Link Prediction through Deep Learning

Xu-Wen Wang[1], Yize Chen[2], Yang-Yu Liu[1,3]*

[1]Channing Division of Network Medicine, Brigham and Women's Hospital and Harvard Medical School, Boston, MA 02115, USA.

[2]Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA.

[3]Center for Cancer Systems Biology, Dana-Farber Cancer Institute, Boston, MA, 02115, USA.

*Correspondence to: yyl@channing.harvard.edu

Abstract: Inferring missing links or predicting future ones based on the currently observed network is known as the link prediction problem, which has tremendous real-world applications. Indeed, a successful link prediction method will substantially reduce the experimental effort required to establish the topology of a network (such as the protein-protein interaction network or the drug-target interaction network). It will also accelerate mutually beneficial interactions (such as potential friendship on social media) that would have taken much longer to form serendipitously. Numerous methods have been proposed to solve this classical problem. Yet, existing methods are typically designed for undirected networks, and their performances differ greatly for networks from different domains. Here, by representing the adjacency matrices of networks as binary images and leveraging the power of deep generative models in computer vision, we developed a new link prediction method, which works for general directed or undirected complex networks. We applied this method to various real networks, finding that overall it shows superior performance against existing methods.

Networks have become an invaluable tool for describing the architecture of various complex systems, be they of technological, biological, or social in nature [1–3]. Mathematically, any real-world network can be represented by a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{1,2,\cdots,N\}$ is the node set and $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ is the link set. A link, denoted as a node pair (i,j) with $i,j\in\mathcal{V}$, represents certain interaction, association or physical connection between nodes i and j, which could be either directed or undirected, weighted or unweighted. For many systems (especially biological systems), the discovery and validation of links require significant experimental effort. Consequently, many real-world networks mapped out so-far are substantially incomplete [4,5]. For example, a recent estimate indicates that in human cells the available protein-protein interaction maps cover less than 20% of all potential protein-protein interactions [6]. How to tease out the missing links based on the discovered ones? Moreover, many systems (especially social systems) are very dynamic, as new links are added to the network over time. How to predict the likelihood of a future interaction between two currently unconnected nodes based on the current snapshot of the network? Both problems are commonly known as the link prediction problem [7–10].

An accurate link prediction method will substantially reduce the experimental effort required to establish the network's topology and/or accelerate mutually beneficial interactions that would have taken much longer to form serendipitously. Obviously, link prediction has many real-world applications [11,12]. In biomedicine, link prediction can be used to infer protein-protein interactions or drug-target interactions [13–15]. In e-commerce, it can help build better recommender systems, e.g., Amazon's "people who bought this also bought" feature [16,17]. On social media, it can help build potential connections such as the "people you may know" feature on Facebook and LinkedIn [18]. In criminal intelligence analysis, link prediction can assist in identifying hidden co-participation in illicit activities [19].

Numerous methods, such as similarity-based algorithms [20–22], maximum likelihood algorithms [23], and probabilistic models [24–26], have been developed to solve the link prediction problem. Yet, those methods are typically designed for undirected networks. Also, their performances differ greatly for networks from different domains. Note that, to quantify the performance of any link prediction method, the standard AUC statistic, i.e., the area under the receiver operating characteristic curve [7,23], is typically employed. Basically, we divide the link set $\mathcal{E}$ into two parts: (i) a fraction f of links as the test or probe set $\mathcal{E}^{\mathrm{P}}$, which will be removed from the network; and (ii) the remaining fraction $(1-f)$ of links as the training set $\mathcal{E}^{\mathrm{T}}$, which will be used to recover the removed links. The AUC statistic is defined to be the probability that a randomly chosen link in the probe set $\mathcal{E}^{\mathrm{P}}$ (a true positive) is given a higher score by the link prediction method than that of a randomly chosen nonexistent link (a true negative). Apparently, $\mathrm{AUC} = 0.5$ serves as the baseline performance of random guess, and the degree to which the AUC exceeds 0.5 indicates how much better a link prediction method is than random guess.

We still lack a powerful link prediction method that works for general directed or undirected complex networks, regardless of their technological, biological, or social nature. Here, we fill this gap by developing a novel link prediction method that works for generic complex networks. Our key idea is to treat the adjacency matrix of a network as the pixel matrix of a binary image. In other words, present (or absent) links will be treated as pixels of value 0 (or 1), respectively. By perturbing the original input in many different ways through randomly removing present links, we obtain a pool of perturbed input images. Those perturbed images will be fed into a deep generative model to create fake images that look similar to the input ones. Those fake images (networks) will be used as training set for link prediction in the original image (network).

For the deep generative model (DGM), here we leverage generative adversarial networks (GANs) that consist of two deep artificial neural networks (the *generator* and the *discriminator, respectively*) contesting with each other in a game theory framework [27,28]. The generator takes random noise from a known distribution as input and transforms them into fake images through a deconvolutional neural network. The discriminator is a binary classifier (based on a convolutional neural network), which determines whether a given image looks like a real one from the input dataset or like a fake one artificially created by the generator. Over the course of training iterations, the discriminator learns to tell real images from fake ones. At the same time, the generator uses feedback from the discriminator to learn how to produce convincing fake images to fool the discriminator so that it can't distinguish from real.

To demonstrate our DGM-based link prediction, let's consider a toy example: a small directed network of 28 nodes and 118 links, whose adjacency matrix looks like a binary image of letter **E** with 12 missing pixels (Fig.1). First, we create M perturbed binary images by randomly removing a fraction q of pixels of value 0 (i.e., those present links) from the original image (network). Second, we use the M perturbed binary images as input to train GANs, which will eventually generate S fake grayscale images that look similar to the input ones. (In this example we choose $M = 5{,}000$, $q = 0.1$, and $S = 500$.) Note that the existent likelihood of the link between nodes i and j, denoted as α_{ij}, in the corresponding fake network is simply given by $\alpha_{ij} = 1 - P_{ij}$, where P_{ij} is the rescaled pixel value (ranging from 0 to 1) in each fake grayscale image. Finally, we take the average value $\langle\alpha_{ij}\rangle = 1 - \langle P_{ij}\rangle$ over all the S fake images to get the overall existent likelihood of the link (i, j). Note that in this toy example all the 12 missing links display higher α_{ij} than that of nonexistent links, so they are successfully recovered.

Fig.1 may remind us the classical image inpainting problem, where we need to reconstitute or retouch the missing or damaged regions of an image to make it more legible and to restore its unity [29]. We emphasize that the link prediction problem addressed here is fundamentally different from the image inpainting problem. For image inpainting, we generally know the locations of the damaged regions of an image. While for link prediction, we don't know which links are missing in a network. In fact, teasing them out is exactly the task of link prediction.

At the first glance, our DGM-based link prediction method seems to heavily rely on the existing patterns in the adjacency matrix of the original network. After all, we are treating a network as an image. But do we have to sophisticatedly label the nodes in the network so that the resulting adjacency matrix (or the binary image) displays certain pattern? To address this concern, we perform the following numerical experiment. We start from a network with an appropriate node labeling such that the adjacency matrix looks exactly as the binary image of letter **E** without any missing pixels. Then we relabel η fraction of the nodes in the network so that the binary image associated with its adjacency matrix looks much more random than the letter **E**. Note that the network structure is fixed, and we just label the nodes differently so that the resulting adjacency matrices (or binary images) look quite different. We then compare the performance of our method at different η values, as well as the performance of two classical link prediction methods for directed networks that do not depend on the node labeling at all. We find that for this small directed network the performance of our method degrades only slightly even after we relabel 25% nodes (Fig.2A). When we relabel more nodes, the performance is actually quite stable. Even if we relabel all the nodes, the AUC of our method is still about 0.9, which is higher than that of other link prediction methods for directed networks, such as the preferential attachment (PA) [20] based method (with AUC~0.85) and the low-rank matrix completion (LRMC) [30] method (with AUC~0.7).

The results presented in Fig.2A indicate that the performance of our method does not heavily rely on the node labeling or the existence of structural features in the network. However, to reach the optimal performance of link prediction, one should still leverage any existing structural features in the network and label the nodes accordingly. This can be achieved by extracting community structure in the network [31–35], for example, using the classical Louvain method [36]. To test this simple idea, we consider the limiting case --- random graphs generated from the classical Erdős–Rényi (ER) model, where any two of N nodes are randomly connected with probability p [37]. By definition, in the large N limit, ER random graphs do not display any structural features and hence all link prediction methods are doomed to fail. For small N, a computer-generated ER random graph might display certain structural features, and link prediction should still be very difficult, if not impossible. We apply our method as well as various traditional link prediction method to ER random graphs ($N = 48$) at different connection probability p and with random node labeling. We find that our method has AUC~0.5, i.e., it behaves like random guess. In fact, no link prediction method performs significantly better than random guess, and some traditional methods even perform worse than random guess, especially for networks with lower connection probability (Fig.2B). However, applying the Louvain method first will capture some patterns in the adjacency matrices, which will significantly improve the AUC of our method (Fig.2B; paired-sample t-test). Real-world complex networks certainly display more prominent structural features than ER random graphs. Our numerical experiments suggest that those features should be exploited for link prediction.

To demonstrate the advantage of our method in real-world applications, we systematically compare the performance of our method with that of several classical methods in link prediction for a wide range of real networks, from social, economic, technological to biological networks. For undirected networks (Fig.3A), we find that generally global similarity indices (e.g., Katz, ACT) and stochastic block model (SBM) based link prediction methods perform better than local similarity indices (e.g., CN, PA, RA) based methods, because more topological information has been taken into account. But the performances of those methods vary a lot over different network domains. Some of them actually perform even worse than random guess, especially when the training set is small (corresponding to large f). By contrast, our DGM-based method displays very robust and high performance for various undirected networks. For directed networks (Fig.3B), most of the classical methods are actually not applicable. We compare the performance of our method with those of PA and LRMC. Again, we find that our method displays robust and high performance for various directed networks.

We emphasize that, since our DGM-based link prediction essentially treats a network as an image, it can be easily parallelized by splitting a large network (image) into different small subnetworks (subimages) and then performing link prediction for each subnetwork (subimage) in parallel. We find that this actually does not decrease the overall AUC, compared with the result of treating the large network as a whole. Furthermore, we can focus on any specific subnetwork of interest and just predict the missing links in that subnetwork. To test this idea, we perform link prediction for 200 subnetworks of size 60 randomly selected from two large directed networks: Facebook wall posts and Google+. We find our method shows much higher AUC than other methods (Fig.4). This result suggests that our method holds great promise in link prediction for large real-world networks.

In summary, our DGM-based link prediction shows superior performance against classical link prediction methods for various types of networks, be they of technological, biological, or social in nature. Since our method treats the adjacency matrix of a network as an image, it can be naturally extended to solve the link prediction problem for bipartite graphs, multi-layer networks and multiplex networks, where the adjacency matrices have certain inherent structure. With small modification, it can also be used to perform link prediction in weighted graphs. In principle, any DGM can be utilized in our method. But we find that, for the link prediction purpose, GANs perform much better than other DGMs, e.g., variational autoencoder [38] and pixelRNN [39]. There are several hyperparameters in training the GANs. In this work, we use the same set of hyperparameters for all the networks to show a conservative AUC estimation of our method. The performance of our method can certainly be further improved by carefully tuning those hyperparameters for a specific network of interest. Moreover, we anticipate that exploiting graphics process unit parallelism to train the GANs [40] will certainly speed up our link prediction method.

References

[1] R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).
[2] M. E. Newman, SIAM Rev. **45**, 167 (2003).
[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang, Phys. Rep. **424**, 175 (2006).
[4] C. Von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork, Nature **417**, 399 (2002).
[5] J.-D. J. Han, D. Dupuy, N. Bertin, M. E. Cusick, and M. Vidal, Nat. Biotechnol. **23**, 839 (2005).
[6] N. Sahni, S. Yi, M. Taipale, J. I. F. Bass, J. Coulombe-Huntington, F. Yang, J. Peng, J. Weile, G. I. Karras, and Y. Wang, Cell **161**, 647 (2015).
[7] A. Clauset, C. Moore, and M. E. J. Newman, Nature **453**, 98 (2008).
[8] C. De Bacco, E. A. Power, D. B. Larremore, and C. Moore, Phys. Rev. E **95**, (2017).
[9] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, and H. E. Stanley, Proc. Natl. Acad. Sci. **112**, 2325 (2015).
[10] L. Duan, S. Ma, C. Aggarwal, T. Ma, and J. Huai, IEEE Trans. Knowl. Data Eng. **29**, 2402 (2017).
[11] L. Lü and T. Zhou, Phys. Stat. Mech. Its Appl. **390**, 1150 (2011).
[12] V. Martínez, F. Berzal, and J.-C. Cubero, ACM Comput. Surv. CSUR **49**, 69 (2016).
[13] M. Campillos, M. Kuhn, A.-C. Gavin, L. J. Jensen, and P. Bork, Science **321**, 263 (2008).
[14] X. Chen, M.-X. Liu, and G.-Y. Yan, Mol. Biosyst. **8**, 1970 (2012).
[15] Y. Luo, X. Zhao, J. Zhou, J. Yang, Y. Zhang, W. Kuang, J. Peng, L. Chen, and J. Zeng, Nat. Commun. **8**, (2017).
[16] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, Phys. Rep. **519**, 1 (2012).
[17] G. Linden, B. Smith, and J. York, IEEE Internet Comput. **7**, 76 (2003).
[18] N. Blagus, L. Šubelj, and M. Bajec, Phys. Stat. Mech. Its Appl. **391**, 2794 (2012).
[19] G. Berlusconi, F. Calderoni, N. Parolini, M. Verani, and C. Piccardi, PloS One **11**, e0154244 (2016).
[20] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).
[21] L. Katz, Psychometrika **18**, 39 (1953).
[22] T. Zhou, L. Lü, and Y.-C. Zhang, Eur. Phys. J. B **71**, 623 (2009).
[23] R. Guimerà and M. Sales-Pardo, Proc. Natl. Acad. Sci. **106**, 22073 (2009).
[24] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, in *IJCAI* (1999), pp. 1300–1309.
[25] D. Heckerman, C. Meek, and D. Koller, Introd. Stat. Relational Learn. 201 (2007).
[26] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu, in *Adv. Neural Inf. Process. Syst.* (2007), pp. 1553–1560.
[27] A. Radford, L. Metz, and S. Chintala, ArXiv Prepr. ArXiv151106434 (2015).
[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, in *Adv. Neural Inf. Process. Syst.* (2014), pp. 2672–2680.
[29] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Tech.* (ACM Press/Addison-Wesley Publishing Co., 2000), pp. 417–424.

[30] R. Pech, D. Hao, L. Pan, H. Cheng, and T. Zhou, EPL Europhys. Lett. **117**, 38002 (2017).
[31] M. Girvan and M. E. Newman, Proc. Natl. Acad. Sci. **99**, 7821 (2002).
[32] M. E. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[33] A. Clauset, M. E. Newman, and C. Moore, Phys. Rev. E **70**, 066111 (2004).
[34] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proc. Natl. Acad. Sci. U. S. A. **101**, 2658 (2004).
[35] R. Guimera, M. Sales-Pardo, and L. A. N. Amaral, Phys. Rev. E **70**, 025101 (2004).
[36] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, J. Stat. Mech. Theory Exp. **2008**, P10008 (2008).
[37] P. ERDdS and A. R&WI, Publ Math Debr. **6**, 290 (1959).
[38] K. Sohn, H. Lee, and X. Yan, in *Adv. Neural Inf. Process. Syst.* (2015), pp. 3483–3491.
[39] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, ArXiv160106759 Cs (2016).
[40] D. J. Im, H. Ma, C. D. Kim, and G. Taylor, ArXiv Prepr. ArXiv161204021 (2016).
[41] A. Arenas, A. Fernández, and S. Gómez, New J. Phys. **10**, 053039 (2008).

Acknowledgments: This work is supported in part by the China Scholarship Council. We thank Marco Tulio Angulo and Tina Eliassi-Rad for valuable discussions.

Contributions: Y.-Y.L. conceived and designed the project. X.-W.W. and Y.C. did the analytical and numerical calculations. X.-W.W. analyzed all the real networks. All authors analyzed the results. Y.-Y.L. and X.-W.W. wrote the manuscript. Y.C. edited the manuscript.

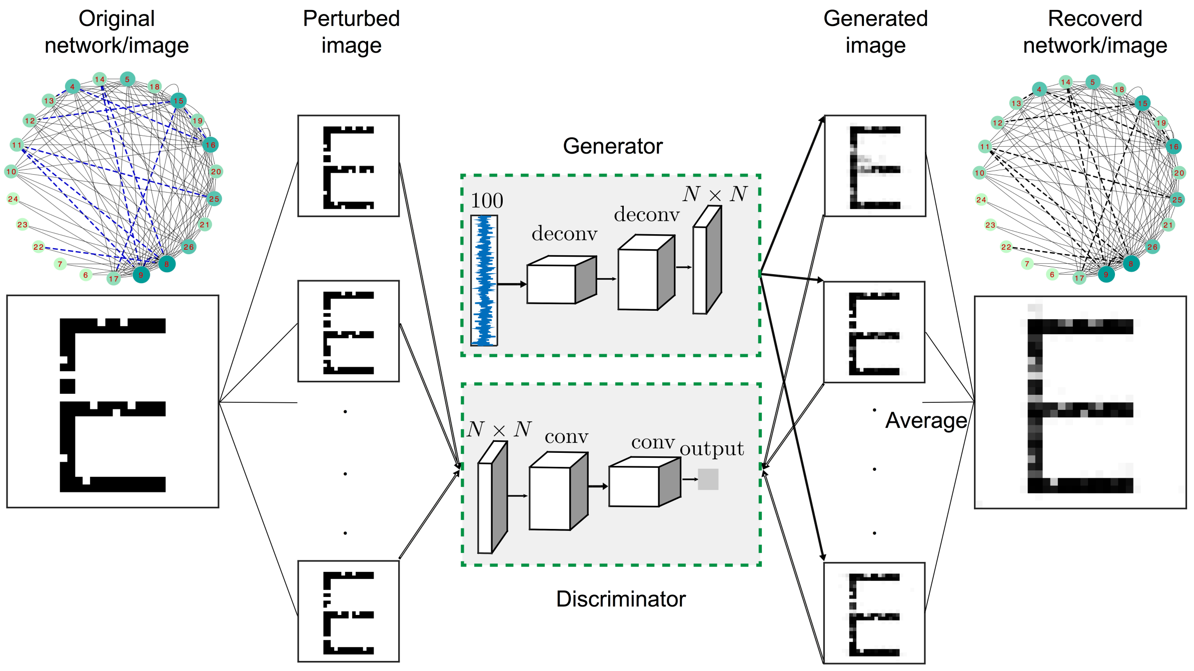

Fig. 1. Schematic demonstration of our link prediction method on a directed network. The adjacency matrix of this directed network (with 28 nodes and 118 links) looks like the binary image of letter **E** with 12 missing pixels. Note that 5 isolated nodes are not shown in the network presentation. We perturb the original network (image) by removing 5 links at random in M different ways to obtain a pool of perturbed networks (images) I_i ($i = 1, \ldots, M$) ($M = 5000$ for this example). This input dataset will be fed into the generative adversarial networks (GANs) that consist of two deep artificial neural networks: generator and discriminator. The generator takes the noise drawn from a uniform distribution as input and produces fake images. The discriminator is a binary classifier that tells whether a given image is a real one from the input dataset or a fake one produced by the generator. Over the course of training iterations, the generator can produce convincing fake images P from the feedback offered by the discriminator. The pixel value P_{ij} in the fake grayscale image P can be used to calculate the existent probability of a link between a node pair: $\alpha_{ij} = 1 - P_{ij}$. The final existent probability is calculated by averaging α_{ij} over S ($S = 500$) generated fake networks.

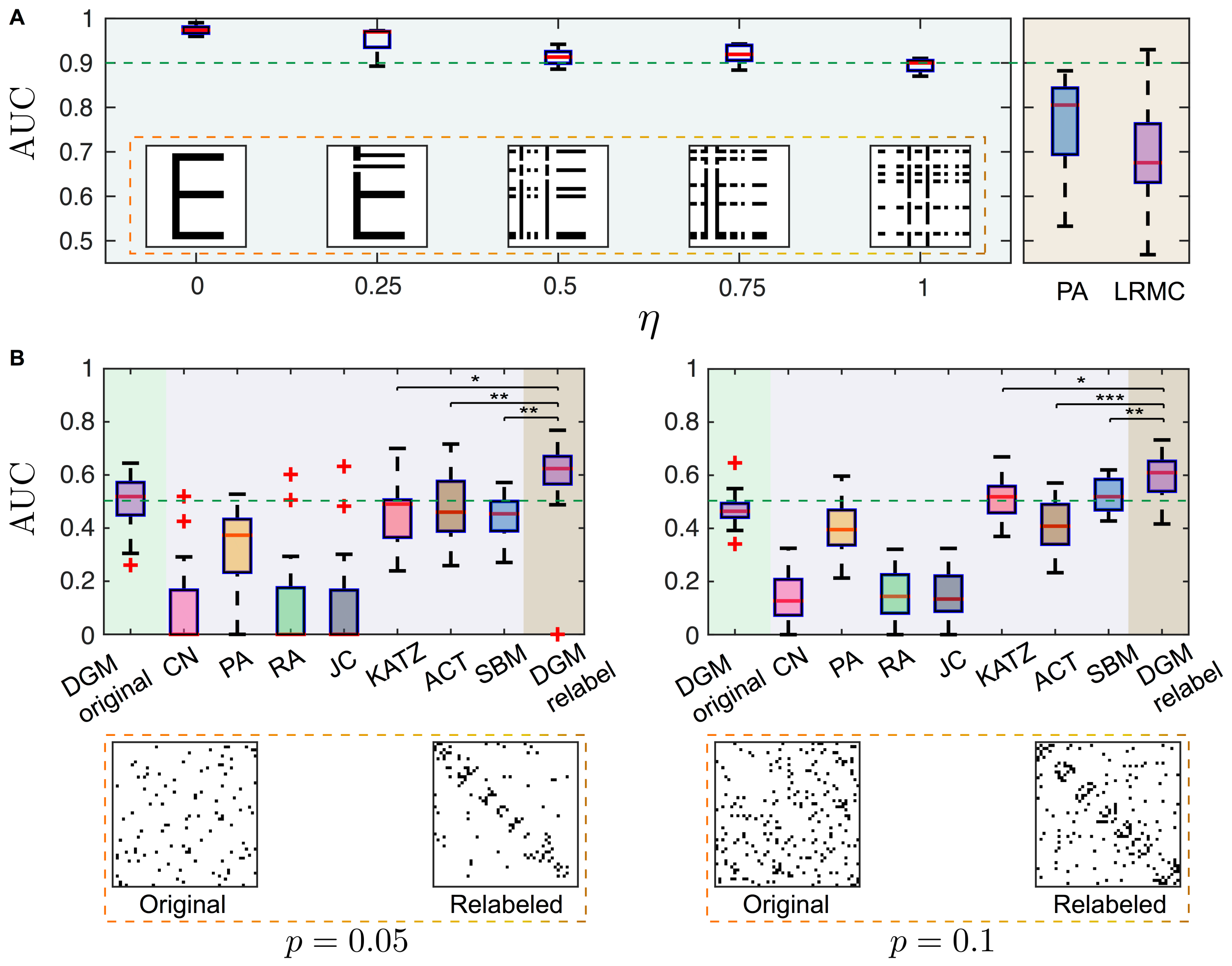

Fig. 2. Stability of the DGM-based link prediction method. A: A randomly selected fraction of η nodes are relabeled in a directed network whose original adjacency matrix looks exactly as the binary image of letter **E**. Then we randomly divide the links into two parts: a fraction of 10% links chosen as the probe set and the remaining 90% fraction of links as the training set. We perform link prediction using three different methods: DGM, PA, LRMC. In this example, we choose $M = 1000$ for our DGM-based method. Even after we relabel all the nodes so that the adjacency matrix doesn't display prominent features, the median AUC of our DGM-based method is still around 0.9, while it is 0.85 for the PA method and 0.7 for the LRMC method. **Inset:** The adjacent matrices corresponding to different relabeling fractions, where black pixels represent existing links. **B:** AUC of DGM-based and other traditional models in the link prediction of Erdős–Rényi (ER) random graphs with different connection probability p. The adjacent matrices (before and after node relabeling) at different connection probabilities are also shown. CN: common neighbors; PA: preferential attachment; RA: resource allocation; JC: Jaccard index; KATZ: Katz index; ACT: average commute time; SBM: stochastic block model (LRMC: the low rank matrix completion method has very low AUC, so it is not shown here). Asterisks in panel B shows whether the AUC of our DGM-based link prediction method is significantly higher than that of the other three traditional algorithms (paired-sample t-test). Significance levels: p-value<0.05(*), <0.01(**), <0.001(***).

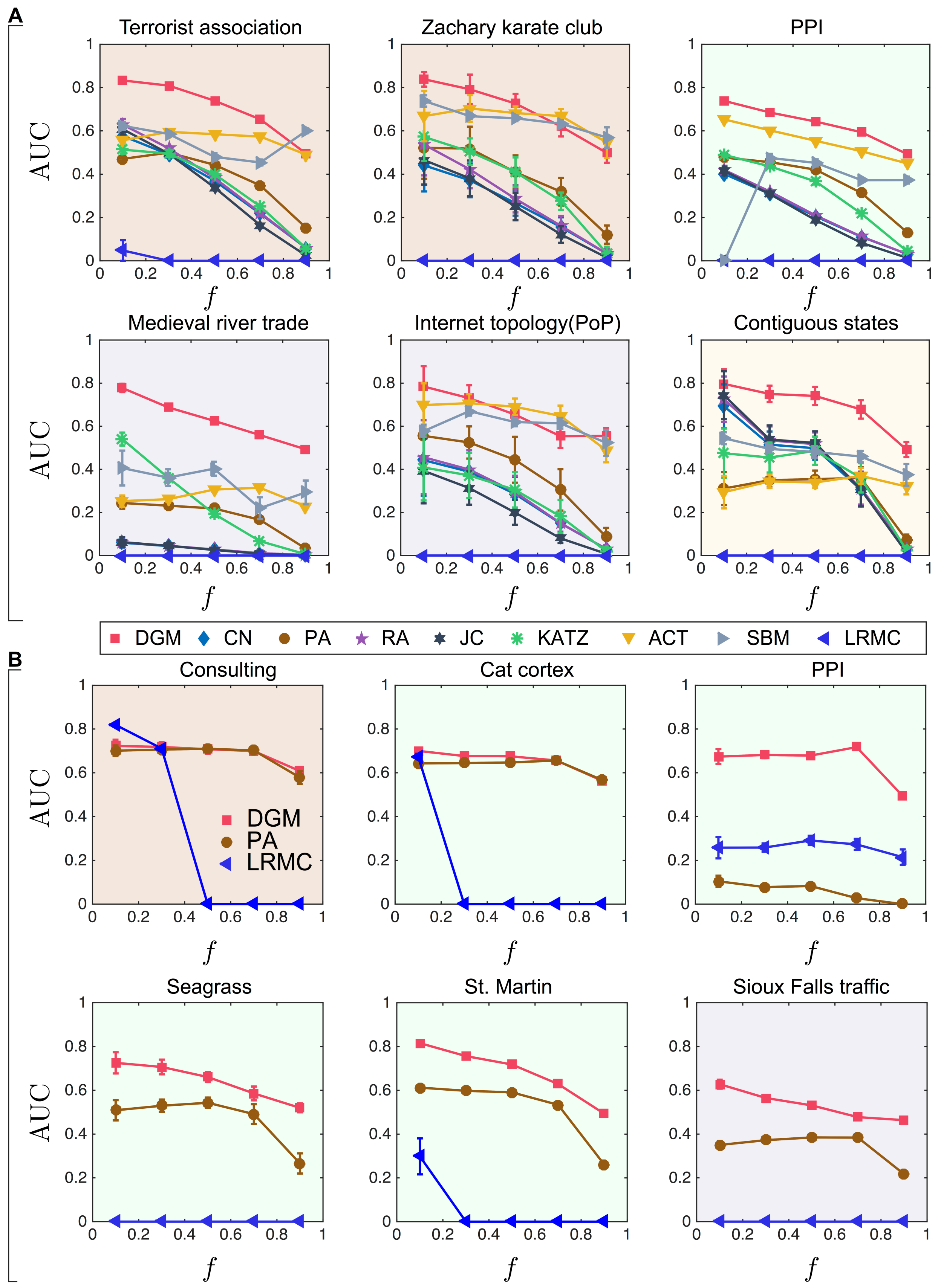
A
Terrorist association
Zachary karate club
PPI
Medieval river trade
Internet topology(PoP)
Contiguous states
AUC
f
DGM
CN
PA
RA
JC
KATZ
ACT
SBM
LRMC
B
Consulting
Cat cortex
PPI
Seagrass
St. Martin
Sioux Falls traffic

Fig. 3. The DGM-based link prediction displays very robust and high performance for both undirected and directed real-world networks. DGM: deep generative model based link prediction; CN: common neighbors; PA: preferential attachment; RA: resource allocation; JC: Jaccard index; KATZ: Katz index; ACT: average commute time; SBM: stochastic block model; LR: low rank matrix completion. **A**: Undirected networks. Top: Terrorist association network, Zachary karate club, Protein-protein interaction (PPI) network (a subnetwork of protein interactions in *S. cerevisiae*). Bottom: Medieval river trade network in Russia, Internet topology (at the PoP level), Contiguous states in USA. **B:** Directed networks. Top: Consulting (a social network of a consulting company), cat cortex (the connection network of cat cortical areas), PPI (a subnetwork chosen from the network of protein interactions in Humans (*Homo sapiens*). Bottom: Seagrass food web, St. Martin food web, Sioux Falls traffic network. AUC of our DGM-based method is the average AUC over the last 20 epochs of the total 150 epochs for all of networks. Here an epoch is one full training cycle on the training set. For all the undirected real networks, we apply the Louvain method first to label the nodes appropriately. Directed networks are labeled by the method proposed in [41].

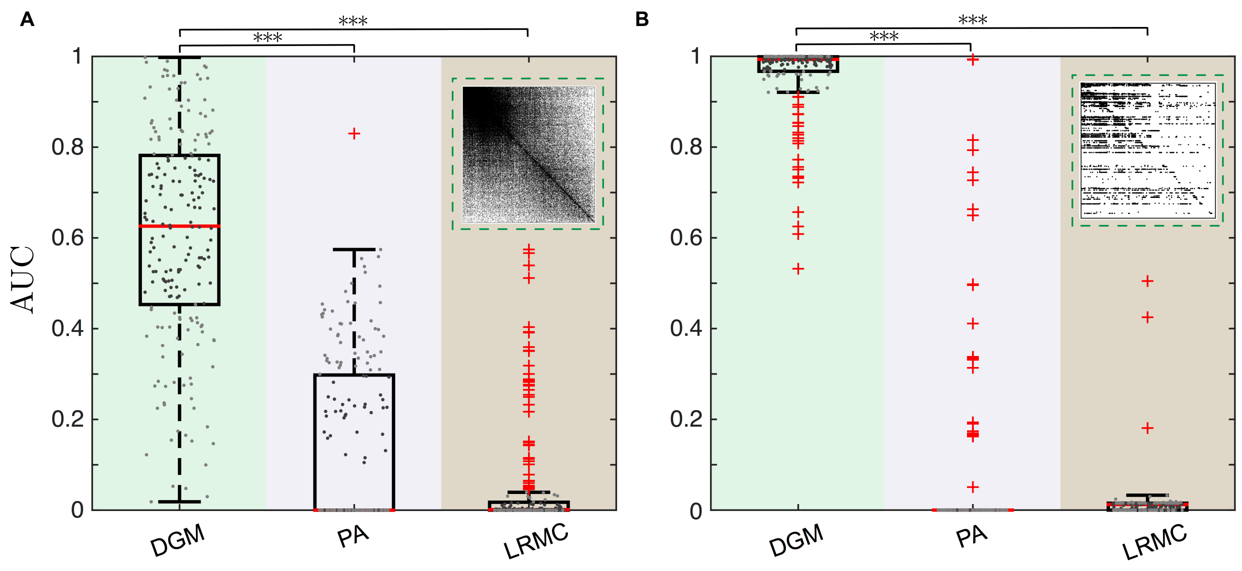

Fig. 4. The DGM-based link prediction method can infer missing links of arbitrarily selected subnetworks within a large network with higher accuracy than other methods. We perform link prediction for 200 randomly selected subnetworks (of size 60) chosen from two large directed networks: **A**, Facebook wall posts; and **B**, Google+, respectively. We randomly divide the links of the relabeled networks into two parts: a fraction of 10% links are chosen as probe set and the remaining 90% fraction of links as training set (here, each subnetwork contains 15 links at least). PA: preferential attachment index, LRMC: low rank matrix completion. **Insets**: adjacency matrices of the networks. Asterisks at the top of each panel shows whether the AUC of our DGM-based link prediction model is significantly higher than that of the other two traditional algorithms (paired-sample *t*-test). Significance levels: p-value<0.05(*), <0.01(**), <0.001(***).