From Closing Triangles to Higher-Order Motif Closures for Better Unsupervised Online Link Prediction

Ryan A. Rossi
Adobe Research
rrossi@adobe.com

Anup Rao
Adobe Research
anuprao@adobe.com

Sungchul Kim
Adobe Research
sukim@adobe.com

Eunyee Koh
Adobe Research
eunyee@adobe.com

Nesreen K. Ahmed
Intel Labs
nesreen.k.ahmed@intel.com

Gang Wu
Adobe Research
gawu@adobe.com

ABSTRACT

This paper introduces higher-order link prediction methods based on the notion of closing higher-order network motifs. The methods are fast and efficient for *real-time* ranking and link prediction-based applications such as online visitor stitching, web search, and online recommendation. In such applications, real-time performance is critical. The proposed methods do not require any explicit training data, nor do they derive an embedding from the graph data, or perform any explicit learning. Most existing unsupervised methods with the above desired properties are all based on closing triangles (common neighbors, Jaccard similarity, and the ilk). In this work, we develop unsupervised techniques based on the notion of closing higher-order motifs that generalize beyond closing simple triangles. Through extensive experiments, we find that these higher-order motif closures often outperform triangle-based methods, which are commonly used in practice. This result implies that one should consider other motif closures beyond simple triangles. We also find that the "best" motif closure depends highly on the underlying network and its structural properties. Furthermore, all methods described in this work are fast for link prediction-based applications requiring real-time performance. The experimental results indicate the importance of closing higher-order motifs for unsupervised link prediction. Finally, these new higher-order motif closures can serve as a basis for studying and developing better unsupervised real-time link prediction and ranking methods.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Mathematics of computing** → **Graph algorithms**; *Combinatorics*; *Graph theory*; • **Information systems** → **Data mining**; • **Theory of computation** → **Graph algorithms analysis**; **Streaming, sublinear and near linear time algorithms**;

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481920

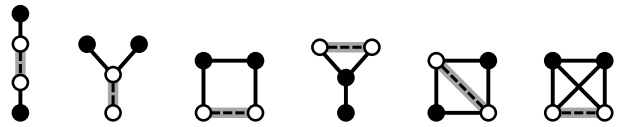

Figure 1: Higher-Order Motif Closures. The unshaded/white nodes are node i and j. Given a node pair $(i, j) \notin E$ (unshaded/white nodes) and any motif/induced subgraph H, the "edge" between i and j (dotted gray line) is said to close an instance F of H if the edge (i, j) were to actually exist in G.

KEYWORDS

Motif closure; higher-order motif closure; unsupervised link prediction; network motifs; graphlets; real-time algorithms; higher-order link prediction; online algorithms

ACM Reference Format:
Ryan A. Rossi, Anup Rao, Sungchul Kim, Eunyee Koh, Nesreen K. Ahmed, and Gang Wu. 2021. From Closing Triangles to Higher-Order Motif Closures for Better Unsupervised Online Link Prediction. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3459637.3481920

1 INTRODUCTION

Link prediction generally refers to predicting the existence of edges (node pairs) in G such that the predicted edges (node pairs) are not in the original edge set E of G. The goal of this task may be to predict future links at time $t + 1$ or to simply predict links that were not observed (*e.g.*, to improve the quality of downstream tasks) [33]. Notice that nearly all link prediction methods first compute a weight $W_{ij} = f(i, j)$ between node i and j and then use W_{ij} to decide whether to predict a link (i, j) or not. We denote the task of estimating a weight $W_{ij} = f(i, j)$ between node i and j as *link weighting* or *link strength estimation*. The weights are then used to derive a ranking of potential links. The potential links may refer to item j that a user i is likely to purchase, or songs that a user is likely to prefer, and so on. In this work, we focus on fast and efficient methods for computing link weights based on closing higher-order network motifs. Such weights based on higher-order motif closures can then be used for ranking-based applications (such as recommender systems and the ilk).

Ranking (and link prediction [26]) is a key component of many real-world applications such as online real-time visitor stitching [24], web search [13], online advertising, and recommendation [28]. In these applications, real-time performance is critical, *e.g.*, in web search users expect an answer to their query in the order of a few hundred milliseconds [8, 13]. This makes it impossible to learn a complex ranking function. Instead, there are usually two components to such a system. In the first component, a *fast online approach* is used to identify the top-k most relevant results in real-time (where k is typically small), which are then displayed to the user. In the second component, a more accurate but computationally expensive model is trained to improve the initial ranking. The ranking learned from the model can be used directly or combined with simpler approaches to obtain a final re-ranking of the web pages (or items). In this work, we primarily focus on the first component.

Triangle closure (common neighbors) and variants based on it such as Jaccard similarity and Adamic/Adar[1] are known to be strong baselines that are hard to beat in practice [40]. These baselines are all fundamentally based on the notion of "closing triangles" [1, 3, 33]. They are both simple and fast for ranking in an online real-time fashion. In this work, we investigate whether other motif closures are as useful as the triangle closure and its variants (*e.g.*, Jaccard similarity, Adamic/Adar, among others [33]) that have been used over the last decade for (un)supervised ranking and link prediction. More specifically, we investigate the 4-node motif closures shown in Figure 1. We find that these new motif closures are often more predictive than their triangle-based counterparts. This result implies that one should consider other motif closures beyond simple triangles. We also find that the "best" motif closure for ranking (and prediction) depends highly on the underlying network and its structural properties.

While most existing work focuses on learning a ranking function [11, 14, 42], we instead focus on direct principled approaches that are: (1) efficient (sublinear in the number of nodes), (2) can be directly computed in real-time, (3) easily parallelizable, and (4) naturally amenable for online real-time ranking in the streaming setting. This work introduces the general notion of closing higher-order motifs and based on this notion we develop direct ranking techniques that are efficient for *real-time online* ranking and prediction. Compared to similar techniques that can be used for this setting such as Common Neighbors and methods based on it (*e.g.*, Jaccard similarity), the proposed techniques are fundamentally more powerful as they naturally generalize over these existing techniques that are all based on closing triangles (a lower-order motif). The proposed notion of higher-order motif closure can serve as a basis for studying and developing better ranking (and prediction) methods based on the higher-order motif closures.

2 PRELIMINARIES

Let $\mathbf{r}$ denote a vector of ranks from an arbitrary link estimation method. Hence, r_k denotes the link at rank k. The label of link i is denoted by $\xi(i)$ (or $\xi(x_i)$) where $\xi(i) = 1$ if link i is relevant and otherwise $\xi(i) = 0$ if non-relevant. Let $Y = \{\, i : \xi(i) = 1 \,\}$ denote the set of relevant links and $\bar{Y} = \{\, i : \xi(i) = 0 \,\}$ denotes the set of irrelevant links. The number of all relevant/non-relevant links is denoted by $m' = |Y| + |\bar{Y}|$.

Definition 1 (Precision at K (P@K)). *Given an integer $1 \leq k \leq m'$ and let Y_k denote the set of* relevant *links in the top-k, then Precision at K (P@K) is defined as:*

$$\mathbb{P}_k = \frac{|Y_k|}{k} \tag{1}$$

Mean Average Precision (MAP) is defined as:

$$\mathbb{E}(\mathbf{x}) = \frac{1}{|Y|} \sum_{k=1}^{m'} \mathbb{P}_k \cdot \mathbb{I}\Big[\, r_k \in Y \,\Big] \tag{2}$$

where r_k denotes the link at rank k and for any predicate p the indicator function $\mathbb{I}[\, p \,] = 1$ iff p holds and 0 otherwise. Intuitively, MAP is the average precision over all recall levels.

Given a set of links (or items for a specific user i) ordered from most likely to least, coverage measures the normalized max position in the ranking such that all proper relevant links are recovered:

$$\mathbb{E}(\mathbf{x}) = \frac{1}{|\bar{Y}|} \left[\max_{k \in Y} \pi(\mathbf{x}, k) \right] - |Y| \tag{3}$$

where Y is the set of relevant links, $\bar{Y}$ is the set of irrelevant links, and $\pi(\mathbf{x}, k)$ is the rank of link $k \in Y$ when the estimated link weight vector $\mathbf{x}$ of an arbitrary method is sorted in descending order. The normalized coverage indicates the fraction of non-relevant links that must be looked at before obtaining all relevant links. Perfect performance is achieved when $\mathbb{E}(\mathbf{x}) = 0$. This implies that all relevant links are ordered first followed by the non-relevant links. The above criteria is normalized for comparison across different data sets.

Discounted Cumulative Gain (DCG) [28] is a popular measure for evaluating the quality of a ranking. It is defined as follows:

$$DCG_k(\mathbf{y}) = \sum_{i=1}^{k} \frac{2^{y_i} - 1}{\log_2(i+2)} \tag{4}$$

where i is the rank and $y_i \in \{0, 1\}$ is the label (relevant/irrelevant) of the link in position i in the ranking. This metric emphasizes the quality of the ranking at the top of the list since $1/\log 2(i+2)$ decreases quickly and then asymptotes to a constant as i increases.

3 HIGHER-ORDER MOTIF CLOSURES

We first introduce the notion of a higher-order motif closure that lies at the heart of this work.

Definition 2 (Motif Closure). *A node pair (i, j) is said to* close *a motif H iff adding an edge (i, j) to E* closes *an instance $F \in I_{G'}(H)$ of motif H where $G' = (V, E \cup \{(i, j)\})$ and $I_{G'}(H)$ is the set of unique instances of motif H in G'.*

Figure 1 provides a few examples of higher-order motif closures. The edge (i, j) shown as a dotted line in Figure 1 closes each motif. For instance, the edge between node i and j in the rightmost motif in Figure 1 closes a 4-clique. We now formally introduce the frequency of higher-order motif closures for a node pair (i, j) as follows:

Definition 3 (Higher-Order Motif Closure Frequency). *Let $G' = (V, E')$ where $E' = E \cup \{(i, j)\}$ and let $I_{G'}(H)$ be the set of*

[1] Nearly all local techniques (9 out of 10) discussed in [33] are based on triangle closure.

Table 1: Mean average precision (MAP) results for ranking (and prediction) methods based on closing higher-order motifs.

	bn-mouse	*bio-DM-LC*	*bio-CE-HT*	*bio-DM-HT*	*ia-reality*	*web-polblogs*	*biogrid-worm*	*biogrid-plant*	*biogrid-yeast*	*email-dnc-corec.*	*soc-advogato*	*econ-wm1*	*bn-macaque-rhe.*	*road-minnesota*	*soc-fb-messages*	*email-EU*	*email-univ*
4-path	0.829	0.687	0.607	0.594	0.649	0.778	0.865	0.729	0.893	0.873	0.914	0.788	0.942	0.326	0.844	0.854	0.707
4-star	0.880	0.787	0.595	0.696	**0.922**	0.814	0.895	0.861	0.840	0.813	0.889	0.688	**0.961**	0.388	0.807	**0.972**	0.695
4-cycle	**0.881**	**0.958**	**0.651**	**0.926**	0.827	0.885	**0.908**	**0.935**	0.927	0.957	0.930	0.900	0.773	**0.950**	0.870	0.902	0.847
4-tailed-triangle	0.804	0.612	0.570	0.752	0.773	0.663	0.773	0.681	0.689	0.779	0.600	0.496	0.530	0.834	0.722	0.937	0.582
4-chordal-cycle	0.801	0.837	0.598	0.842	0.312	**0.966**	0.840	0.854	**0.977**	0.996	**0.986**	0.947	0.750	0.939	0.935	0.782	0.969
4-clique	0.804	0.838	0.595	0.843	0.293	0.963	0.842	0.847	0.972	**0.997**	**0.986**	**0.965**	0.759	0.939	**0.960**	0.798	**0.982**
CN	0.705	0.872	0.613	0.839	0.422	0.814	0.833	0.897	0.839	0.960	0.949	0.852	0.342	0.945	0.790	0.890	0.941
Jaccard Sim.	0.705	0.873	0.618	0.841	0.537	0.933	0.853	0.918	0.955	**0.997**	0.973	0.918	0.764	0.944	0.841	0.933	0.949
Adamic/Adar	0.705	0.883	0.621	0.842	0.549	0.940	0.856	0.920	0.959	**0.997**	0.976	0.919	0.777	0.945	0.848	0.935	0.953

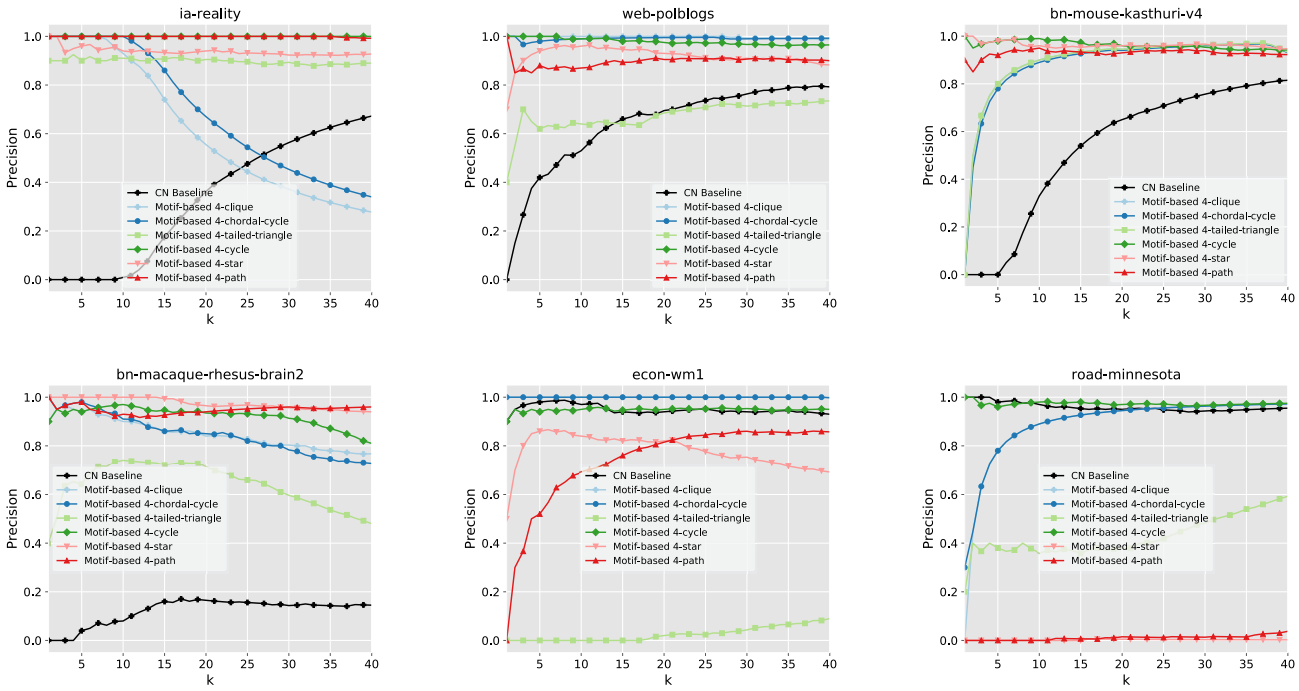

Figure 2: Precision at $k = 1, \ldots, 40$ for different motif closure rankings.

unique instances of motif H in G'. Then the frequency of closing a higher-order motif H between node i and j is:

$$W_{ij} = \sum_{F \in I_{G'}(H)} \mathbb{I}\big(\{i,j\} \in E'(F)\big) \tag{5}$$

where W_{ij} is equal to the number of unique instances of H that contain nodes $\{i,j\} \subset V(G')$ as an edge.

We now discuss an approach for computing the motif closure weight W_{ij} representing the frequency of closing motif H between node i and j. The approach has two simple steps. First, given an arbitrary node pair (i,j), a motif H of interest, and the current graph $G = (V,E)$, we add the node pair (i,j) as an edge by setting $E' \leftarrow E \cup \{(i,j)\}$ and $G' = (V,E')$.[2] As an aside, this can be performed *implicitly* without any additional work. After adding (i,j) to the edge set, we count the occurrences of motif (induced subgraph/graphlet) H between node i and j in G'. For computing the number of instances of motif H that contain nodes i and j in G', we can always use the fastest known algorithm [3]. Nevertheless,

[2]Note that if edges are arriving continuously over time in a streaming fashion, then we may also encounter a node i (or j) such that $i \notin V$. In this case, we also set $V' \leftarrow V \cup \{i\}$ and $G' = (V', E')$.

one can always modify the algorithm to count all motif instances that contain node i and j such that the method *implicitly* treats the pair of nodes (i, j) as an edge to determine the number of instances of H that would be closed if (i, j) was to be added to G. Given a set $\mathcal{Y} = \{y_1, y_2, \ldots, y_j, \ldots\}$ of nodes (items, ads, songs, friends) to be ranked, we can use the above routine to obtain $W_{ij} = f(x_i, y_j)$, $\forall j = 1, \ldots, |\mathcal{Y}|$.

Extending Other Measures using Motif Closure. Given two nodes i and j, Common Neighbor-based methods are those that use the quantity $|\Gamma_i \cap \Gamma_j|$ where Γ_i and Γ_j are the set of neighbors for node i and j, respectively. Common neighbors is simply $W_{ij} = |\Gamma_i \cap \Gamma_j|$ where W_{ij} represents the number of *potential triangles* that *would be closed* if there were an edge between i and j. The notion of "closing" triangles lies at the heart of many other existing methods that are based on $|\Gamma_i \cap \Gamma_j|$ such as Jaccard similarity, Adamic/Adar (AA), among others. All of these methods can be viewed as extensions of Common Neighbors with some form of normalization, *e.g.*, Jaccard similarity is $W_{ij} = |\Gamma_i \cap \Gamma_j| / |\Gamma_i \cup \Gamma_j|$. Extending the proposed higher-order motif-based link ranking and prediction techniques is left for future work. This includes extending the notion of "closing" higher-order network motifs for other measures such as Jaccard similarity, Adamic/Adar, among any others where the notion of *closing triangles* can be replaced with the notion of *closing a higher-order motif* introduced in this work.

Parallelization. The motif closures lend themselves to an efficient parallel implementation. Observe that each motif closure is defined precisely for a pair of nodes and thus each weight representing the strength of that link (node pair) can be estimated independently using only the local structure surrounding the nodes. Thus, the edge weights can be computed efficiently in parallel. The communication costs are also minimal. Moreover, all such 4-node motif closures require only local information of their surrounding neighborhood. Therefore, the motif closures can also be computed in the streaming or semi-streaming setting where the amount of memory is limited and thus storing the entire graph is impossible. In such a setting, it may be useful to implement the parallelization at a finer-granularity, which is also straightforward for such motif closures.

4 EXPERIMENTS

The experiments are designed to evaluate the effectiveness of different motif closures that go beyond closing triangles. In particular, the experiments investigate the following key questions:

- **Q1** Do other motif closures perform better than triangle closure and its variants for some graphs?
- **Q2** Does the "best" motif closure depend highly on the underlying network and its structural properties or is there one motif closure that always outperforms the others?
- **Q3** Are the motif closures more robust to noise in the graph (*e.g.*, random link additions) compared to triangle closure methods?

To ensure the significance and generality of our findings (as much as possible), we evaluate the proposed methods using a wide variety of networks from different application domains. All data was obtained from NetworkRepository [31].

To investigate the above questions, we compare the higher-order motif closures against triangle closure (common neighbors) and its variants (Jaccard similarity, Adamic/Adar) since these are all based on closing triangles and have the same desired properties as the higher-order motif closure methods described in this paper. In particular, we compare against other online approaches that can be used in a streaming real-time fashion with similar runtime. In this work, we only investigate the most basic and fundamental higher-order motif closures. Using the new higher-order motif closures as a basis to develop more sophisticated measures for ranking and link prediction is left for future work. However, we did run a few experiments using an extended higher-order Jaccard similarity (one for each motif closure, giving 6 total for 4-node motifs) and higher-order Adamic/Adar ranking measures, again giving 6 new rankings total. Since each variant provides 6 additional rankings, the results were removed for brevity, but in some cases performed better than the most basic motif closures introduced in this paper. As such, the proposed notion of higher-order motif closures serve as fundamental building blocks for developing better higher-order ranking and prediction methods.

Unless otherwise mentioned, we hold-out 10% of the observed node pairs uniformly at random and randomly sample the same number of negative node pairs. We repeat this 10 times and average the results. We then use the methods to obtain a ranking of the node pairs in this set.[3] Recall the proposed techniques do not require learning a sophisticated model nor do they require training data. As such, the notion of motif closure proposed in this work can be used in a real-time streaming fashion and has many other advantages to more sophisticated model-based approaches. Mean Average Precision (MAP) results are provided in Table 1 whereas coverage is provided in Table 2.

Result 1. *Higher-order motif closures can outperform triangle closure (common neighbors) and other methods based on it.*

In nearly all cases, the higher-order motif closures achieve better precision and coverage than techniques based on closing lower-order triangles. This result has a number of important implications. First, it implies that one should also consider other motif closures that go beyond simple triangles. Furthermore, this finding also brings new opportunities for research on different and more useful variants based on these new motif closures, similar to how triangle closure has been used to derive many variations including Jaccard similarity, Adamic/Adar, RA, cosine similarity, Sorensen index, hub index, hub depressed index, among others discussed in [33]. Second, the "best" motif closure for a given task depends highly on the underlying network structure and domain-level processes that govern it. Third, existing supervised learning methods can benefit from these new motif closures by leveraging the full range of motif closures (going from least to most dense as shown in Figure 1).

Result 2. *There is no single higher-order motif closure that performs best for all graphs. The best motif depends highly on the structural characteristics of the graph and its domain (biological vs. social network) as shown in Table 1 and Table 2.*

[3] In recommender systems, the set of node pairs to be ranked is actually a smaller set of "relevant items" $\mathcal{Y}_i = \{y_1, \ldots, y_j, \ldots\} \subset \mathcal{Y}$ for a user i. Nevertheless, this can also be viewed as a ranking of node pairs where user i is fixed.

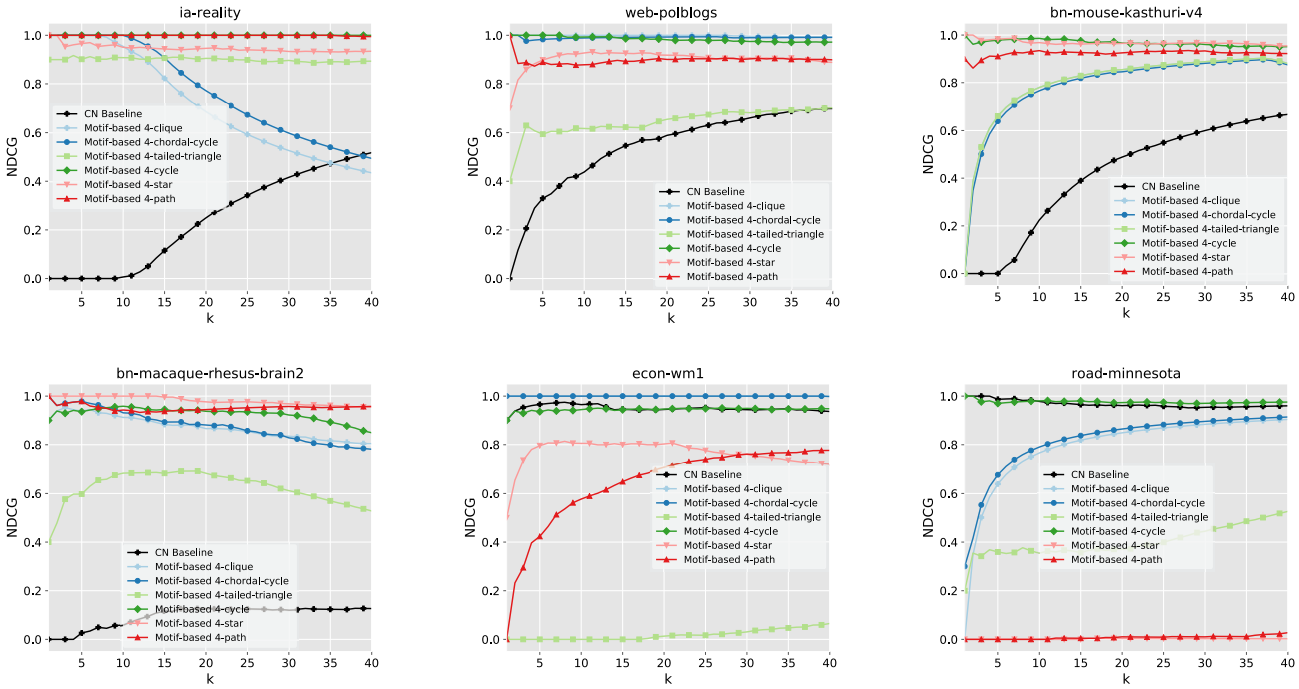

Figure 3: nDCG at $k = 1, \ldots, 40$ for different motif closure rankings.

Table 2: Coverage (↓) results for the ranking methods. Coverage measures the normalized max position in the ranking such that all positive node-pairs are recovered. Lower is better.

	bn-mouse	*bio-DM-LC*	*bio-CE-HT*	*bio-DM-HT*	*ia-reality*	*web-polblogs*	*biogrid-worm*	*biogrid-plant*	*biogrid-yeast*	*email-dnc-corec.*	*soc-advogato*	*econ-wm1*	*bn-macaque-rhe.*	*road-minnesota*	*soc-fb-messages*	*email-EU*	*email-univ*
4-path	0.606	0.964	0.822	0.962	0.537	0.958	0.828	0.980	0.963	0.911	0.936	0.976	1	**0.999**	0.970	0.579	0.991
4-star	0.637	0.950	**0.815**	0.944	**0.024**	0.911	0.972	0.963	0.945	0.998	0.985	0.915	**0.088**	**0.999**	0.953	**0.245**	0.986
4-cycle	**0.300**	**0.375**	0.922	**0.645**	0.457	0.942	**0.542**	**0.483**	0.869	0.801	0.952	0.942	0.995	1	0.971	0.366	0.897
4-tailed-triangle	1	0.910	1	1	0.451	0.967	0.654	0.756	0.986	0.893	0.997	0.964	1	1	0.981	0.404	0.992
4-chordal-cycle	1	0.913	1	0.913	0.936	**0.613**	1	1	**0.228**	**0.211**	**0.217**	0.796	0.965	1	1	0.620	**0.390**
4-clique	1	1	1	1	1	0.709	1	1	0.322	0.216	0.315	**0.521**	0.989	1	0.902	0.62	0.409
CN	1	0.732	1	0.826	0.652	0.705	0.752	0.739	0.788	0.224	0.782	0.894	0.972	1	**0.864**	0.461	0.766
Jaccard Sim.	1	0.732	1	0.826	0.658	0.707	0.752	0.739	0.824	0.225	0.782	0.902	0.960	1	0.885	0.462	0.762
Adamic/Adar	1	0.732	1	0.826	0.653	0.706	0.747	0.739	0.823	0.218	0.783	0.905	0.923	1	0.866	0.462	0.761

This implies that different motif closures perform better than others depending on the underlying graph characteristics and structural properties, and the best motif closure appears to be correlated with the underlying domain. This result is consistent with the no-free-lunch-theorem [41]. In particular, there are specific motif-closures that often perform best for specific data types/domains(e.g., bio, social,...), which is consistent with no-free-lunch-theorem [41]. Furthermore, there is almost always a motif-closure that outperforms the baselines, which is significant as well.

Result 3. *The best performing motif closure for a given network is consistent across different evaluation measures. The motif closure that achieves the best precision (Table 1) for a given network is typically the same motif that achieves the best coverage (Table 2).*

In Table 1-2, biological and brain networks achieve best performance using the ranking given by 4-cycle and 4-star closures. This also holds true for the interaction (ia-reality) and road network investigated. The 4-star and 4-cycle motif closures are more sparse compared to the 4-chordal-cycle (paw motif) and 4-clique motif

Table 3: Robustness results (MAP). See text for discussion.

	bn-mouse	*bio-DM-LC*	*bio-CE-HT*	*bio-DM-HT*	*ia-reality*	*web-polblogs*	*biogrid-worm*	*biogrid-plant*
4-path	0.764	0.698	0.495	0.582	0.774	0.812	**0.899**	0.847
4-star	**0.868**	0.781	**0.52**	0.645	**0.920**	0.81	0.894	0.836
4-cycle	0.763	**0.915**	0.482	**0.871**	0.665	0.855	0.863	**0.872**
4-tailed-triangle	0.684	0.700	0.444	0.755	0.725	0.708	0.765	0.658
4-chordal-cycle	0.781	0.816	0.440	0.811	0.237	0.942	0.812	0.776
4-clique	0.781	0.825	0.440	0.809	0.205	**0.944**	0.807	0.761
CN	0.686	0.800	0.458	0.801	0.283	0.806	0.791	0.819
Jaccard Sim.	0.693	0.809	0.461	0.803	0.357	0.910	0.808	0.828
Adamic/Adar	0.704	0.809	0.462	0.803	0.375	0.912	0.808	0.831

closure. In the web graph, economic, and social networks, both the 4-chordal-cycle (diamond motif closure) and 4-clique motif closure achieves significantly better performance than the other motif closures. Notice that both these motif closures are composed of two or more triangles and thus can be seen as a stronger triadic closure motif. The 4-path, 4-tailed-triangle, and triangle (CN) motif closures did not perform the best in any of the graphs investigated. That is, there were always a higher-order motif closure with better performance as shown in Table 1 and Table 2. In Figure 2, we also show the precision at $k = 1, \ldots, 40$ for closing different higher-order network motifs. In nearly all cases, the rankings given by the 4-node motif closures are better than the lower-order CN approach that is based on closing triangles. In Figure 3, we also provide the normalized Discounted Cumulative Gain (nDCG) [28] at $k = 1, \ldots, 40$ for the different motif closures. nDCG [28] is another standard ranking quality measure that emphasizes the quality of ranking at the top of the list. The results in Figure 3 are shown to be consistent with those in Figure 2.

Robustness of Ranking from Higher-Order Motif Closures. In addition, we investigate the robustness of the higher-order motif closures to noise in the graph, *i.e.*, random link additions. To understand the robustness of the motif closure methods for graphs with noisy and spurious links, we select pairs of nodes uniformly at random that are not linked in G and create a link between each pair. In this set of experiments, we sample $|E|/2$ node pairs (negative/unobserved edges) and add them to G. Results are shown in Table 3.

Result 4. *Motif closures are robust to noise and the robustness of the ranking is often better than triangle closure techniques.*

Runtime performance. We report the average runtime in milliseconds to compute all motif closures for each node pair in G. The methods were implemented in python and all experiments were performed on a laptop (MacBook Pro 2017, 3.1 GHz Intel Core i7, 16GB RAM). For most graphs, it takes *less than a millisecond* on average as shown in Figure 4 and therefore is fast for large-scale ranking problems. Note these results include the runtime to compute 3-node motif closures as well (and thus includes methods such as common neighbors), since the algorithm used to count them leverages 3-node motifs to derive the 4-node motifs efficiently (and many in $o(1)$ constant time) [3].

Result 5. *For any 4-node motif H, counting the number of motif closures W_{ij} that would arise if an edge between i and j was added to G is fast taking less than a millisecond on average across all graphs.*

The runtime can be significantly improved for certain problem settings: Suppose we are interested in only the top-k most relevant node pairs (or items for a user i) given by a ranking from an arbitrary motif closure for motif H, then for possibly many such node pairs, we can avoid computing W_{ij} (*i.e.*, # of instances of motif H in G that would be closed if the node pair (i, j) actually existed/observed in G) altogether by first deriving an upper bound UB of W_{ij} in $o(1)$ constant time and only computing W_{ij} if UB $> \delta$ where δ is the weight of the node pair in the top-k ranking with minimum weight (the node pair with rank k). Since otherwise we know W_{ij} is not large enough to beat the node pair with the k-th largest weight.

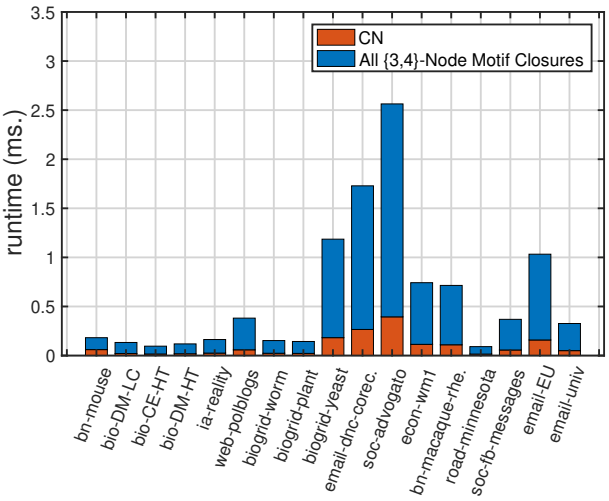

Figure 4: Average runtime in milliseconds to compute *all* $\{3, 4\}$*-node motif closures* for each node pair. Note the average runtime includes the CN and other baselines since they are based on 3-node motifs. Nevertheless, we show the average runtime for CN as well.

5 RELATED WORK

Model-based methods: There has been a lot of work on designing supervised learning methods for link prediction [6, 22, 27, 39]. In general, these methods typically compute a set of features based on the graph topology. For instance, one such method called HPLP+ [27] uses features such as in/out degree, in/out volume, common neighbors, and proximity-based measures such as Katz and shortest path distance between all pairs of nodes. Many of these features are computationally expensive to compute such as the proximity-based measures. Furthermore, HPLP+ and related supervised methods are not incremental and unable to be leveraged for the online setting studied in this work. As an aside, many of the individual features are non-trivial to update in real-time given a new edge that arrives, *e.g.*, the shortest path feature used in [27]

alone in the worst-case may trigger n updates, and each update may also be expensive. There has even been some work that specifically focuses on computationally efficient features [18]. However, some features used by that work still require the full graph and are relatively inefficient to compute and/or update in real-time such as shortest paths. Besides the above difference, this work also leverages ensemble techniques for link prediction, which are even more expensive due to learning a set of supervised models as opposed to a single one. Some work has also been proposed in the case of dynamic networks represented as a sequence of discrete static snapshot graphs [7, 10]. In particular, Berlingerio *et al.* [7, 10] proposed an approach called graph evolution rule miner (GERM) for finding graph-evolution rules with support and confidence above some threshold.

Ensemble methods: There have also been some work on using ensemble methods for link prediction [6, 16–18, 25, 30]. For instance, one simple ensemble technique is to learn a set of m models using different subsets of features, and then apply each one for prediction. Such ensemble methods are known to reduce variance at the expensive of increasing the runtime due to now having to learn a set of m different models and use each to obtain m different predictions, which are then combined to obtain the final prediction. One recent work [30] uses a set of topological features that includes a number of triangle-based features such as Common Neighbors, Jaccard similarity, average clustering coefficient (CC) as well as simpler degree-based features and more computationally expensive path-based features such as the Katz measure, closeness centrality, among others. These features are then used to learn an ensemble of decision tree classifiers (random forest classifier) that are used for link prediction. Other work has used ensemble methods for link prediction in knowledge graphs [25] and miRNA-disease association prediction [12]. Since ensemble methods simply combine multiple models that are learned in a supervised fashion, they come with at least the same set of disadvantages as the supervised link prediction methods described previously. They are also more computationally expensive since instead of learning a single model for link prediction, they learn a set of models that are combined to obtain a slightly more powerful model with less variance. In terms of features, these methods use simple triangle-based closure features and do not leverage any other motifs, or even motifs of a larger size such as the 4-node motifs used in this work.

Unsupervised methods: Unsupervised methods that include triangle closure features such as common neighbors and the ilk are used as features by the vast majority of supervised learning methods for link prediction, *e.g.*, see [6, 18, 27, 30]. As such, the importance of these features should not be understated. Just as common neighbor-based features have been used to achieve superior performance by including all of them as input for learning a supervised model, the motif closure features introduced in our work can also be leveraged in a similar fashion to further improve performance and the generalizability of these models. In this work, we argue that while such triangle-based features (and their variants) serve as a basis for use in supervised link prediction[4], there are many other potentially more important motif closures that can also be used as features to further improve performance. Our work demonstrates this fact empirically as we observe that different motif closures lead to better performance compared to the simple triangle-based methods used in most previous work. Furthermore, the specific motif closure that achieves best performance depends highly on the data. This result has a number of important implications. First, it implies that one should also consider motif closures that are different from triangles. Second, the motif closure that is most predictive of a link depends highly on the underlying network structure and processes that govern it. Third, this result can also be used to improve performance of supervised link prediction methods by leveraging the motif closures as features (going from least to most dense as shown in Figure 1). This can also improve the generalization of such supervised link prediction models.

While the majority of existing methods have primarily been designed for the offline (non-streaming setting), there has recently been some work focusing on link prediction and ranking techniques in the *online streaming setting* where edges arrive continuously over time (which is sometimes referred to as graph streams or edge streams) [2, 38, 43, 44]. This research is the closest work related to our own. However, this work focuses on triangle closure link predictors such as common neighbors or Jaccard similarity [2, 44]. In contrast, our work proposes the notion of a general motif closure for ranking and prediction, and shows that motif closures that go beyond triangles are often better, and the best motif closure depends highly on the underlying characteristics and structural properties of the network.

Network embeddings: Some recent work has used network embedding methods for link prediction [9, 15, 19–21, 23, 34–37]. These works typically learn embeddings for nodes in the graph, then use them to compute embedding vectors for edges. Given these edge embedding vectors along with the labels (*i.e.*, whether an edge is an actual edge or not), they learn a model (*e.g.*, using Logistic Regression) and then apply it to predict whether a given node pair in the test set exists or not. Similarly, some work such as DeepGL [34] is able to learn edge embeddings/features directly given an edge of interest, which may be an actual edge in G or simply a node pair of interest for estimating a weight. While previous work on link prediction in dynamic networks use a discrete approximation of the actual continuous-time dynamic network (edge stream) by constructing a sequence of static snapshot graphs, one recent work called CTDNE [29] uses temporal walks to generalize (static) walk-based embedding methods for link prediction. More recently, [21] proposed a Siamese adaptation of LSTM for link prediction in dynamic networks. Another recent work [37] uses the resource allocation of nodes in the network to learn embeddings for link prediction. However, all these methods are supervised as the embeddings are used as features for learning a model. They also require the full graph. There is also a small but growing amount of research that uses higher-order network motifs (*i.e.*, motifs larger than 3 nodes) as base features to derive node or edge embeddings, which are then used for link prediction [4, 5, 32, 34]. These works are not based on the proposed notion of closing higher-order network motifs, nor do they leverage network motifs directly for link prediction. Instead, they use the output embeddings as features for learning a supervised model for link prediction.

[4]Most work includes many different features based on triangle closure, despite them being very similar to one another, *e.g.*, simple variations with different normalization.

6 DISCUSSION & CONCLUSION

In this paper, we generalized the notion of triangle closure to other higher-order motifs such as the 4-node motif closures shown in Figure 1. Triangle closure has traditionally been used as a fundamental basis for link prediction over the last decade. Indeed, the notion of closing triangles lies at the heart of many important (unsupervised) link prediction techniques including common neighbors, Jaccard similarity, among others. Moreover, these are often used as features for learning a supervised model for link prediction. In this work, we demonstrated that other motif closures are sometimes more predictive than their triangle-based counterparts. This result has three important implications. First, it implies that one should also consider motif closures that are different from triangles. Second, the "best" motif closure (*i.e.*, the motif closure that is most predictive of a link) depends highly on the underlying network structure and processes that govern it. Third, existing supervised learning methods can benefit from these new motif closures by leveraging the full range of motif closures (going from least to most dense as shown in Figure 1).

The findings of this work open up many important future research directions. While our work focused solely on demonstrating that different motif closures (such as the 4-node ones investigated in this work) are sometimes better than triangle closure (and variants based on it), future work should investigate the use of these motif closures as features for supervised learning techniques. For instance, how much improvement in predictive performance can be achieved when including the 4-node motif closures as features for supervised link prediction? Can the proposed motif closures also improve ensemble techniques by reducing variance further and improving the overall quality of predictions? Second, can we characterize the motif closures that perform best for a given network based on its underlying domain? In this work, we only investigated the simplest notion of motif closure, however, future work can investigate using these new motif closures to extend other techniques such as a higher-order Jaccard similarity or higher-order Adamic/Adar measures based on closing higher-order motifs such as 4-cliques, 4-cycles, among others. Finally, another important question is whether using the proposed notion of *closing higher-order motifs* to derive 5-node motif closures is useful or not, *i.e.*, is the additional runtime worth the predictive performance gain?

REFERENCES

[1] Lada A Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social networks* 25, 3 (2003), 211–230.

[2] Nesreen K Ahmed, Nick Duffield, and Liangzhen Xia. 2018. Sampling for approximate bipartite network projection. *IJCAI* (2018).

[3] Nesreen K. Ahmed, Jennifer Neville, Ryan A. Rossi, and Nick Duffield. 2015. Efficient Graphlet Counting for Large Networks. In *ICDM*. 10.

[4] Nesreen K. Ahmed, Ryan A. Rossi, John Boaz Lee, Theodore L. Willke, Rong Zhou, Xiangnan Kong, and Hoda Eldardiry. 2019. role2vec: Role-based Network Embeddings. In *DLG KDD*.

[5] Nesreen K. Ahmed, Ryan A. Rossi, Theodore L. Willke, and Rong Zhou. 2017. Edge Role Discovery via Higher-Order Structures. In *PAKDD*. 291–303.

[6] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. 2006. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*.

[7] Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. 2009. Mining graph evolution rules. In *ECML/PKDD*. Springer, 115–130.

[8] Michael W Berry and Murray Browne. 2005. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. Vol. 17. SIAM.

[9] Stephen Bonner, Amir Atapour-Abarghouei, Philip T Jackson, John Brennan, Ibad Kureshi, Georgios Theodoropoulos, Andrew Stephen McGough, and Boguslaw Obara. 2019. Temporal Neighbourhood Aggregation: Predicting Future Links in Temporal Graphs via Recurrent Variational Graph Convolutions. *arXiv:1908.08402* (2019).

[10] Björn Bringmann, Michele Berlingerio, Francesco Bonchi, and Arisitdes Gionis. 2010. Learning and predicting the evolution of social networks. *IEEE Intelligent Systems* 25, 4 (2010), 26–35.

[11] Sougata Chaudhuri and Ambuj Tewari. 2015. Online ranking with top-1 feedback. In *Artificial Intelligence and Statistics*. 129–137.

[12] Xing Chen, Zhihan Zhou, and Yan Zhao. 2018. ELLPMDA: Ensemble learning and link prediction for miRNA-disease association prediction. *RNA biology* 15, 6 (2018), 807–818.

[13] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *TOIS* 22, 1 (2004), 143–177.

[14] Ernesto Diaz-Aviles, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. 2012. Real-time top-n recommendation in social streams. In *RecSys*. ACM, 59–66.

[15] Aswathy Divakaran and Anuraj Mohan. 2019. Temporal Link Prediction: A Survey. *New Generation Computing* (2019), 1–46.

[16] Liang Duan, Charu Aggarwal, Shuai Ma, Renjun Hu, and Jinpeng Huai. 2016. Scaling up link prediction with ensembles. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 367–376.

[17] Liang Duan, Shuai Ma, Charu Aggarwal, Tiejun Ma, and Jinpeng Huai. 2017. An ensemble approach to link prediction. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 29, 11 (2017), 2402–2416.

[18] Michael Fire, Lena Tenenboim, Ofrit Lesser, Rami Puzis, Lior Rokach, and Yuval Elovici. 2011. Link prediction in social networks using computationally efficient topological features. In *Third International Conference on Privacy, Security, Risk and Trust*. IEEE, 73–80.

[19] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*. 855–864.

[20] Di Jin, Mark Heimann, Ryan A. Rossi, and Danai Koutra. 2019. Node2BITS: Compact Time- and Attribute-aware Node Representations for User Stitching. In *ECML/PKDD*. 22.

[21] Hemant Kasat, Sanket Markan, Manish Gupta, and Vikram Pudi. 2019. Temporal Link Prediction in Dynamic Networks. *MLG KDD* (2019).

[22] Hisashi Kashima and Naoki Abe. 2006. A parameterized probabilistic model of network evolution for supervised link prediction. In *Sixth International Conference on Data Mining (ICDM)*. IEEE, 340–349.

[23] Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems*. 4284–4295.

[24] Sungchul Kim, Nikhil Kini, Jay Pujara, Eunyee Koh, and Lise Getoor. 2017. Probabilistic visitor stitching on cross-device web logs. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1581–1589.

[25] Denis Krompaß and Volker Tresp. 2015. Ensemble solutions for link-prediction in knowledge graphs. In *Workshop on Linked Data for Knowledge Discovery (LD4KD)*.

[26] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *JASIST* 58, 7 (2007), 1019–1031.

[27] Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. 2010. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 243–252.

[28] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. 2010. Introduction to Information Retrieval. *Nat. Lang. Eng.* 16, 1 (2010), 100–103.

[29] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen K. Ahmed, Eunyee Koh, and Sungchul Kim. 2018. Dynamic Network Embeddings: From Random Walks to Temporal Random Walks. In *IEEE BigData*. 1085–1092.

[30] Shruti Pachaury, Nilesh Kumar, Ayush Khanduri, and Himanshu Mittal. 2018. Link Prediction Method Using Topological Features and Ensemble Model. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*. IEEE, 1–6.

[31] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*. 4292–4293. http://networkrepository.com

[32] Ryan A. Rossi, Nesreen K. Ahmed, and Eunyee Koh. 2018. Higher-Order Network Representation Learning. In *Proceedings of the 27th International Conference Companion on World Wide Web (WWW)*.

[33] Ryan A. Rossi, Luke K. McDowell, David W. Aha, and Jennifer Neville. 2012. Transforming graph data for statistical relational learning. *JAIR* (2012), 363–441.

[34] Ryan A. Rossi, Rong Zhou, and Nesreen K. Ahmed. 2018. Deep Inductive Graph Representation Learning. In *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. 14.

[35] R. V. Shaptala and G. D. Kyselev. 2019. Using graph embeddings for wikipedia link prediction. *Combintorial optimization under uncertainty and formal models of expert estimation* (2019), 48.

[36] Han Hee Song, Tae Won Cho, Vacha Dave, Yin Zhang, and Lili Qiu. 2009. Scalable proximity estimation and link prediction in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. ACM, 322–335.

[37] Xinghao Song, Chunming Yang, Hui Zhang, Xunjian Zhao, and Bo Li. 2019. Network Embedding by Resource-Allocation for Link Prediction. In *Pacific Rim*

International Conference on Artificial Intelligence. Springer, 673–683.

[38] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2017. Non-parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[39] Huynh Thanh Trung, Nguyen Thanh Toan, Tong Van Vinh, Hoang Thanh Dat, Duong Chi Thang, Nguyen Quoc Viet Hung, and Abdul Sattar. 2019. A comparative study on network alignment techniques. *Expert Systems with Applications* (2019), 112883.

[40] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, and Emmanuel Müller. 2018. Verse: Versatile graph embeddings from similarity measures. In *WWW*. 539–548.

[41] David H Wolpert and William G Macready. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1, 1 (1997), 67–82.

[42] Hyokun Yun, Parameswaran Raman, and S Vishwanathan. 2014. Ranking via robust binary classification. In *NIPS*. 2582–2590.

[43] Jianpeng Zhang, Kaijie Zhu, Yulong Pei, George Fletcher, and Mykola Pechenizkiy. 2019. Cluster-preserving sampling from fully-dynamic streaming graphs. *Information Sciences* 482 (2019), 279–300.

[44] Peixiang Zhao, Charu Aggarwal, and Gewen He. 2016. Link prediction in graph streams. In *ICDE*. IEEE, 553–564.