Exploiting Cross-Order Patterns and Link Prediction in Higher-Order Networks

Hao Tian
Data Lab, EECS Department
Syracuse University
Syracuse, NY, USA
haotian@data.syr.edu

Shengmin Jin
Data Lab, EECS Department
Syracuse University
Syracuse, NY, USA
shengmin@data.syr.edu

Reza Zafarani
Data Lab, EECS Department
Syracuse University
Syracuse, NY, USA
reza@data.syr.edu

***Abstract*—With the demand to model the relationships among three or more entities, higher-order networks are now more widespread across various domains. Relationships such as multi-author collaborations, co-appearance of keywords, and co-purchases can be naturally modeled as higher-order networks. However, due to (1) computational complexity and (2) insufficient higher-order data, exploring higher-order networks is often limited to order-3 motifs (or triangles). To address these problems, we explore and quantify *similarites* among various network orders. Our goal is to build relationships between different network orders and to solve higher-order problems using lower-order information. Similarities between different orders are not comparable directly. Hence, we introduce a set of general cross-order similarities, and a measure: *subedge rate*. Our experiments on multiple real-world datasets demonstrate that most higher-order networks have considerable consistency as we move from higher-orders to lower-orders. Utilizing this discovery, we develop a new cross-order framework for higher-order link prediction method. These methods can predict higher-order links from lower-order edges, which cannot be attained by current higher-order methods that rely on data from a single order.**

***Index Terms*—higher-order networks, hypergraph, measurement, link prediction**

I. Introduction

With the rapid development of social platforms and online technologies, graph data has become richer not only in scale, but also in variety and complexity. As a result, researchers have proposed several representations for more complex networks. Examples include *Multi-Layer Graphs* [1], which separate distinct types of relationship to multiple layers; *Heterogeneous Information Networks (HIN)* [2] that distinguish different types of nodes and edges; and *Hypergraphs* [3], that extend edges to relations between sets of nodes of unlimited size.

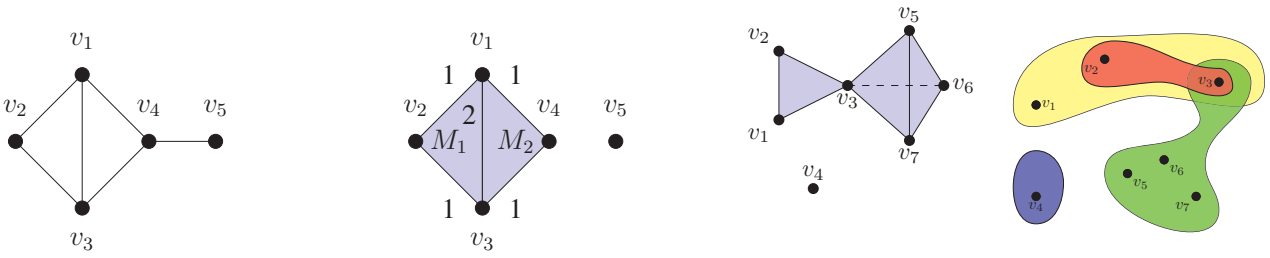
While most graphs only model relationships between two entities (are *dyadic*), we often observe co-occurrences of more than two entities. Examples include co-authorships on publications, co-appearances at group events, or multiple tags in a single news article. In most cases, these relationships are not exactly equal to a set of two-entity relationships. For example, author A, B, and C publishing one paper is not equivalent to AB, BC, AC publishing three. To address this issue, *Higher-Order Networks (HON)* [4] model various *order*s of relationships, e.g. triangles are order-3 relationships.

Higher-Order studies have the potential to explore richer information in networks. However, there exist several difficulties in studies of higher-order network. The main concern is the computational costs. One can naturally model higher-order networks as tensors, extending the adjacency matrix. However, computing on these often large, but sparse, tensors is expensive and seldom provides intuitive information due to uncertainties in tensor algorithms [5]. Another issue is insufficient data on higher-order relations. For example, while users can add unlimited hashtags to a tweet, people often prefer to add fewer than three most relevant hashtags. The consequence is that the space of higher-orders becomes extremely sparse, often negatively impacting the performance of learning models.

Due to these issues, most studies on higher-order networks focus on some fixed small orders. In studies of network *motifs* (frequent subgraphs) [6], the motifs counted are often small and specific, such as triangles [7], chains/loops [8] or cliques [9]. Also, for direct higher-order models such as *simplicial complexes* [10] and *hypergraphs* [11], studies are usually limited to some fixed upper-bound on the larges orders studied [12]. *It is still challenging to include all (across all orders) higher-order information.*

The present work: Cross-Order Similarities. Our work here aims to capture higher-order information in graphs across various orders. Instead of modeling the graph as a whole, we explore relationships between different orders. Instead of mining sparse higher-order information, we find evidence for the existence of higher-order edges from richer lower-order information. To do so, we study whether there exists *consistency* among various cross-order relationships; that is, how the appearances of lower-order edges relate to higher-order ones. Specifically, we study cases in which lower-order edges such as (A, B) appear as subsets of higher-order ones (A, B, C). In this way, any higher-order edge set can be *downgraded* to lower-order edge sets, which also enables comparison between any order edges. First, the higher-order edge set h can be compared with the lower-order edge set l by enumerating *subedges* of order-l in h; Second, two different higher-order edge sets h_1 and h_2 ($h_1 \neq h_2$) can be compared by downgrading both to a unified lower-order space l. Such comparisons reveal cross-order connections

2375-9259/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDMW58026.2022.00156



(a) Simple graph and its corresponding motif graph (b) Simplicial complex (c) Hypergraph

Fig. 1. Comparison among Higher-Order Network Models. (a). A simple graph and its motif graph for the motif: 3-clique (triangles). Motif graphs have the same set of nodes as the original graph, but their edges represent memberships in given motifs. For this example, the given motif is a triangle. As two triangles are in this graph and edge (v_1, v_3) is shared by both triangles, its edge weight is 2. Similarly, edge (v_4, v_5) does not exist (its weight is 0) as it is not in any triangle. (b). A simplicial complex, including a 0-simplex (single node), a 2-simplex (triangle) and a 3-simplex (tetrahedron). Note that any face (sub-simplex) of existing simplex are also included in the simplicial complex, for example (v_5, v_6, v_7). (c). A hypergraph similar to (b). Unlike simplicial complexes, any subedge of hyperedges can appear in the edge set independently, such as (v_1, v_2, v_3) and (v_2, v_3) that are two different edges.

especially for higher-order edges, which allows on to study their emergence and structures. By studying these connections, we present a new family of link prediction methods for higher-order edges.

The main contribution of this paper can be summarized as follows:

1) We propose techniques to measure cross-order consistency in higher-order networks. Such measurements can be used naturally as network features in machine learning methods for higher-order networks;
2) Through extensive experiments, we show various patterns of cross-order relationships from 19 real-world datasets; and
3) Based on the insights derived from cross-order consistencies, we develop a series of new higher-order link prediction methods.

The rest of this paper is organized as follows. We detail the related work in Section II. We introduce the preliminaries and models of higher-order networks in Section III. We propose our similarity measure in Section IV and show some findings on real-world datasets. Using case studies, we present a new higher-order link prediction method in Section V. We conclude with a discussion and potential future work in Section VI.

II. Related Work

We survey related research from two perspectives. First, we review various higher-order network models that study higher-order interactions in networks. Second, we review algorithms that can be applied to higher-order networks.

A. Higher-Order Network Modeling

We survey higher-order network models and their applications; see figure 1 for details. In earlier studies, Milo et al. [6] noticed that specific subgraphs (denoted as *network motifs*) have unusual frequencies of appearance. Such motifs are found to be closely related to network functionalities [13], [14] and can be used as features to identify the types of networks [15]. Using network motifs, Benson et al. [4] have developed a series of techniques for higher-order network analysis. Yin et al. [16] propose a clustering coefficient based on higher-order cuts, clustering the graph by minimizing the cost of breaking the motifs. Network motifs are also used for higher-order measurements such as *modularity* [1] and representation learning [17].

Besides exploring higher-order structures such as motifs in dyadic networks, some studies analyze higher-order networks using models specifically designed for such networks. A *simplicial complex* [10] can be interpreted as one kind of such higher-order generalizations of the graph: a collection of nodes, edges, triangles, and higher-order entities. Each entity of size k represents an interaction that involves k nodes simultaneously. One of the common usages of a simplicial complex is to extract networks from geometric information. Such a process is called *filtration*, and is widely used in sensor coverage [18], disease detection [19], and mobility analysis [20], among others. There are also various network analysis tools for simplicial complexes, including configuration models [21], random walk techniques [22] and sparsification methods [12].

Another higher-order generalization of a graph is a *hypergraph* [11]. Its main difference from the simplicial complex is that the hypergraphs are not inclusive by default. As shown in Figure 1 (c), the hypergraph can have edges (v_1, v_2, v_3) and (v_2, v_3) independently. Many graph-based concepts are extended to hypergraphs, including random walks [23], centralities [24] and tensors (adjacency matrix) [25]. For applications, hypergraphs are used in classification [26], clustering [27] and generative models [28].

B. Higher-Order Link Prediction

For predicting links in higher-order networks, most edge scoring methods for dyadic graphs can be easily extended to higher-order networks. Examples include neighborhood similarity methods such as *Adamic-Adar* [29], as well as path-based methods like *Katz* [30] and *PageRank* [31].

In addition, some methods are developed specifically for higher-order models. Network motif counts can be used either

directly as local features [32] for unsupervised scoring or can be used as link prediction features in a supervised setting [33]. For simplicial complexes, new higher-order links can be predicted using local counts of their existing sub-edges that form a smaller simplex [34]. As for hypergraphs, recent studies have applied neural networks to achieve this goal [35].

In general, link prediction methods that utilize the whole-graph structure (are *global*) may be able to reach higher accuracy but require extensive computational power. Such a trade-off is intensified in higher-order networks, making them less scalable especially for higher-order edges.

III. Preliminaries of HON Modelings

A. Hypergraph

Here, we use $\mathcal{G} = (V, \mathcal{E})$ to denote a hypergraph, where the vertex set V remains the same. The edge set $\mathcal{E} = \{e | e \subseteq V\}$ is the set of subsets of V. We only consider undirected hypergraphs. The main difference from dyadic graphs is that the edges become sets/unordered tuples instead of pairs.

A k-*uniform* hypergraph is a hypergraph where each edge contains k vertices.

Random hypergraph $\mathcal{G}^k(n, M)$ is defined as a hypergraph chosen uniformly at random from the family of all possible $\binom{\binom{n}{k}}{M}$ k-uniform hypergraphs with vertex set $n = |V|$ and M number of edges [36].

B. Modeling in this Paper

We model higher-order networks naturally using a hypergraph. For example, at time t, co-occurrence of node v_1, v_2 and v_3 yields an order-three edge: (v_1, v_2, v_3). However, slightly different from traditional hypergraphs, *we store edges of different orders separately* instead of an unordered collection of hyperedges. This allows us to explore the influence across orders and measure similarities across different orders. Hence, we represent a higher-order graph as

$$\mathcal{G} = (V, E_1, E_2, \dots), \tag{1}$$

where V is the set of vertices, and

$$E_i \subseteq \{(x_1, \dots, x_i) \mid (x_1, \dots, x_i) \in V^i \text{ and } x_1 \neq \dots \neq x_i\}$$

is the set of order-i edges. Each layer (V, E_k) is a k-uniform hypergraph, which is a subgraph of $\mathcal{G}$. Note that all orders of edges are among Cartesian powers of the same set of vertices. Here, we do not consider the direction of edges, i.e., any edge in E_i is an unordered tuple of vertices.

We define a *subedge* relationship when a lower-order edge is a subset of a higher-order edge, that is $e_l \subsetneq e_h$, where $e_l \in E_l, e_h \in E_h$ and $l < h$. Here we call e_l an l-*subedge* of e_h.

Depending on the dataset and/or the purpose of the study, there could be a weight function $w : E_i \to \mathbb{R}$ for all higher-order edges. We mainly focus on analyzing the structure of higher-order networks so for the majority of our analysis, the graph is unweighted.

IV. Cross-Order Similarities

We first present an approach to measure similarities between any orders, along with subedge distributions. In particular, we explore whether the existence of higher-order edges is consistent with lower-order ones. For example, how likely are the order-2 subedges of E_3 to appear in E_2? We further generalize the similarities to spaces lower than both edge sets, and quantitatively analyze their relationships. Finally, we collect these scores from real-world datasets and report our findings.

A. Similarity of E_h and E_l

For higher-order edges, we enumerate their possible lower-order subedges. This approach allows comparing them to the original lower-order edges. For a given hypergraph $\mathcal{G} = (V, E_1, E_2, \dots)$, we want to measure the similarity of two layers E_h and E_l, where $h > l$. First, we *downgrade* the order of E_h to order l by enumerating all its l-subedges of every single edge.

$$E_{h\to l} = \{e | e \subseteq e_h \text{ where } e_h \in E_h, |e| = l\}, \tag{2}$$

where E_{h-l} is the downgraded edge set of E_h at order-l.

Then we can define the similarity of E_h and E_l by *Jaccard Similarity* [39] of $E_{h\to l}$ and E_l.

$$sim(E_h, E_l) = Jaccard(E_{h\to l}, E_l) = \frac{|E_{h\to l} \cap E_l|}{|E_{h\to l} \cup E_l|}. \tag{3}$$

The similarity of edge sets is a symmetric measure. However, in real-world graphs the density of edges can be extremely imbalanced on higher- and lower-orders. As a result, we also define two one-way similarities.

$$hsim(E_h, E_l) = \frac{|E_{h\to l} \cap E_l|}{|E_{h\to l}|}; \tag{4}$$

$$lsim(E_h, E_l) = \frac{|E_{h\to l} \cap E_l|}{|E_l|}. \tag{5}$$

Here, $hsim$ measures the overlap rate over higher-order subedges, while $lsim$ measures the overlap rate over lower-order edges. Once we explore the network step by step across different orders, these one-way similarities provide more utility than the symmetric measure.

B. Subedge Distribution

In addition to the overall similarities, we also investigated the subedge distribution from higher→lower edges. Each hyperedge of order-h has $\binom{h}{l}$ order-l subedges. We count the number of occurrences of those subedges in E_l. For each $e_h \in E_h$,

$$subedge\ rate(e_h, E_l) = \frac{\left|\{e_l | e_l \subseteq e_h \text{ and } e_l \in E_l\}\right|}{\binom{h}{l}}, \tag{6}$$

abbreviated as se_rate. Examining the subedge rate of all hyperedges in E_h, we obtain a discrete distribution of subedge existences from E_h to E_l, which is

$$P(X = a) = \frac{|\{e_h | e_h \in E_h, se\_rate(e_h, E_l) = a\}|}{|E_h|}. \tag{7}$$

Graphs	Vertices	Timestamps	Unique Edges	Max Order	Average Order
coauth-DBLP [34]	1,924,991	3,700,067	2,599,087	25	2.78
coauth-MAG-Geology [34]	1,256,385	1,590,335	1,207,390	25	2.78
coauth-MAG-History [34]	1,014,734	1,812,511	895,668	25	1.31
congress-bills [34]	1,718	260,851	85,082	25	3.66
contact-high-school [34]	327	172,035	7,937	5	2.05
contact-primary-school [34]	242	106,879	12,799	5	2.10
DAWN [34]	2,558	2,272,433	143,523	16	1.58
email-Enron [34]	143	10,883	1,542	18	2.47
email-Eu [34]	998	234,760	25,791	25	2.33
NDC-classes [34]	1,161	49,724	1,222	24	3.14
NDC-substances [34]	5,311	112,405	10,025	25	1.85
tags-ask-ubuntu [34]	3,029	271,233	151,441	5	2.71
tags-math-sx [34]	1,629	822,059	174,933	5	2.19
tags-stack-overflow [34]	49,998	14,458,875	5,675,497	5	2.97
threads-ask-ubuntu [34]	125,602	192,947	167,001	14	1.80
threads-math-sx [34]	176,445	719,792	595,778	21	2.24
threads-stack-overflow [34]	2,675,955	11,305,343	9,705,709	25	2.26
twitter-hashtag-covid19 [37]	12,033	59,892	10,074	33	2.21
twitter-hashtag-ira [38]	190,481	2,585,982	242,988	30	1.44

TABLE I
DATA STATISTICS

Here, we simply denote this distribution as $distr(E_h, E_l)$.

Although the subedge distribution is a measurement at different levels with $lsim$, they are indeed similar to each other. The main difference is that $lsim$ first downgrades all the subedges to a set without duplicate; While subedge rate records the overlap rate at the level of individual higher-order edges. For a random hypergraph, the expectation of the mean of the subedge distribution is equal to the expectation of $lsim$. This is obvious as in a random graph, order-l subedges are selected with equal probabilities, and hyperedges of order-h are generated independently with order-l. No matter what the sampling process is, the expectation is always equal to $|E_l|/\binom{|V|}{l}$.

In real-world graphs, their difference can somehow reflect the graph structure as the intersecting subedges of E_h are counted multiple times by subedge rates. Let us consider an extreme example: all hyperedges in E_3 are intersecting at the same order-2 subedge e. If all subedges in $E_{3\to 2}$ except e belong to E_2, the mean of the subedge distribution is equal to 2/3, while $hsim(E_3, E_2)$ goes to 1. On the contrary, if only e belongs to E_2, then the mean of subedge distribution equals 1/3 while the $hsim(E_3, E_2)$ goes to 0.

C. k-Similarity of E_h and E_l

In Equation 3, we compare the similarity at order-l. We can further compare the similarities of two layers at order less than l. The cross-order similarity can be further generalized to order-k, where $1 \le k \le l$, noted as $k - sim(E_h, E_l)$ (for simplicity, from this point denoted as $k - sim$).

$$k{-}sim(E_h, E_l) = Jaccard(E_{h\to k}, E_{l\to k}) = \frac{|E_{h\to k} \cap E_{l\to k}|}{|E_{h\to k} \cup E_{l\to k}|}. \tag{8}$$

Here, we downgrade the edge sets of both orders to be compared in a lower-order space. When $k = l$, the $k - sim$ is exactly the sim defined in Equation 3.

For given E_h and E_l, their $k - sim$ for different k are correlated with each other, as subedges of order $k-1$ can be interpreted as further subedges of order k. Assume we know the $k{-}sim = p$, and the total number of subedges in $|E_{h\to k} \cup E_{l\to k}| = X$. Then, the lower and upper bounds of $(k{-}1){-}sim$ are

$$\left[\frac{p}{1+(1-p)\sqrt[k]{k!pX}}, \frac{p+p\sqrt[k]{k!(1-p)X}}{1+p\sqrt[k]{k!(1-p)X}}\right] \tag{9}$$

Proof. First, pX is the number of overlapped subedges of $E_{h\to k}$ and $E_{h\to l}$, and $(1-p)X$ is the rest. The lower bound occurs when the overlapped subedges form cliques while the non-overlapped subedges are not connected. In this way, at $(k-1)$ order the overlapped subedges is at minimum while the non-overlapped is at maximum. Here, we assume n is the number of nodes in $E_{h\to k} \cap E_{l\to k}$ in the form of a complete graph; then $\binom{n}{k} = pX$. The number of overlapped $(k-1)$ subedges is $A = \binom{n-1}{k} = pXk/(n{-}k{+}1)$. On the other hand, each subedge from non-overlapped side derive $\binom{k}{k-1} = k$ subedges that are still not overlapped, the total number is $B = (1-p)Xk$. So the lower bound of $(k-1) - sim$ is $A/(A+B)$. From the inequalities $\frac{n^k}{k!} \ge \binom{n}{k} \ge \frac{(n-k)^k}{k!}$ we can derive the upper and lower bound of n represented by k, p and

Dataset	E_h	E_l	2-sim	2-hsim	2-lsim	3-sim	3-hsim	3-lsim	4-sim	4-hsim	4-lsim
tags-ask-ubuntu	E_3	E_2	**0.2489**	**0.2935**	**0.6212**	-	-	-	-	-	-
	E_4	E_2	**0.2243**	**0.2582**	**0.6310**	-	-	-	-	-	-
	E_5	E_2	**0.2117**	**0.2452**	**0.6074**	-	-	-	-	-	-
	E_4	E_3	0.3187	0.4510	0.5208	**0.1024**	**0.1349**	**0.2985**	-	-	-
	E_5	E_3	0.3032	0.4315	0.5049	**0.0843**	**0.1024**	**0.3232**	-	-	-
	E_5	E_4	0.3206	0.4824	0.4889	0.1289	0.1942	0.2771	**0.0409**	**0.0530**	**0.1522**
coauth-DBLP	E_3	E_2	**0.0992**	**0.1275**	**0.3095**	-	-	-	-	-	-
	E_4	E_2	**0.0547**	**0.0696**	**0.2042**	-	-	-	-	-	-
	E_5	E_2	**0.0342**	**0.0465**	**0.1144**	-	-	-	-	-	-
	E_4	E_3	0.1248	0.2027	0.2450	**0.0525**	**0.0710**	**0.1680**	-	-	-
	E_5	E_3	0.0748	0.1382	0.1401	**0.0225**	**0.0297**	**0.0854**	-	-	-
	E_5	E_4	0.1148	0.2258	0.1893	0.0610	0.1048	0.1272	**0.0283**	**0.0391**	**0.0936**

TABLE II
k-SIMILARITIES

X, which is $\sqrt[k]{k!pX} + k \geq n \geq \sqrt[k]{k!pX}$. Taking the upper bound of n, we get the final lower bound of $(k-1) - sim$.

Similarly, the upper bound is calculated by taking overlapped subedges at maximum and the non-overlapped subedges at minimum. □

Note that the expectation of $(k-1)-sim$ is equal to $k-sim$ in random graphs. But in real-world graphs, highly connected structures are usually overlapped, so we expect $(k-1)-sim$ to be greater than $k-sim$ in most cases.

D. Observations on Real-World Datasets

By calculating the similarities and subedge distributions of 19 real world datasets, we find two representative patterns, where each of the 19 datasets exhibits one of these patterns. Here we select one dataset from each category as examples to compare their similarities and subedge distributions – tags-ask-ubuntu and coauth-DBLP. We summarize all datasets we used in this and the following sections in Table I.

First, we calculate all cross-order k-similarities of both datasets. As shown in Table II, for each dataset we divide the cross-order similarities into three sets, depending on E_l. For the same E_l, $l-sim$s (bold) are the original cross-order similarities defined in equation 3, while smaller-order similarities are also collected. For *tags-ask-ubuntu* the similarities from different orders to same E_l are extremely close, for example $sim(E_4, E_3)$ and $sim(E_5, E_3)$. Such consistency indicates that for this kind of datasets, subedges from various higher-orders share similar aspects over lower-order spaces. But for *coauth-DBLP*, the similarities obviously decrease when the order gap increases. For these kind of datasets, similarities are higher for orders closer to each other. Another finding is for same E_h and E_l, their $k-sim$ increases significantly when k is decreased. This finding also matches our discussion after the similarity bounds 9, in which highly connected components are more likely to be overlapped across orders rather than scattered parts.

We can also show these two patterns from the perspective of subedge distribution. As shown in figure 2 (a)(b)(c)/(d)(e), subedge distributions of *tags-ask-ubuntu* have similar shapes and means, where their means are similar as $lsim$ in table II (we have already analyzed this after equation 6). On the contrary, for *coauth-DBLP* (f)(g)(h)/(i)(j) they have similar shapes but decreasing means for larger order gaps. This discovery provides a chance to imitate the cross-order relations of existing orders to predict the unknown higher-order structures.

Note that there is a subedge rate that measures the correlation between higher-order subedges to lower-order edges. One may wonder if there is a symmetric relationship between how lower-order edges aggregate to become higher-order edges. For example, at what rate do triangles in order-2 become an edge in order-3? Unfortunately, we find that this rate is extremely low among all datasets. That is, *lower-order structures provide insufficient evidence for formation of higher-order edges*.

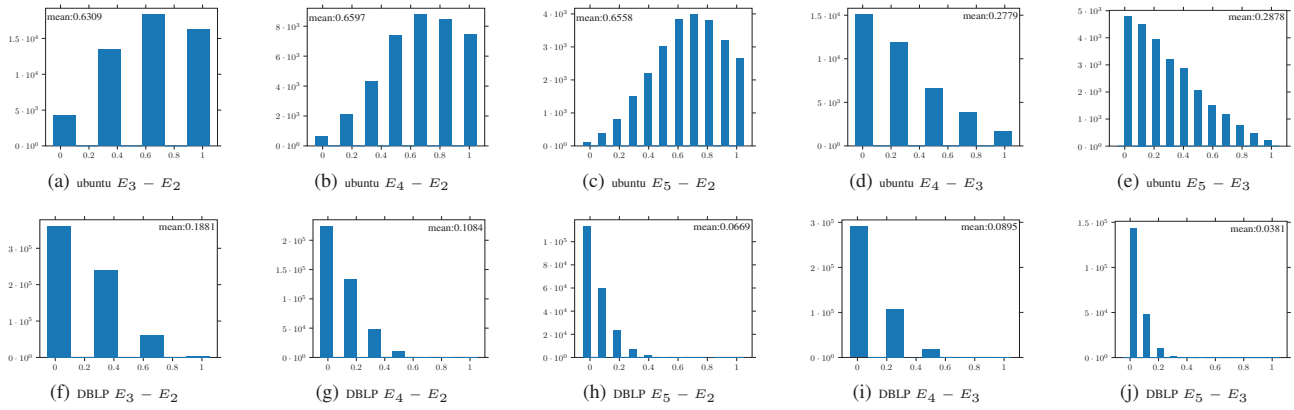
V. CROSS-ORDER LINK PREDICTION

Knowing that there exist one-way consistencies across various orders, we introduce a series of cross-order link prediction approaches that harness this finding. Our goal is to predict higher-order edges from richer lower-order edges. For example, we try to predict higher-order edges, such as order-3,-4, and -5 from denser lower-order edges. Note that here we assume we do not know any existing hyperedge from the target order.

To that end, we first introduce our specific experimental setup and some baselines. Then we propose our cross-order link prediction methods based on this consistency. Finally, we summarize the results.

A. Experimental Setups and Baselines

Formally speaking, our goal is to predict order-3, order-4, and order-5 edges, using one or two lower-order edge sets. Note that training and testing instances are not in the same space, and it is impossible to enumerate all possible links in higher-order space. Hence, we generate random negative candidates by 30%/40%/50%/60%/70% from *random walk sampling* and the rest from *random node sampling*. That is, to increase the difficulty as random walk sampling is more likely to generate negative edges with higher rate of connectivity. We



(a) ubuntu $E_3 - E_2$ (b) ubuntu $E_4 - E_2$ (c) ubuntu $E_5 - E_2$ (d) ubuntu $E_4 - E_3$ (e) ubuntu $E_5 - E_3$

(f) DBLP $E_3 - E_2$ (g) DBLP $E_4 - E_2$ (h) DBLP $E_5 - E_2$ (i) DBLP $E_4 - E_3$ (j) DBLP $E_5 - E_3$

Fig. 2. Consistency of subedge rate distributions illustrated using two typical cases. (a)-(e) stable discrete distribution of tags-ask-ubuntu. (f)-(j) decreasing stable discrete distribution of coauth-DBLP. X-axis indicates ratio of subsets from higher-order edges existing in lower-order space. Y-axis counts the number of higher-order edges with such ratio. These distributions exhibit high consistency under the setting of same lower-order spaces, e.g. (a)(b)(c) are from higher-order to order-2. We calculate means of each distribution at the corner.

shuffle the percentage of random walk sampling to see how it reflects on the performance.

For baselines, we generalize several regular link prediction methods. Most similarity-based methods for dyadic graphs can be extended to higher-order networks by summing up scores of its subedge similarities in the downgraded order-2 graph. In this paper we apply several state-of-the-art methods for comparisons – *Adamic-Adar* [29], *Jaccard* similarity [39], *Preferential Attachment* [40] and *Resource Allocation* [41].

Besides the generalized baselines from common graphs, we add two more baselines that count similarities using cross-order information. The idea of these baselines are exactly same as the work of Benson et al. [34], which scores the candidates directly by the number of subedges in the lower-order edge set. The only difference is that in this work we consider the graph as unweighted, while in the original paper they applied *Harmonic/Geometric/Arithmetric* means of weighted edges. These methods are considered optimistic methods, since more supports from lower-order spaces directly lead to higher prediction scores. The experimental results from [34] have already shown that the performances of such cross-order methods are competitive (outperform for some datasets) with those of the classic scoring methods.

Baseline 1: Single-order Prediction. We formally introduce our scoring algorithm from basic settings – predict higher-order edges E_h from single lower-order layer E_l. For a candidate higher-order edge $|e| = h$, we directly score it by is subedge rate from equation 6

$$score_S(e) = se\_rate(e, E_l) \tag{10}$$

Baseline 2: Multi-order Prediction. Once using multiple orders of edges as training instances, we create a linear combination of multiple order scores to enhance precision, controlled by a weighting parameter α.

$$score_M(e) = \sum_{i=1}^{n} \alpha_i \times se\_rate(e, E_{l_i}) \tag{11}$$

All of these baselines rely on a common assumption: new edges are likely to appear to form more well-connected components. This is generally true for most real-world graphs. However, in higher-order networks there might be an opposite power that comes from the sparsity of higher-order edges. Taking this effect into account, there is the possibility of further improving the accuracy of classic methods.

B. Proposed Method Based on Subedge Distribution

Here we define a general framework of link prediction that fully utilizes the consistencies of subedge distributions from different orders. In Section IV-D we have already shown that there exist consistencies, either stable or decreasing, of subedge distributions from different higher-orders to same lower-orders. The first step of this link prediction method is to predict the subedge distribution of the unknown order to the existing orders.

$$score_{SD}(e) = \sum_{i=1}^{n} \alpha_i \times f(se\_rate(e, E_{l_i})) \tag{12}$$

For subedge rates of candidate edges to different training orders, we apply a function f that depends on all accessible subedge distributions from training instances. This function maps the actual subedge rate to a score based on the generated distribution that is similar to the existing one. Assume our training edge sets are $E_{l_1}, E_{l_2}, \ldots, E_{l_M}$ and the testing edges are from order h, where $l_1 < l_2 < \cdots < l_M < h$. In order to keep generality, we do not assume $l_1, l_2, \ldots$ to be continues orders, for example it can be order- 1,3,4.

Here we consider the following three conditions of $f(se\_rate(e, E_{l_i}))$:

1) $l_i = l_M$: When the training edge set E_{l_M} is the largest order of all training sets, there do not exist any other higher-order training instance to learn the subedge distribution to

Training	Testing	Adamic-Adar	Jaccard	Preferential Attachment	Resource Allocation	Single-order Prediction	Multi-order Prediction	Subedge Distribution Prediction
E_2	E_3	**0.7444**	0.4290	0.7413	0.7168	0.5283	-	-
E_2	E_4	0.6228	0.4304	0.6457	0.6161	0.4645	-	-
E_3	E_4	0.7991	0.7326	0.7460	0.7987	0.7946	-	-
E_2,E_3	E_4	0.7466	0.4054	0.7137	0.7475	-	0.6137	**0.9099**
E_2	E_5	0.5485	0.4211	0.5747	0.5626	0.4443	-	-
E_3	E_5	0.7597	0.7146	0.6834	0.7611	0.8031	-	-
E_4	E_5	0.8007	0.8144	0.7168	0.8040	0.7681	-	-
E_2,E_3	E_5	0.6806	0.3893	0.6498	0.6839	-	0.6402	**0.8542**
E_2,E_4	E_5	0.7225	0.4031	0.6657	0.7256	-	0.5175	**0.8550**
E_3,E_4	E_5	0.7765	0.7101	0.6953	0.7774	-	0.8004	**0.8034**
E_2,E_3,E_4	E_5	0.8026	0.4315	0.7617	0.8056	-	0.7153	**0.8621**

TABLE III
AUC-PR OF LINK PREDICTION (50% RANDOM WALK, TWITTER-HASHTAG-COVID19)

E_{l_M}. Without any per-knowledge one can only directly use the subedge rate without any change, such that

$$f(se\_rate(e, E_{l_M})) = se\_rate(e, E_{l_M}).$$

2) $l_i = l_{M-1}$: When the target edge set is the second largest order $E_{l_{M-1}}$, we already have one subedge distribution to $E_{l_{M-1}}$ in training set, that is from E_{l_M}. Here we assume that the unknown subedge distribution of $(E_h, E_{l_{M-1}})$ will be similar to $distr(E_{l_M}, E_{l_{M-1}})$. Since there is only one reference, we simply assume that they have the same mean. Now the goal is to convert the existing distribution to a synthetic distribution with same possible values of the random variable. For $distr(E_{l_M}, E_{l_{M-1}})$ the random variable $\mathbf{X} = 0, 1/N_1, 2/N_1, \ldots, 1$ where $N_1 = \binom{l_M}{l_{M-1}}$, and for $distr(E_h, E_{l_{M-1}})$ the random variable $\mathbf{Y} = 0, 1/N_2, 2/N_2, \ldots, 1$ where $N_2 = \binom{h}{l_{M-1}}$.

Now we want to build a random variable with the same mean of $\mathbf{X}$ but has same possible values with $\mathbf{Y}$, say $\mathbf{X}^*$. Let $\mathbf{Z} \sim U(0, 1)$ be a uniform distribution, then let $\mathbf{L} = N_1\mathbf{X} + \mathbf{Z}$ becomes a continuous random variable of $(0, N_1 + 1)$. Here we apply a linear transformation to $\mathbf{L}$ to scale it to $(0, N_2+1)$, such that

$$\mathbf{L}^* = \frac{N_2 + 1}{N_1 + 1} \cdot \mathbf{L}. \tag{13}$$

By applying the integral binning operator:

$$bin(\cdot) = \cdot - mod(\cdot, 1),$$

we obtain a discrete distribution $bin(\mathbf{L}^*)$, which equals to $N_2\mathbf{X}^*$. Finally, $\mathbf{X}^* = bin(\mathbf{L}^*)/N_2$ is the estimated subedge distribution of $distr(E_h, E_{l_{M-1}})$, noted as $\hat{distr}$. Then

$$f(se\_rate(e, E_{l_{M-1}})) = P(\mathbf{X}^* = se\_rate(e, E_{l_{M-1}})),$$

where $\mathbf{X}^* \sim \hat{distr}(E_h, E_{l_{M-1}})$.

3) $l_i = l_{M-2}, l_{M-3}, \ldots, l_1$: When the target edge set is neither the first nor the second highest order, there are more than one subedge distributions from the training sets. It is necessary to learn the trend of means of the subedge distributions, either stable or decreasing. Formally, the problem is to predict the distribution $distr(E_h, E_{l_i})$ from distributions we already have $distr(E_{l_M}, E_{l_i})$, $distr(E_{l_{M-1}}, E_{l_i})$,..., $distr(E_{l_{i+1}}, E_{l_i})$. Since there is no need to change the shape of distributions, we just calculate a expected mean of $distr(E_h, E_{l_i})$ from the means of the given distributions. This process is carried out by a simple linear regression where $x_i = l_{i+1}, \ldots, l_M$ and $y_i = \mu_{i+1}, \ldots, \mu_M$.

After obtaining $\hat{\mu_h}$, we transform the $distr(E_{l_M}, E_{l_i})$ to $\hat{distr}(E_h, E_{l_i})$ following the same steps above. Note here we use l_M rather than else because l_M is the closest order with h. The only difference is in Equation 13, we also apply the expected mean to the transformation, such that

$$\mathbf{L}^* = \frac{N_2 + 1}{N_1 + 1} \cdot \frac{\hat{\mu_h}}{\mu_M} \cdot \mathbf{L}. \tag{14}$$

Finally, the function

$$f(se\_rate(e, E_{l_i})) = P(\mathbf{X}^* = se\_rate(e, E_{l_i})),$$

where $\mathbf{X}^* \sim \hat{distr}(E_h, E_{l_i})$.

Note that both domain and range of function f are $[0, 1]$, so it can be used as single order similarity without any other scaling. In summary, the proposed method predicts the subedge distribution from the testing order to each training order. And it applies probabilities from the predicted distributions as scores. Such a mapping prevents the model to be too optimistic of predicting most well-connected hyperedges.

C. Overall Performances

For each dataset, we create negative samples by varying random walk ratios. For each training and testing set, we preform all combinations of cross-order link prediction from the lower-order(s) to higher-order. In total, we tested 7 methods on **5 samples** $\times$ **11 tasks** $\times$ **19 datasets**. All weights are selected by a simple grid search. We observe that similarity scores are not so sensitive to weights; basically, they just represent the ranking of significance of each component.

As an example, Table III shows a single set of link prediction results. *Single-order Prediction* prediction can only be applied to single-order training sets, while *Multi-order Prediction* and *Subedge Distribution Prediction* can only be applied to training sets with multiple orders. Here, we further split the tasks into three meta-groups based on the testing set,

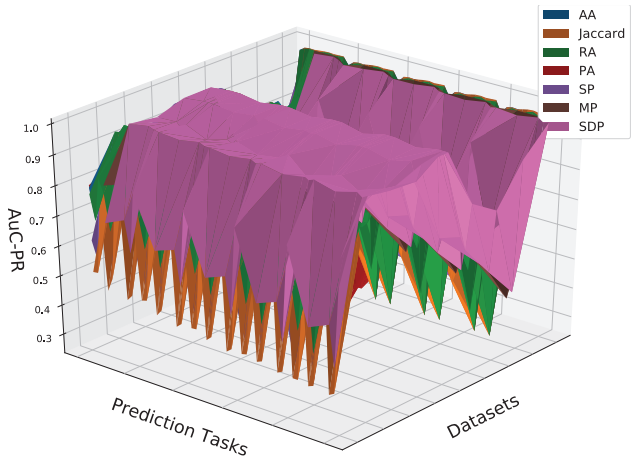


Fig. 3. Summarizes all AuC-PR results of all datasets and settings. X- and Y-axis are datasets and various training/testing sets. The Z-axis represents the AuC-PR of the prediction result. Some methods are just fit for single/multiple training orders, so for different surfaces there may not exist points with corresponding X- and Y- coordinates. However, this do not influence the observation of the overall performances.

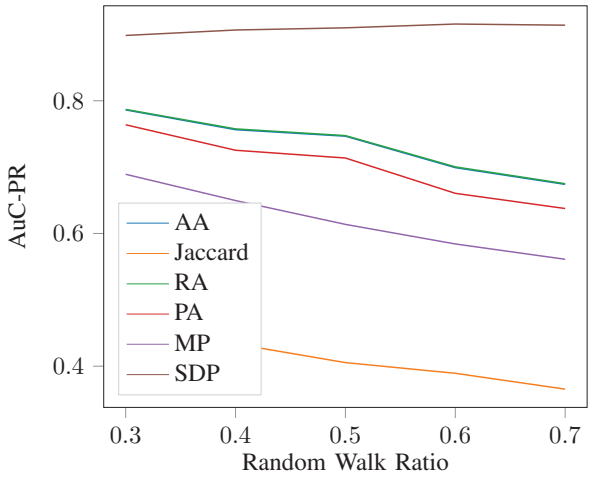


Fig. 4. Prediction performances of all methods while varying the ratio of negative instances sampled by random walk (an example of twitter-hashtag-covid19, training by E_2 and E_3, predicting E_4). Apparently, all methods except the proposed method, are vulnerable to the random walk sampled negative edges. Note that AA and RA are overlapped.

which shows the influence of selecting different training sets. Among all baselines, *Adamic-Adar* and *Resource Allocation* show relatively stable performances, while *Single-/Multi- order Prediction* can outperform other baselines for just some cases. The proposed method is stable and always outperforms the others.

Figure 3 summarizes all AuC-PR results as a surface. The proposed *Subedge Distribution Prediction (SDP)* method clearly outperforms other methods in almost all cases.

D. Predict Random Walk Samples

We explore the influence of sampling more negative instances from random walk sampling. A random walk process on existing edges can generate hyperedges with more common neighbors or subedge relationships with existing edges. For example, with the training set E_2, E_3, E_4, we perform a hypergraph random walk and generate a hyperedge with every 5 unvisited nodes. Such an edge will likely get a higher score from those optimistic prediction methods. However, for the proposed method more subedges in training instances may not lead to a higher score, since these are already an assumption on subedge distributions.

As shown in Figure 4, when the percentage of negative instances sampled by random walk is increased, all link prediction methods are less accurate except *Subedge Distribution Prediction (SDP)*. This confirms our discovery that similarity-based methods tend to be too optimistic with positive cross-order correlations. Only *Subedge Distribution Prediction (SDP)* adapts to different cross-order distributions as it learns and simulates the cross-order distribution from the training set.

VI. Discussion and Future Work

Higher-order networks are proposed and modeled in various domains. Due to computational costs, most studies stop at order-3 (triangles). Our proposed solution to address this issue involves building relationships between different orders and solving higher-order problems in lower-order spaces. Thus, whether there exists any consistency becomes crucial for further research and applications.

We propose cross-order consistency measures for higher-order networks. Our experiments on real-world data show that higher-order links exhibit only one-way consistency across different orders. Harnessing this similarity, we propose new cross-order link prediction method based on lower-order edges, which are richer and easier to be processed.

There exist several limitations to our work: (1) the proposed approach cannot still effectively predict occurrences of higher-order edges from any single lower-order space. The sufficient condition for forming a higher-order edge from lower-order edges is still unclear; (2) while dividing hypergraphs into fixed orders simplifies the computation, it may also lose some information on the local cross-order structures. Our future work aims to address these limitations and build more comprehensive and computationally friendly higher-order network models.

Acknowledgment

This research was supported in part by the National Science Foundation under award CAREER IIS-1942929.

References

[1] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of Complex Networks*, vol. 2, no. 3, pp. 203–271, 07 2014.

[2] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han, "Heterogeneous network representation learning: Survey, benchmark, evaluation, and beyond," 2020.

[3] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proceedings of the 19th NIPS*. Cambridge, MA, USA: MIT Press, 2006, p. 1601–1608.

[4] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *CoRR*, vol. abs/1612.08447, 2016.

[5] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[6] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[7] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," 2010.

[8] C. Duclos, D. Nadin, Y. Mahdid, V. Tarnal, P. Picton, G. Vanini, G. Golmirzaie, E. Janke, M. S. Avidan, M. B. Kelz, G. A. Mashour, and S. Blain-Moraes, "Brain network motifs are markers of loss and recovery of consciousness," *bioRxiv*, 2020.

[9] A. Sizemore, C. Giusti, A. Kahn, J. Vettel, R. Betzel, and D. Bassett, "Cliques and cavities in the human connectome," *Journal of Computational Neuroscience*, vol. 44, pp. 1–31, 02 2018.

[10] A. Hatcher, C. U. Press, and C. U. D. of Mathematics, *Algebraic Topology*, ser. Algebraic Topology. Cambridge University Press, 2002.

[11] C. Berge, *Graphs and Hypergraphs*. GBR: Elsevier Science Ltd., 1985.

[12] B. Osting, S. Palande, and B. Wang, "Towards spectral sparsification of simplicial complexes based on generalized effective resistance," *CoRR*, vol. abs/1708.08436, 2017.

[13] N. Kashtan and U. Alon, "Spontaneous evolution of modularity and network motifs," *Proceedings of the National Academy of Sciences*, vol. 102, no. 39, pp. 13 773–13 778, 2005. [Online]. Available: https://www.pnas.org/content/102/39/13773

[14] O. Sporns and R. Kötter, "Motifs in brain networks," *PLoS biology*, vol. 2, p. e369, 12 2004.

[15] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, "Superfamilies of evolved and designed networks," *Science*, vol. 303, no. 5663, pp. 1538–1542, 2004. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1089167

[16] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proceedings of the 23rd KDD*. ACM, 2017, p. 555–564.

[17] R. A. Rossi, N. K. Ahmed, E. Koh, S. Kim, A. Rao, and Y. A. Yadkori, "Hone: Higher-order network embeddings," 2018.

[18] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005.

[19] D. DeWoskin, J. Climent, I. Cruz-White, M. Vázquez, C. C. Park, and J. Arsuaga, "Applications of computational homology to the analysis of treatment response in breast cancer patients," *Topology and its Applications*, vol. 157, pp. 157–164, 2010.

[20] A. Ghosh, B. Rozemberczki, S. Ramamoorthy, and R. Sarkar, "Topological signatures for fast mobility analysis," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 159–168.

[21] O. T. Courtney and G. Bianconi, "Generalized network structures: The configuration model and the canonical ensemble of simplicial complexes," *Physical Review E*, vol. 93, no. 6, jun 2016.

[22] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie, "Random walks on simplicial complexes and the normalized hodge laplacian," *CoRR*, vol. abs/1807.05044, 2018. [Online]. Available: http://arxiv.org/abs/1807.05044

[23] S. G. Aksoy, C. A. Joslyn, C. O. Marrero, B. Praggastis, and E. Purvine, "Hypernetwork science via high-order hypergraph walks," *EPJ Data Sci.*, vol. 9, no. 1, p. 16, 2020.

[24] A. R. Benson, "Three hypergraph eigenvector centralities," *CoRR*, vol. abs/1807.09644, 2018. [Online]. Available: http://arxiv.org/abs/1807.09644

[25] A. R. Benson, D. F. Gleich, and J. Leskovec, "Tensor spectral clustering for partitioning higher-order network structures," *CoRR*, vol. abs/1502.05058, 2015.

[26] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, "Hypergcn: A new method of training graph convolutional networks on hypergraphs," 2019.

[27] X. Li, W. Hu, C. Shen, A. Dick, and Z. Zhang, "Context-aware hypergraph construction for robust spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2588–2597, 2014.

[28] P. S. Chodrow, "Configuration models of random hypergraphs," *Journal of Complex Networks*, vol. 8, no. 3, 08 2020, cnaa018. [Online]. Available: https://doi.org/10.1093/comnet/cnaa018

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

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

[31] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," in *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, 1998, pp. 161–172.

[32] H. Tian and R. Zafarani, "Exploiting common neighbor graph for link prediction," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3333–3336.

[33] G. AbuOda, G. D. F. Morales, and A. Aboulnaga, "Link prediction via higher-order motif features," *CoRR*, vol. abs/1902.06679, 2019. [Online]. Available: http://arxiv.org/abs/1902.06679

[34] A. R. Benson, R. Abebe, M. T. Schaub, A. Jadbabaie, and J. M. Kleinberg, "Simplicial closure and higher-order link prediction," *CoRR*, vol. abs/1802.06916, 2018.

[35] N. Yadati, V. Nitin, M. Nimishakavi, P. Yadav, A. Louis, and P. Talukdar, "Nhp: Neural hypergraph link prediction," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1705–1714.

[36] M. Karoński and T. Łuczak, "The phase transition in a random hypergraph," *Journal of Computational and Applied Mathematics*, vol. 142, no. 1, pp. 125–135, 2002, probabilistic Methods in Combinatorics and Combinatorial Optimization.

[37] "Corona virus (covid-19) tweet metadata compilation 2020," www.trackmyhashtag.com/data/COVID-19.zip, accessed: 2020-12-01.

[38] "Twitter internet research agency dataset," https://archive.org/details/twitter-ira, accessed: 2020-12-01.

[39] Jaccard, "The distribution of the flora of the alpine zone," in *New Phytologist*, vol. 11, 1912, pp. 37–50.

[40] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Am. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, p. 1019–1031, May 2007.

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