A}}_1$ and $\mathbf{B}_2 = \mathbf{I} - c\tilde{\mathbf{A}}_2$. Let $\mathbf{B}_1^*$ and $\mathbf{B}_2^*$ be the adjugate matrix of $\mathbf{B}_1$ and $\mathbf{B}_2$. Then it is easy to see that the first entry of $\mathbf{p}_1$ and $\mathbf{p}_2$ can be written as

$$\mathbf{p}_1[1] = \frac{\mathbf{B}_1^*[1,1]}{|\mathbf{B}_1^*|},\ \ \mathbf{p}_2[1] = \frac{\mathbf{B}_2^*[1,1]}{|\mathbf{B}_2^*|}$$

Now we examine the j-th entry of $\mathbf{p}_1$ and $\mathbf{p}_2$, with $j \neq 1$. They can be written as

$$\mathbf{p}_1[j] = \frac{\mathbf{B}_1^*[j,1]}{|\mathbf{B}_1^*|},\ \ \mathbf{p}_2[j] = \frac{\mathbf{B}_2^*[j,1]}{|\mathbf{B}_2^*|}$$

If we have the following condition

$$\frac{\mathbf{B}_1^*[1,1]}{\mathbf{B}_1^*[j,1]} \neq \frac{\mathbf{B}_2^*[1,1]}{\mathbf{B}_2^*[j,1]}$$

Then equivalently

$$\frac{\mathbf{p}_1[1]}{\mathbf{p}_1[j]} \neq \frac{\mathbf{p}_2[1]}{\mathbf{p}_2[j]}$$

In this case, we must have $\mathbf{p}_1 \neq \mathbf{p}_2$ under the partial ordering specified, and the proof is complete.

Appendix B
Proof of Theorem 2

Let $\mathbf{P}_1$ and $\mathbf{P}_2$ be two equal-sized point-clouds, and they only have one pair of non-overlapping points, $\mathbf{u}$ and $\mathbf{v}$, while all the remaining $n-1$ points from the two point-set overlap exactly. This setting allows to focus on the impact of the bandwidth parameter on the difference of the distributions of the two point-clouds. Let $\boldsymbol{\mu}_j's$ be the set of k sensors, and the distance between $\mathbf{u}$ and $\mathbf{v}$ to the sensors can be written as the following distance vector

$$\begin{aligned}\mathbf{d}^u &= \left[\|\mathbf{u}-\boldsymbol{\mu}_1\|^2, \|\mathbf{u}-\boldsymbol{\mu}_2\|^2, \ldots, \|\mathbf{u}-\boldsymbol{\mu}_k\|^2\right]\\ \mathbf{d}^v &= \left[\|\mathbf{v}-\boldsymbol{\mu}_1\|^2, \|\mathbf{v}-\boldsymbol{\mu}_2\|^2, \ldots, \|\mathbf{v}-\boldsymbol{\mu}_k\|^2\right]\end{aligned}$$

we then define

$$\mathbf{w}^u = \exp\left(-\frac{\mathbf{d}^u}{2h^2}\right) \in \mathbb{R}^{1\times k},$$

$$\mathbf{w}^v = \exp\left(-\frac{\mathbf{d}^v}{2h^2}\right) \in \mathbb{R}^{1\times k}$$

Here $\mathbf{w}_u$ and $\mathbf{w}_v$ encode the similarity from $\mathbf{u}$ and $\mathbf{v}$ to the set of k sensor. It can be easily shown that the distance between the two density profiles reduces to the distance between the non-overlapping pair of points, $\mathbf{u}$ and $\mathbf{v}$, as

$$\|\mathbf{F}_1 - \mathbf{F}_2\|^2 = \left\|\frac{\mathbf{w}^u}{|\mathbf{w}^u|_1} - \frac{\mathbf{w}^v}{|\mathbf{w}^v|_1}\right\|^2. \tag{7}$$

Let's first study the unnormalized term $\mathbf{w}^u - \mathbf{w}^v$, as

$$\mathbf{w}^u - \mathbf{w}^v = \left[\exp\left(-\frac{\mathbf{d}_j^u}{2h^2}\right) - \exp\left(-\frac{\mathbf{d}_j^u}{2h^2}\right)\right]_{j=1,2,\ldots,k}$$

Each dimension can be bounded as follows according to the mid-value theorem in calculus

$$\frac{|\mathbf{d}_j^u - \mathbf{d}_j^v|}{2h^2} \cdot \min\left(\exp(-\frac{\mathbf{d}_j^u}{2h^2}), \exp(-\frac{\mathbf{d}_j^v}{2h^2})\right) \leq$$

$$\left|\mathbf{w}_j^u - \mathbf{w}_j^v\right| \leq \frac{|\mathbf{d}_j^u - \mathbf{d}_j^v|}{2h^2} \cdot \max\left(\exp(-\frac{\mathbf{d}_j^u}{2h^2}), \exp(-\frac{\mathbf{d}_j^v}{2h^2})\right)$$

For the convenience of notation, we define

$$\mathbf{d}_j^{\text{min}} = \min\left(\mathbf{d}_j^u, \mathbf{d}_j^v\right),\ \mathbf{d}_j^{\text{max}} \quad = \max\left(\mathbf{d}_j^u, \mathbf{d}_j^v\right),\ \mathbf{d}_j^{\text{uv}} = \left|\mathbf{d}_j^u - \mathbf{d}_j^v\right|$$

Then the bound can be simplified as

$$\frac{\mathbf{d}_j^{\text{uv}}}{2h^2} \cdot \exp\left(-\frac{\mathbf{d}_j^{\text{max}}}{2h^2}\right) \leq \left|\mathbf{w}_j^u - \mathbf{w}_j^v\right| \leq \frac{\mathbf{d}_j^{\text{uv}}}{2h^2} \cdot \exp\left(-\frac{\mathbf{d}_j^{\text{min}}}{2h^2}\right) \tag{8}$$

Having obtained the bounds along each dimension $|\mathbf{w}_j^u - \mathbf{w}_j^v|$, we can then normalize it by $|\mathbf{w}^u|_1$ and $|\mathbf{w}^v|_1$ to obtain the bounds of $\|\mathbf{F}_1 - \mathbf{F}_2\|$. To do this, we can simply use the smaller one and the larger one of the two ℓ_1-norms to bound the denominator of the two sides; then we square the bounds along each dimension, sum them up, and finally obtain Theorem 2.

Next, we show that we can further tighten the lower and upper bounds in (8), which was used instead in the plot of Figure 3 because it looks cleaner and would allow the readers to observe the interesting trend more easily. For the upper bound, since the absolute difference between two positive numbers must also be upper bounded by the larger one of the two, we can tighten the upper-bound as

$$\left|\mathbf{w}_j^u - \mathbf{w}_j^v\right| \leq \exp\left(-\frac{\mathbf{d}_j^{\text{min}}}{2h^2}\right) \cdot \min\left(\frac{\mathbf{d}_j^{\text{uv}}}{2h^2}, 1\right)$$

For the lower bound, we will also tighten it by observing that the difference between two non-negative numbers that are significantly different in their magnitude should be more closely lower-bounded. Let we define

$$\lambda_j = \frac{\exp\left(-\frac{\mathbf{d}_j^{\min}}{2h^2}\right)}{\exp\left(-\frac{\mathbf{d}_j^{\max}}{2h^2}\right)} = \exp\left(-\frac{\left(\mathbf{d}_j^{\min} - \mathbf{d}_j^{\max}\right)}{2h^2}\right) \tag{9}$$

which is the ratio of the maximum and minimum of the j-th entry in $\mathbf{w}^{\mathrm{u}}$ and $\mathbf{w}^{\mathrm{v}}$. Note that $\lambda_j \geq 1$. Then the lower bound can be tightened as follows

$$\begin{aligned}\left|\mathbf{w}_j^u - \mathbf{w}_j^v\right| &\geq \max\left(\frac{\mathbf{d}_j^{\mathrm{uv}}}{2h^2}\cdot\exp\left(-\frac{\mathbf{d}_j^{\max}}{2h^2}\right), (\lambda_j - 1)\exp\left(-\frac{\mathbf{d}_j^{\max}}{2h^2}\right)\right)\\ &= \exp\left(-\frac{\mathbf{d}_j^{\max}}{2h^2}\right)\cdot\max\left(\frac{\mathbf{d}_j^{\mathrm{uv}}}{2h^2}, (\lambda_j - 1)\right)\end{aligned}$$

So we have a tightened version of the bounds in (8), as

$$\begin{aligned}&\sum_j \exp\left(-\frac{\mathbf{d}_j^{\max}}{h^2}\right)\cdot\max\left(\frac{\mathbf{d}_j^{\mathrm{uv}}}{2h^2}, \lambda_j - 1\right)^2 \leq\\ &\|\mathbf{w}^u - \mathbf{w}^v\|^2 \leq \sum_j \exp\left(-\frac{\mathbf{d}_j^{\min}}{h^2}\right)\cdot\min\left(\frac{\mathbf{d}_j^{\mathrm{uv}}}{2h^2}, 1\right)^2.\end{aligned}$$

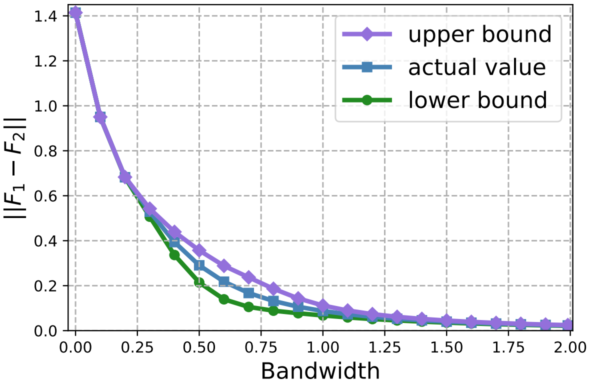


Fig. 8. The distance between two density profiles, and its tightened lower/upper bounds, drop with the bandwidth h in the toy example. This quantifies the smoothing effect of the Gaussian kernel.

To verify the correctness of the bounds, we use a simple data example that satisfies the condition stated in Theorem 2.1. The two equal-sized point-clouds are chosen as uniform points in $[0,1]^2$, and they only have one pair of non-overlapping points, i.e., $\mathbf{u} = [0.2, 0.5]^\top$ and $\mathbf{v} = [0.7, 0.5]^\top$.The Gaussian sensor locations $\boldsymbol{\mu}_j$'s are chosen as uniform grids in the dipole plane with interval 0.25. This setting allows us to minimize the impact of the point-cloud distribution and the sensor locations, and focus only on the role of the bandwidth h in modulating the distance between the distribution of two point-clouds. As can be seen from Figure 8, the actual distance and its lower and upper bounds all decay steadily with the bandwidth h.

Appendix C
Details of the Network Data

1) Synthetic Networks: We have adopted the BA-model and the WS-model to generate synthetic networks, in order to test the performance of different link prediction algorithms. The parameters of the two models are specified as follows.

The Watts–Strogatz model produces graphs with small-world properties. It starts from a ring-shaped graph with m nodes, where each node is connected with $\frac{K}{2}$ neighbors on both sides. For every node, pick the $\frac{K}{2}$ links connecting to its rightmost neighbors and rewire them (replace them with a random node) with probability β, while avoiding self-loops and duplication. We chose three sets of model parameters, namely $\beta = 0, 0.5, 0.8$, corresponding to a regular ring, a totally random network, and a small-world network.

The Barabási–Albert (BA) model generates random scale-free networks with preferential attachment. It begins with an initial network of m_0 nodes. New nodes are added one at a time. Each new node is connected to $m \leq m_0$ existing nodes with a probability p_i proportional to the number of links that the existing nodes already have, namely $p_i = k_i / \sum_j k_j$, where k_i is the degree of node i and the sum is made over all pre-existing nodes j. We chose $m = m_0 = 1, 3, 5$ as small integers so that the resultant degree distributions are scale free .

2) Real-world Networks: We used 13 popular benchmark datasets whose key statistics are in Table IV, detailed below. Facebook [54]: social network from facebook with 4039 users and 88234 edges; USAir [55]: US air transportation network with 332 airports and direct flights; NetSci [56]: collaboration network among 1589 researchers from a variety of fields in network science; GRQ [57]: collaboration network from researchers of general relativity and quantum cosmology in arXiv from 1993 to 2003; Yeast [58]: a protein-protein interaction network in yeast with 2,375 nodes and 11,693 edges; Router [59]: a router-level Internet with 5,022 nodes and 6,258 edges; PPI [60]: protein-protein interaction in human tissues with 3890 nodes and 38292 edges; Power [61]: the power grid of the Western States of the U.S, with 4941 nodes (a generator, a transformator or a substation), and 6594 links (high-voltage power supply line); CiteSeer [62]: citation network with 3312 scientific publications with 4732 links. Cora [62]: citation network with 2708 publications and 5429 links; Pubmed [62]: citation network with 19717 publications of diabete research with 44338 links; HPD [63]: protein-protein interaction about human protein in health and disease with 8756 nodes and 32331 edges; Email [6]: the email communication network at the University Rovira i Virgili in Tarragona in the south of Catalonia in Spain. Nodes are users and each edge represents that at least one email was sent.

Appendix D
Distance between two point-cloud Distribution

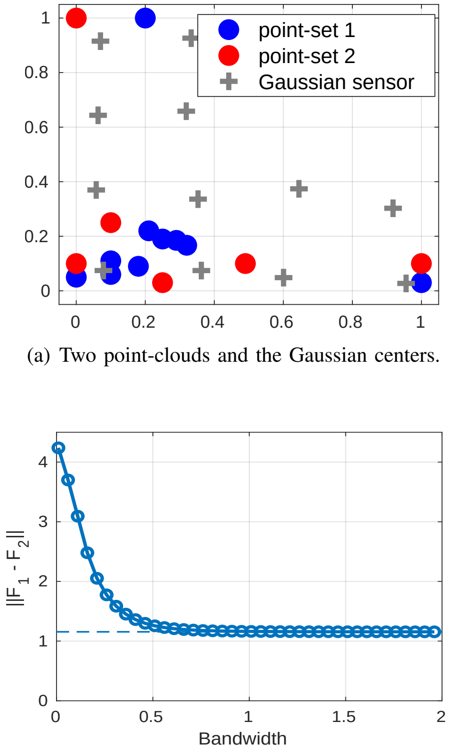
In the case of more complicated point-cloud distribution or different point-cloud sizes, an explicit bound could be hard to derive. However, empirically, we find that the smoothing effect of the bandwidth can still be observed. For example, when the bandwidth is large enough, the density-profile $\mathbf{F}_{xy}$ will be a vector with constant entry $\frac{n_{xy}}{k}$, and so the distance between two density profiles will approach $k\sqrt{|1/n_1 - 1/n_2|}$, which would be zero if the two point-clouds have the same number of nodes, i.e., $n_1 = n_2$. When n_1 and n_2 differ significantly,

TABLE IV

SUMMARY OF DATA-SETS AND NETWORK STATISTICS.

Network	#nodes	#links	link ratio	avg degree	clus. coeff.
Facebook	4039	88234	0.0108	43.69	0.6055
USAir	332	2126	0.0386	12.8	0.6252
Cora*	2708	5278	0.00144	3.90	0.2406
NetSci	1589	2742	0.00217	3.45	0.6377
Citeseer	3327	4552	0.00083	2.75	0.1414
PB	1222	16714	0.0224	27.4	0.3202
Pubmed	19717	44324	0.0023	4.50	0.0601
Yeast	2375	11693	0.00415	9.85	0.3057
PPI	3890	37845	0.005	19.5	0.1464
Router	5022	6258	0.0049	2.49	0.0115
Power	4941	6594	0.0054	2.70	0.0801
C.ele	297	2148	0.0487	14.5	0.2923
GRQ	5241	14484	0.00105	5.53	0.5297
BUP	105	441	0.080	8.40	0.4875
HPD	8756	32331	0.0084	7.38	0.1130
Email	1133	5451	0.0085	9.62	0.2203

the distance between the two point-cloud distributions will still approach the constant, but it will be non-zero.



(a) Two point-clouds and the Gaussian centers.

(b) The distance $\|\mathbf{F}_1 - \mathbf{F}_2\|$ between two distributions.

Fig. 9. The relation between $\|\mathbf{F}_1 - \mathbf{F}_2\|$ and the bandwidth for two point-clouds with different numbers of nodes and more complicated contributions.

In Figure 9, we provide an empirical example showing the relation between $\|\mathbf{F}_1 - \mathbf{F}_2\|$ and the bandwidth h for two point-clouds with different numbers of nodes and distributions. As shown in Figure 9(a), the two point-sets have more complicated distributions than the example used in Theorem 2.1 (in that scenario there is only one non-overlapping pair of points from the two point-sets and all the rest points do overlap exactly). As can be seen in Figure 9(b), the distance still again with the bandwidth h, but would converge to a non-zero value when the bandwidth is large enough.

APPENDIX E
DETAILS OF THE MODELS

For SEAL model, the hop of the enclosing subgraph is chosen as 1 or 2. An Adam optimizer with initial learning rate 1e-5 is used. Each node is chosen as the latent embedding of node2vec (128-dimensional), concatenated with the hop distance to the two anchors of a target link m (which equals to the number of unique structural roles of the nodes to the two anchors, as computed by the node labeling algorithm DRNL, and averagely $m = 100$). Therefore the node features will have a dimension 128+m+1. Altogether 4 GNN layers are used, with hidden dimension (128+m+1, 32), (32,32), (32,32), (32,1). Finally, two 1D-Convolutional layers are used, where the kernel size and out-put dimensions are (97,16) and (5,32), respectively. The dimension of the MLP is (352,128) and (128,2). In case m is 100, the number of parameters of the model is (128+100+1)*32+32*32+32*32+32*1+97*16+5*32+352*128 +128*2 = 56,432.

For Walk-Pooling model, the hop of the enclosing subgraph is chosen as 2. An Adam optimizer with initial learning rate 5e-5 is used. The representation for each node is chosen as 1, and a random-walk based on the graph topology is adopted. Altogether 2 GNN layers are used, with hidden dimension (32,32) and (32,32). Key matrices in attention module have dimension (96,32) and (32,64), the same goes for query matrices. The dimension of the MLP is chosen as (72,1440),(1440,1440), (1440,720), (720,72), (72,1), which is a large model. The number of parameters of the model is 32*32+32*32+96*32*2+32*64*2+72*1440+1440*1440+1440 *720+720*72+72*1=3,278,280.

For LGLP model, the hop of the enclosing subgraph is chosen based on the validation set. An Adam optimizer with initial learning rate 5e-3 is used. The representation for each node is chosen as the latent embedding obtained through DRNL. Altogether 3 GNN layers are used, with hidden dimension *(2(m+1),32), (32,32), (32,32), where m is the number of node-labeling obtained through DRNL as in [38]. The dimension of the MLP is chosen as (96,128) and (128,2). In case m is chosen as 100, the number of parameters of the model is 2*(100+1)*32+32*32+32*32+96*128+128*2=21,056.

APPENDIX F
AVERAGE LINK PREDICTION AUC ON NETWORKS WITH NODE ATTRIBUTES.

TABLE V

AVERAGED AUC USING BOTH GRAPH TOPOLOGY AND NODE ATTRIBUTE.

Data	Citeseer	Cora	Pubmed	PPI
SEAL	87.61±1.12	88.72±1.36	96.13±0.31	91.81±0.36
W-P	92.39±0.64	95.14±0.21	98.64±0.04	93.09±0.45
LGLP	90.89±0.88	94.63±0.97	98.12±0.22	91.10±0.24
DSDN	**92.46±0.70**	**95.43±0.65**	**98.77±0.61**	**93.25±0.27**

Table V we report the link prediction results on 4 networks with node attributes. We can see that our model has a very competitive performance when compared with GNN based methods. This shows that it has a strong advantage in combining both the network structure and attribute information.

Appendix G
Network-Level Similarity and Grouping

With a fixed-dimensional link representation, we can model the link pattern distribution of different networks and examine their similarity as follows. Let we have L networks $\mathcal{G}_p$'s each with n_p links whose representation or embedding is denoted as $\mathbf{z}_j^{(p)}$ for $j = 1, 2, ..., n_l$. Then the similarity between two networks can be assessed by the distance of their respective link-pattern distributions. To achieve this, We first compute the link-level distance between the n_p and n_q links of the two networks, $\mathcal{G}_p$ and $\mathcal{G}_q$, as an $\mathbb{R}^{n_p \times n_q}$ distance matrix $\mathbf{W}^{pq}_{[i,j]} = \|\mathbf{z}_i^{(p)} - \mathbf{z}_j^{(q)}\|^2$. Then we compute the average distance of the k-smallest entries in each row and column of $\mathbf{W}^{pq}$, separately, and sum them up together, as

$$Dis(\mathcal{G}_p, \mathcal{G}_q) = \frac{1}{2}\left(\frac{1}{n_p}\sum_{i=1}^{n_p}\overline{\mathbf{W}^{pq}_{[i,:](k)}} + \frac{1}{n_q}\sum_{j=1}^{n_q}\overline{\mathbf{W}^{pq}_{[:,j](k)}}\right) \quad (10)$$

The resultant $Dis(\mathcal{G}_p, \mathcal{G}_q)$ is the averaged k-nearest neighbor distance from one link in $\mathcal{G}_p$ to the links in $\mathcal{G}_q$, plus the averaged k-nearest neighbor distance from one link in $\mathcal{G}_q$ to the links in $\mathcal{G}_p$. The larger the value, the more different the link-pattern distributions of the two networks. The link-pattern distribution is a stable statistical measure to quantify the similarity between networks in terms of how their links are organized collectively.

Appendix H
Details of the extra networks

The extra networks included are listed as follows: ca-HepTh [64]: collaboration network from researchers of high energy physics theory in arXiv (Jan. 1993 to Apr. 2003); ca-HepPh [64]: collaboration network of high energy physics phenomenology researchers in arXiv (Jan. 1993 to Apr. 2003); ca-CondMat [57]: collaboration network from researchers of condense matter physics in arXiv (Jan. 1993 to Apr. 2003); ADV [65]: social network from advogato with 5155 users and 39285 edges; PB [66]: network of 16714 hyperlinks between 1222 political blogs in the United States of America; ogbl-citation2 [67]: citation network between a subset of papers extracted from MAG [68]; ogbn-proteins [69]: protein-protein interaction network extracted from 8 species; fm-social [70]: social network from LastFM with 7624 users and 27806 edges; ogbl-collab [67]: a subset of the collaboration network between authors in MAG [68]; S1Anonymized [71]: anonymized social network with 320 users and 2368 edges; S2Anonymized [71]: anonymized social network with 165 users and 725 edges; M1Anonymized [71]: anonymized social network with 1429 users and 19357 edges; M1Anonymized [71]: anonymized social network with 1300 users and 9662 edges; Github [72]: social network collected from github with 37700 users and 289003 edges; ogbl-ddi [73]: a drug-drug interaction network in which each node represents an FDA-approved or experimental drug; Ecoli [74]: pairwise reaction relation of metabolites in E.coli with 1805 nodes and 15660 edges; KHN: collaboration network with 3772 nodes and 12718 edges (https://noesis.ikor.org/datasets).