Prediction Is NOT Classification: On Formulation and Evaluation of Hyperedge Prediction

Taehyung Yu, Soo Yong Lee, Hyunjin Hwang, Kijung Shin
Kim Jaechul Graduate School of AI, KAIST, Seoul, Republic of Korea
{taehyung.yu, syleetolow, hyunjinhwang, kijungs}@kaist.ac.kr

***Abstract*—A hypergraph, which consists of nodes and hyperedges (i.e., subsets of nodes), naturally represents group relations, such as recipes consisting of ingredients, outfits consisting of fashion items, and collaborations among researchers. The hyperedge prediction (HP) problem, which involves predicting future or missing hyperedges, has gained attention for applications, including recipe development, outfit recommendation, and collaborator search. However, due to the vast number of hyperedge candidates, which is about 2^n for n nodes, it is extremely challenging to identify the most promising ones among the entire candidate set. Thus, the problem is commonly reformulated as the classification of the real hyperedges and artificially generated ones in order to simplify both training and evaluation.**

Our work offers three significant contributions regarding HP. First, we present an improved formulation that is semantically aligned, computationally feasible, and better suited for various applications. Second, we make striking observations based on this improved formulation: (a) the performance in the classification formulation does not accurately reflect HP performance and is often negatively correlated, and (b) simple rule-based methods outperform advanced deep-learning approaches. Lastly, we present MHP, a novel HP method that utilizes masking-based training and outperforms all competing HP methods by up to 40%.

***Index Terms*—Hypergraph, Hyperedge Prediction, Hyperedge Recommendation, Hypergraph Neural Network**

I. Introduction

A hypergraph consists of nodes and hyperedges, and it naturally models group relations where each hyperedge indicates a set of nodes that interact as a group [1]–[3]. For instance, a hyperedge can depict a set of ingredients (or compounds) that interact together to compose a recipe (or drug) [4], as illustrated in Fig. 1. Similarly, a hyperedge can represent a team of individuals working together [1] and an outfit consisting of fashion items worn together [5].

Hyperedge prediction (HP) refers to a broad class of tasks related to predicting unobserved (future or missing) group interactions (i.e., hyperedges) in a given hypergraph [4], [6]–[9]. HP holds natural and significant potential for various applications, such as suggesting ingredients for recipes (or compounds for drug discovery) [4], [10], identifying potential research collaborators [4], [11], and recommending sets of fashion items to be purchased together [5], [12].

However, the formulation and evaluation of HP present unique and significant challenges due to the exponentially large (spec., about 2^n for n nodes) population of hyperedge candidates [13] in comparison to the real hyperedges. That is, due to the nature of hyperedges connecting any number of nodes, the observed, true hyperedge population (i.e., real hyperedges) is typically too small compared to the unobserved, candidate hyperedge population (i.e., non-hyperedge sets of nodes). Thus, there is an extreme class imbalance between the real hyperedges and non-hyperedge sets of nodes. The performance evaluation can also be computationally unwieldy, since the predictor may need to compute the likelihood for all hyperedge candidates.

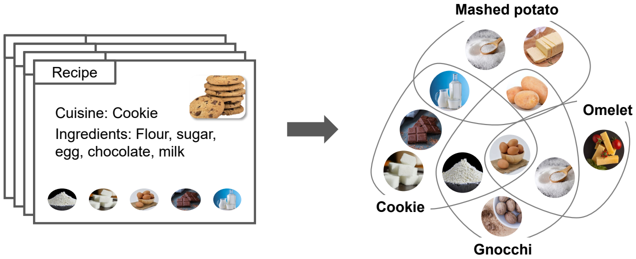

Fig. 1. An example of hypergraph modeling. The hypergraph on the right models the recipes on the left. Each node represents an ingredient (e.g., flour), and each hyperedge represents a cuisine (e.g., cookie). Each hyperedge contains the nodes corresponding to the ingredients of the corresponding cuisine.

To detour such challenges, many prior works *reformulated* HP into a classification problem [4], [6], [13]–[15]. In their problem formulation, they aim to classify positive hyperedges (i.e., the real ones) and negative hyperedges (i.e., non-hyperedge sets of nodes). A tiny subset of negative hyperedges is artificially generated by heuristic rules for model training and evaluation. This formulation simply assumes that classifying positive hyperedges and a tiny sample of negative hyperedges generalizes to predicting the entire population of negative hyperedges. However, such formulation poses significant issues, as discussed below.

In this work, we propose an improved formulation of HP that is semantically aligned, computationally feasible, and readily applicable. Given (a) a query set of nodes and (b) a target size of hyperedges, its objective is to identify unobserved hyperedges of the specified size that include the query set. Our formulation eliminates the need for sampling negative hyperedges and avoids reliance on heuristic generation rules. Instead, it leverages the query set and target size to naturally narrow down the candidate space to a feasible level. Moreover, HP with our improved formulation is more suitable for various applications, such as recipe development, outfit recommendation, and collaborator search.

Based on our improved prediction formulation, we have made significant observations regarding HP. First, the performance in the classification formulation does not accurately reflect prediction performance, and in many cases, it is even negatively correlated. This highlights the limitations of using classification as a proxy for evaluating HP methods. Second, simple approaches based on pairwise node similarities, such as Adamic-Adar [16] and Katz [17] indices, outperform all the state-of-the-art learning-based HP methods [4], [6], [13], highlighting the previously overlooked limitations of existing learning-based methodologies.

Motivated by the limitations, we propose **M**asking-based training for **H**yperedge **P**rediction (MHP), a novel deep-learning-based approach for HP. It leverages a masking and filling process as a proxy task for HP. Based on our improved formulation, we evaluate various HP methods on 6 real-world hypergraphs, and MHP performs the best in most cases. We summarize our contributions as follows:

- **Formulation**: We propose an improved formulation of HP that is semantically aligned, computationally feasible, and better-applicable.
- **Observations**: We report striking observations suggesting that prior works have used a misleading formulation of HP.
- **Algorithm**: We propose MHP, a novel HP method that outperforms the second-best method by up to 40%.

For **reproducibility**, we make our code and datasets publicly available at https://github.com/yu1012/Hyperedge_Prediction.

II. Related Work: Algorithms for HP

In this section, we present related work, focusing on algorithms for hyperedge prediction (HP). Refer to [18] for an extensive survey on this topic.

For HP, node similarity measures (e.g., Adamic-Adar index [16]), which have been widely used for edge prediction in ordinary graphs, have been extended to hypergraphs [14], [19].

In some studies on HP, the input hypergraph is transformed into an ordinary graph using clique expansion, where each hyperedge is replaced by a clique of its constituent nodes. Various techniques, such as non-negative matrix factorization and least square matching [19], are then employed for HP. For example, [4] apply graph neural networks to obtain node embeddings and subsequently aggregate them to generate hyperedge embeddings, which serve as the input for HP.

Alternatively, neural networks specialized for hypergraph structure or even for HP have been developed [6], [13], [20], [21], tackling HP-specific challenges (refer to a survey on hypergraph neural networks [22]). Notably, [13] employ GANs to generate negative hyperedges for training, while [6] focus on learning hyperedge-specific embeddings and encouraging similarity among general node embeddings within each hyperedge.

However, the learning-based approaches mentioned above rely on negative sampling or equivalent techniques for training and/or evaluation, while classification based on them is not a proper proxy for the HP problem, as discussed in Sect. V.

TABLE I
Frequently-used symbols and their definitions.

Symbol	Definition
$H = (V, E, X)$	input hypergraph
$V = \{v_1, \cdots, v_{\lvert V\rvert}\}$	set of nodes in H
$E = \{e_1, \cdots, e_{\lvert E\rvert}\}$	set of hyperedges H
$X \in \mathbb{R}^{\lvert V\rvert \times f}$	features of nodes in V
$I \in \{0,1\}^{\lvert V\rvert \times \lvert E\rvert}$	incidence matrix of H
$E_T \subseteq 2^V \setminus E$	target set, i.e., unobserved hyperedges in E
$Q \subset V$	a query node set
s	target size of each answer
k	target number of answers
d	embedding dimension of nodes & hyperedges
S	sample set (i.e., masked hyperedges) for training

III. Preliminaries

In this section, we present preliminary concepts. See Table I for a symbol table.

A. Basic Concepts

Hypergraph. A *hypergraph* $H = (V, E, X)$ consists of a node set $V = \{v_1, \cdots, v_{|V|}\}$, a hyperedge set $E = \{e_1, \cdots, e_{|E|}\}$, and node features $X \in \mathbb{R}^{|V| \times f}$. Each *hyperedge* $e_i \in E$ is a subset of nodes, i.e., $e_i \subseteq V$. In the *incidence matrix* $I \in \{0,1\}^{|V| \times |E|}$ of H, $I_{ij} = 1$ if and only if $v_i \in e_j$, for all i and $j \in \{1, \cdots, |V|\}$.

Hyperedge Prediction (HP). Given a hypergraph $H = (V, E, X)$, *hyperedge prediction* (HP) aims to predict the target set $E_T \subseteq 2^V \setminus E$. Typically, the *target set* E_T represents the set of missing hyperedges in H and/or hyperedges that will be added to H in the near future.

Hypergraph Neural Networks. Hypergraph neural networks (HNNs) are a family of neural networks designed for representation learning on hypergraphs [23]–[25] (refer to a survey on HNNs [22]). HNNs share a common overall structure. Given an incidence matrix I and initial node features $X_V^{(0)} = X$, each l-th HNN layer is composed of node-to-hyperedge (Eq. (1)) and hyperedge-to-node (Eq. (2)) propagation steps:

$$X_E^{(l)} = \sigma(I^T \cdot X_V^{(l)} \cdot W_E^{(l)} + b_E^{(l)}), \tag{1}$$

$$X_V^{(l+1)} = \sigma(I^T \cdot X_E^{(l)} \cdot W_V^{(l)} + b_V^{(l)}), \tag{2}$$

where $W_E^{(l)}, W_V^{(l)} \in \mathbb{R}^{d \times d}$ (exceptionally, $W_E^{(0)} \in \mathbb{R}^{f \times d}$) are weight matrices, $b_E^{(l)}, b_V^{(l)} \in \mathbb{R}^d$ are bias vectors, and σ is nonlinear activation. The node embeddings from the last layer, i.e., $X_V^{(L)} \in \mathbb{R}^{|V| \times d}$, serve as output representations.

B. Classification Formulation of HP

In most real-world hypergraphs, directly inferring the target set E_T poses significant challenges due to the vast number of *hyperedge candidates* that can potentially be included in E_T. Since the number grows with a speed of $O(2^{|V|})$, it is infeasible to score or rank all candidates. To detour such challenges, HP is commonly reformulated as a classification task [4], [6], [13]–[15], as follows:[1]

[1] Despite the challenges, some studies [7], [26] adhered to the original HP formulation. However, the search space they can consider is extremely limited compared to the entire set of candidates.

Problem 1 (Classification Formulation of HP):

- **Given**: (a) a hypergraph $H = (V, E, X)$ and (b) the union $C = E_T \cup E_N$ of the target set E_T of *positive hyperedges* (i.e., the real ones) and the set $E_N \subseteq 2^V \setminus E \setminus E_T$ of *negative hyperedges* (i.e., non-hyperedges sets of nodes),
- **to classify accurately**: whether each candidate $c \in C$ is positive (i.e., belongs to the target set E_T) or negative.

Typically, the set E_N of negative hyperedges is a tiny subset of the entire set of potential hyperedges (i.e., $2^V \setminus E \setminus E_T$), and the subset is artificially generated relying on heuristic rules [4], [6], [13], [14], which include the following strategies [15]:

- **Sized negative sampling (SNS)**: fill each hyperedge with nodes drawn uniformly at random.
- **Motif negative sampling (MNS)**: grow each hyperedge by repeatedly adding adjacent nodes.
- **Clique negative sampling (CNS)**: pick a random hyperedge and replace a random constituent node with a random node that is adjacent to all other constituent nodes.

These negative sampling techniques are designed to follow the size distribution of positive hyperedges. It is important to note that this formulation *assumes* that the classification of positive hyperedges and a tiny sample of negative hyperedges generalizes to the prediction over the entire population of negative hyperedges. However, empirical evidence presented in Sect. V challenges this assumption.

IV. Proposed Formulation of HP

In this section, we propose an improved formulation of hyperedge prediction (HP) that is semantically aligned, computationally feasible, and readily applicable.

Formulation. Our prediction formulation of HP (also see Fig. 2) is as follows:

Problem 2 (Proposed Prediction Formulation of HP):

- **Given**: (a) a hypergraph $H = (V, E, X)$, (b) a query node set $Q \subset V$, (c) the target size s of hyperedges, and (d) the target number k of answers,
- **to find**: up to k hyperedges $e'_1, \cdots, e'_k$ in the target set E_T,
- **subject to**: $Q \subset e'_i$ and $|e'_i| = s, \forall i \in \{1, \cdots, k\}$.

Advantages. Our formulation has advantages over the classification formulation (Problem 1) for the following reasons:

- **Semantically aligned**: Our formulation retains its nature as a prediction task of the original (yet impractical) definition of HP in Sect. III-A, where the goal is to "predict" rather than "classify" hyperedges.
- **Computationally feasible**: Our formulation reduces the space of candidates to a manageable size by incorporating the query set and target size as additional input parameters. Our formulation eliminates the need for sampling negative hyperedges and, thus, the reliance on heuristic generation rules for negative sampling.
- **Readily applicable**: Moreover, any HP method based on this new formulation can be readily used for a wide range of HP applications, including:
  1) **Collaborator search**: Given current members (query set), find new collaborators for them.
  2) **Outfit recommendation**: Given current fashion items (query set), find new items that pair well with them.
  3) **Recipe development**: Given current ingredients (query set), find new ingredients that can be balanced with them.

  Note that, in the above applications, the target size s is naturally interpreted as budget constraints.

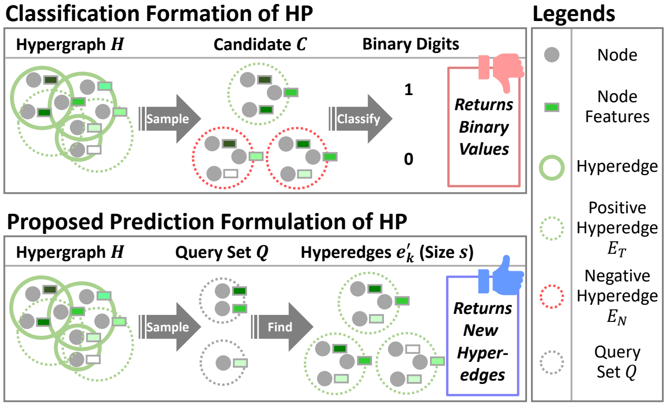

Fig. 2. The classification formulation and our proposed prediction formulation of HP. Refer to Sect. IV for the superiority of our formulation.

V. Experiments and Observations

In this section, we present empirical observations highlighting the limitations of the classification formulation of hyperedge prediction (HP). The observations are based on the performances of several HP methods under the classification formulation (Problem 1) and our improved formulation (Problem 2).

Baselines. For HP algorithms, we use AHP [13], NHP [4], and Hyper-SAGNN [6], which are the state-of-the-art deep-learning-based methods (see Sect. II). For detailed hyperparameter settings, refer to Appendix C. In addition, we use four heuristic methods that simply rank candidates based on the average pairwise similarity between constituent nodes: Common Neighbors (CN), Jaccard Index (JI), Adamic-Adar Index (AA) [16], and Katz Index [17] (refer to Appendix B for formulae). Since these heuristic methods are defined on a graph, in order to utilize them for HP, we transform the hypergraph into an ordinary graph. Specifically, we clique expand the input hypergraph by replacing each hyperedge with the clique of the nodes contained in the hyperedge.

Datasets. In Table II, we summarize statistics of the six real-world hypergraph datasets we used, which are the most widely used benchmarks for deep learning-based HP [4], [6], [13]. Their details can be found in Appendix A. In each dataset, we use a split ratio of 6:2:2 for train, validation, and test hyperedges, respectively. Since heuristic baselines (i.e., CN, JI, AA, and Katz Index) do not need training, their rules are computed with both train and validation hyperedges. We use five different splits and report the mean performance.

Evaluation Protocols. We generally follow the experimental protocols in [13], including hyperparameter tuning. As the classification performance measure, we use AUROC, considering each of the three negative sampling methods (SNS, MNS, and

TABLE II
DATA STATISTICS. ALL DATASETS ARE PUBLICLY AVAILABLE; SEE APPENDIX A.

	Category	# Nodes	# Hyperedges	#Features	Avg. Hyperedge Size	Max. Hyperedge Size
Cora-A	Co-authorship	2,388	970	1,433	4.48	43
Citeseer	Co-citation	1,458	1,004	3,703	3.25	26
Cora	Co-citation	1,434	1,483	1,433	3.06	5
Pubmed	Co-citation	3,840	7,531	500	4.48	171
iAF1260b	Metabolic Reaction	1,668	2,047	26	4.30	67
iJO1366	Metabolic Reaction	1,805	2,216	26	4.38	106

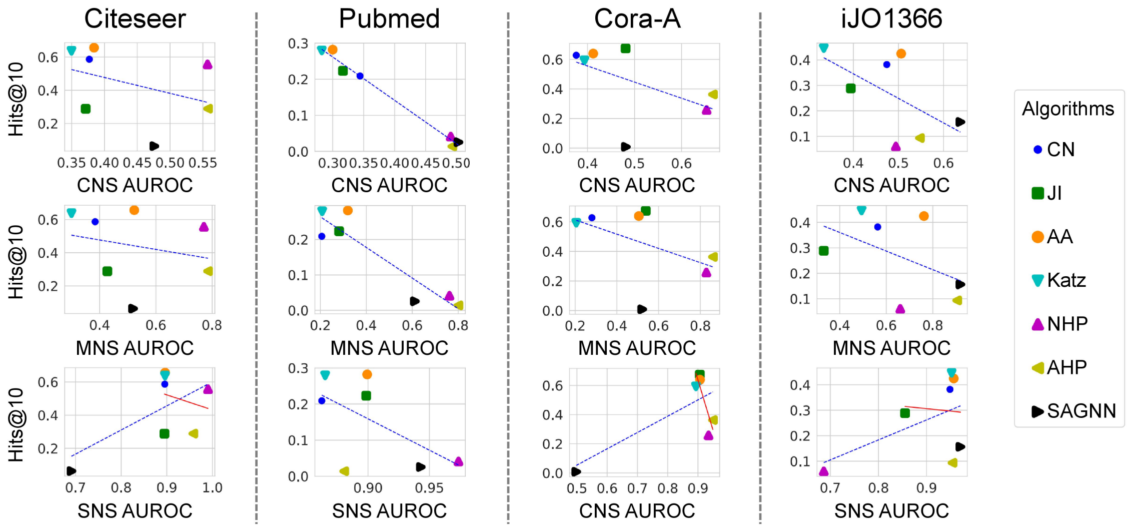

Fig. 3. **"Prediction is NOT classification."** The relationship between classification performances (on the X-axis) and prediction performances (on the Y-axis) in HP. The correlations (depicted by blue dotted lines) are generally weak and mostly even negative. Moreover, all positive correlations are weak, and removing the single worst-performing method leads to negative correlations (depicted by red solid lines).

CNS) described in Sect. III-B. As the prediction performance measure (i.e., performance in our prediction formulation), we use Hits@10 (i.e., the ratio of answer nodes that appear on the top-10 candidates provided), one of the most widely used metrics for recommender systems. Specifically, we (a) use query sets Q obtained by removing one random node from the hyperedges in the target set E_T, (b) set the target size to $s = |Q| + 1$, and (c) set the target number k of answers to 10. For each method, we rank all candidates based on their outputs (average node similarity, softmax values, etc.). For the final top-10 candidates, we discard any of the predictions that include *existing* train or validation hyperedges. Predicting the same hyperedges as those used in the train and validation sets does not align with our hyperedge prediction formulation, and we consider this neither a correct nor an incorrect prediction.

Results and Observations. Fig. 3 shows the (a) classification performances (on the X-axis) and (b) prediction performances (i.e., performances in our improved prediction formulation of HP) of the evaluated HP algorithms (on the Y-axis) on four hypergraph datasets. Similar trends are observed also in the other datasets.

***Observation** 1: The correlations between classification performance and prediction performance are generally weak and, in many cases, even negative.*

Even in cases where a positive correlation is observed, the strength of the correlation is weak, and removing the single worst-performing method leads to a negative correlation. This finding highlights that classification performance is not an effective proxy for evaluating performances in HP.

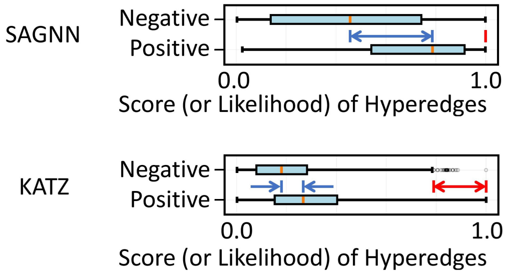

Fig. 4. The score distributions from SAGNN and Katz, which are the representative baselines for the classification and prediction formulations, respectively, in the iJO1366 dataset. While SAGNN (the classification formulation) increases the gap between average scores of positive and negative hyperedges (blue arrows), the gap in their top scores (red arrows) remains small. This accounts for satisfactory classification performance but unsatisfactory prediction performance. The opposite results are observed for Katz.

***Observation** 2: The prediction performances of deep-learning-based methods are consistently and significantly lower than the simple heuristic methods.*

This observation highlights the previously overlooked limitations of recent methodologies, which are based on the classification formulation. For a further analysis regarding the reasons behind these observations, please refer to Fig. 4.

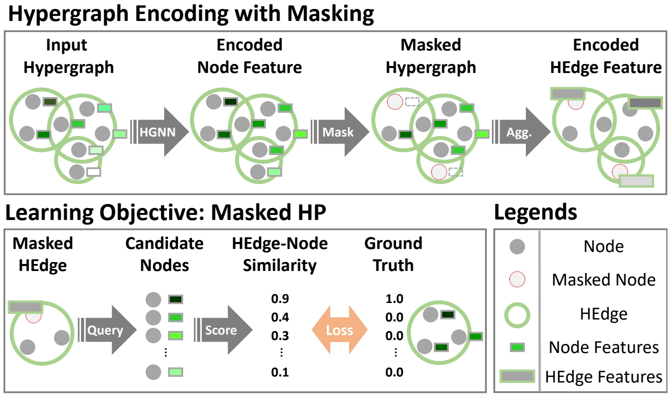

Fig. 5. The architecture and learning objective of MHP. HP stands for hyperedge prediction, and HEdge stands for hyperedge.

VI. Proposed Method: MHP

In this section, we propose **M**asking-based Training for **H**yperedge **P**rediction (MHP), a novel approach for hyperedge prediction (HP). We first present its training method and then its structure. Lastly, we offer an empirical evaluation.

A. Training Objective: Masking and Filling (Fig. 5)

Observations 1 and 2 reveal that the classification task defined in Problem 1 is an inadequate proxy for HP. This highlights the need for a new proxy task for training models (e.g., neural networks) effectively. We suggest training models by a random masking and filling process, inspired by NLP-model training [27].

Given a hypergraph $H = (V, E, X)$, we generate samples $S = \{s_1, \cdots, s_{|S|}\}$, where each sample $s_i = (Q_i, \hat{V}_i)$ comprises (1) a subset Q_i obtained by masking (i.e., removing) a member of a hyperedge and (2) the set of all answer nodes $\hat{V}_i$, i.e., $\forall v \in \hat{V}_i$, $Q_i \cup \{v\} \in E$. We train a model to predict the answer nodes $\hat{V}_i$ from Q_i for each sample $s_i = (Q_i, \hat{V}_i)$. Specifically, a model outputs a probability distribution p_{Q_i} over $V \setminus Q_i$ from Q_i, and we aim to minimize Eq. (3).

$$\mathcal{L} = \frac{1}{|S|} \sum\nolimits_{(Q_i, \hat{V}_i) \in S} \left(\frac{1}{|\hat{V}_i|} \sum\nolimits_{v_j \in \hat{V}_i} -\log p_{Q_i}(v_j) \right). \tag{3}$$

Note that this process directly mimics the simplest case of Problem 2, where the target size s is equal to $|Q| + 1$ (i.e., only one node is added to the query set Q).

B. Structure of MHP (Fig. 5)

Feature-based Embedding. MHP adapts HGNN [23], a simple HNN architecture. As described in Sect. III (see Eq. (1) and Eq. (2)), HGNN employs an alternating message-passing approach, passing messages from nodes to hyperedges and vice versa to obtain node embeddings $X_V^{(L)} \in \mathbb{R}^{|V| \times d}$.

Structure-based Embedding. To further utilize structural information, we compute additional structural node embeddings by multiplying the (normalized) adjacency matrix[2] with a learnable diagonal weight matrix $W_s \in \mathbb{R}^{|V| \times |V|}$. The final node embeddings $Z \in \mathbb{R}^{|V| \times (d + |V|)}$ are obtained by concatenating the feature-based and structure-based embeddings as follows:

$$Z = [X_V^{(L)} \parallel W_s \cdot \hat{A}].$$

Pooling and Scoring. Based on the node embeddings, we compute the embedding of each query (or masked hyperedge) Q by aggregating the embeddings of its constituent nodes. That is, we compute the embedding Y_Q of Q as follows:

$$Y_Q = aggregate(\{Z_j : v_j \in Q\}),$$

where Z_j is the j-th row of the final node embedding matrix Z. For aggregation, we employ simple mean, while a variety of aggregation functions (e.g., maxmin [4]) can be used as alternatives. Then, in order to compute the score $p_Q(v_{j'})$ of each candidate node $v_{j'} \in V \setminus Q$ for forming a hyperedge with Q (or filling the masked hyperedge Q), we compute the inner product of the embeddings of $v_{j'}$ and Q as follows:

$$p_Q(v_{j'}) = \frac{\exp(Z_{j'}^T \cdot Y_Q)}{\sum_{v_i \in V \setminus Q} \exp(Z_i^T \cdot Y_Q)}. \tag{4}$$

This score is used in Eq. (3), where $Q = Q_i$, for training.

Top-k Selection. For a single answer (i.e., if $k = 1$), MHP returns the hyperedge where the top-$(s - |Q|)$ nodes with the highest scores are added to Q. For more answers, MHP performs a beam search, i.e., it grows the size of each (partial) answer one by one, while retaining top-k best partial answers as candidates. We use Eq. (4) with Q substituted by a partial answer to choose nodes to be added.

The time complexity of the beam search is $O(k|V|(s-|Q|))$. To achieve this, we maintain the node embeddings of size $O(|V| \cdot d)$ from HGNN and reuse them in the other steps. For beam search, we select the top-k nodes among $O(|V|)$ nodes in each of $s - |Q|$ steps. Thus, the running time of the beam search is $O(k|V|(s - |Q|))$.

C. Empirical Evaluation of MHP

For evaluation of MHP, we use the same datasets, splits, baseline methods, hyperparameter settings used in Sect. V. We conduct a grid search for the hyperparameters of MHP. Its details can be found in Appendix D.

Evaluation Protocols. We consider three different evaluation scenarios, such that for each target hyperedge $e \in E_T$,

- **S1. Find One**: create a query $Q \subset e$ of size $|e| - 1$; set the target size s to $|e|$.
- **S2. Find Two**: create a query $Q \subset e$ of size $|e| - 2$; set the target size s to $|e|$.
- **S3. Find Half**: create a query $Q \subset e$ of size $\lceil \frac{|e|}{2} \rceil$; set the target size s to $|e|$.

For each scenario, each query node set Q is constructed by removing random nodes from each target set hyperedge. A model, then, finds nodes from the hypergraph for the query node set Q. If the found nodes are in the target set, the model is considered to have made an accurate prediction. Consequently, for each query node set Q, multiple valid answers may exist.

[2] $\hat{A} = D_v^{-1/2} I D_e^{-1/2} I^T D_v^{-1/2}$. Here, I is the incident matrix. $D_v \in \mathbb{R}^{|V| \times |V|}$ and $D_e \in \mathbb{R}^{|E| \times |E|}$ denote the degree matrices of nodes and hyperedges, respectively.

TABLE III
HYPEREDGE PREDICTION PERFORMANCE UNDER SCENARIO S1 (FIND *one* NODE PER QUERY NODE SET Q). VALUES ARE RESCALED TO 100.

Datasets	Metric	CN	JI	AA	Katz	SAGNN	NHP	AHP	MHP
Cora-A	Hits@10	62.78±2.2	67.42±1.4	63.92±2.3	59.38±1.6	0.82±0.7	25.77±6.1	36.29±2.4	**71.65±1.5**
	MRR@10	32.40±2.5	37.57±1.4	32.27±2.7	26.15±2.3	0.31±0.3	9.50±3.4	30.01±1.9	**40.32±3.2**
Citeseer	Hits@10	58.66±3.6	56.71±1.1	65.61±4.3	63.78±4.2	6.34±2.1	55.61±4.2	28.90±6.1	**66.95±1.9**
	MRR@10	34.50±2.8	28.84±1.8	37.05±2.5	36.80±2.7	2.91±1.9	28.15±2.8	14.14±6.0	**41.20±3.1**
Pubmed	Hits@10	20.90±1.1	22.32±0.9	28.20±1.7	27.96±1.1	2.55±0.6	4.10±0.5	1.39±2.2	**36.23±2.0**
	MRR@10	9.69±0.7	9.79±0.3	13.38±0.7	12.16±0.7	0.75±0.1	1.32±0.2	0.46±0.8	**18.09±0.6**
Cora	Hits@10	59.66±1.1	51.21±2.1	59.06±2.1	13.59±0.6	13.96±6.5	33.83±4.1	39.87±3.7	**60.07±2.1**
	MRR@10	34.59±1.0	25.67±1.1	**34.81±1.3**	5.82±0.2	6.52±3.5	14.90±2.7	22.57±2.4	34.74±1.3
iAF1260b	Hits@10	37.42±3.6	32.21±3.4	42.48±2.7	18.47±0.7	17.52±2.2	7.79±0.7	3.02±3.4	**50.75±2.9**
	MRR@10	23.87±3.2	21.73±1.7	25.99±2.8	8.17±0.7	5.59±1.7	2.99±0.6	0.96±1.1	**30.40±1.2**
iJO1366	Hits@10	38.20±1.2	28.85±2.1	42.47±2.0	42.95±17.5	15.64±1.9	6.02±1.2	9.35±2.2	**52.94±1.7**
	MRR@10	25.04±1.1	19.27±1.0	27.25±1.4	25.37±11.2	4.32±0.4	2.22±0.4	2.96±0.8	**30.91±1.0**

TABLE IV
HYPEREDGE PREDICTION PERFORMANCE UNDER SCENARIO S2 (FIND *two* NODES PER QUERY NODE SET Q). VALUES ARE RESCALED TO 100.

Datasets	Metric	CN	JI	AA	Katz	SAGNN	NHP	AHP	MHP
Cora-A	Hits@10	46.91±1.5	50.72±2.2	47.94±0.6	43.09±1.2	0.41±0.2	14.64±3.8	31.44±2.9	**53.51±1.9**
	MRR@10	23.02±2.1	28.22±2.3	23.26±1.2	19.09±0.5	0.29±0.2	7.01±2.6	27.39±3.7	**29.99±1.8**
Citeseer	Hits@10	48.66±3.2	42.07±1.5	53.90±1.3	51.10±2.7	3.54±2.3	35.12±5.6	16.95±6.7	**55.12±3.1**
	MRR@10	26.58±3.5	20.56±1.5	28.68±2.9	28.14±3.4	1.83±1.5	21.23±4.0	9.80±4.8	**32.47±2.1**
Pubmed	Hits@10	12.74±0.7	11.79±1.0	15.12±1.0	13.59±0.6	1.78±0.5	1.66±0.4	0.53±0.8	**20.31±0.9**
	MRR@10	5.90±0.6	5.01±0.5	7.36±0.4	5.82±0.2	0.53±0.1	0.79±0.1	0.24±0.4	**10.32±0.6**
Cora	Hits@10	**41.81±3.2**	31.34±2.6	40.20±3.3	35.96±13.6	5.17±2.1	15.10±4.9	21.48±3.0	39.93±2.6
	MRR@10	22.91±1.9	16.53±0.9	22.80±1.9	21.69±8.1	3.69±1.8	8.25±2.8	13.66±2.7	**23.67±1.8**
iAF1260b	Hits@10	12.21±1.3	10.02±1.4	12.02±0.6	10.95±1.4	0.78±0.7	1.12±0.2	0.15±0.3	**13.58±2.0**
	MRR@10	7.56±0.8	6.62±1.1	7.59±0.6	6.83±0.9	0.30±0.1	0.67±0.2	0.12±0.3	**8.48±0.7**
iJO1366	Hits@10	13.66±1.4	7.55±0.6	13.53±1.1	12.22±1.1	0.81±0.5	1.84±0.7	0.54±0.4	**14.47±1.3**
	MRR@10	8.9±1.0	5.33±0.8	8.56±0.9	7.91±1.1	0.40±0.2	1.07±0.5	0.15±0.1	**9.24±0.9**

We follow the protocols in Sect. V. To evaluate various aspects of HP, we use Hits@k and Mean Reciprocal Rank (MRR@k). We fix the target number $k = 10$. However, similar results are obtained regardless of k values (see Fig. 6).

Results. Tables III, IV, and V present the results for the aforementioned scenarios: S1 (Find One), S2 (Find Two), and S3 (Find Half). Remarkably, MHP outperforms (1) all other HP methods, (2) under each scenario, (3) in terms of each evaluation metric, and (4) across all but one dataset. Besides having an average rank near 1, MHP outperforms the baselines by a large margin. For example, its MRR@10 is 35.2%, 40.2%, and 39.7% higher than those of the second-best performance in the Pubmed dataset under Scenarios S1, S2, and S3, respectively. Note that deep-learning-based baselines are consistently and significantly outperformed not only by MHP but also by the simple heuristic methods. This outcome aligns with Observation 2 in Sect. V. The superiority of MHP is consistent across different target number k's (see Fig. 6).

Ablation Study. We present an ablation study using six variants of MHP. Four of them employ the classification formulation, wherein the MHP is trained to classify positive hyperedges from negative hyperedges generated by different strategies (spec., those described in Sect. III-B and their combination). As shown in Table VI, our proposed formulation plays a crucial role in enhancing performance, and the structure-based embedding (SE, refer to Sect. VI-B) also exhibits utility.

VII. CONCLUSION

To detour the challenges posed by the vast number of candidates, hyperedge prediction (HP) is commonly formulated as a classification problem. However, our findings present significant observations indicating that the classification performance of HP does not consistently generalize to overall HP performance and often exhibits a negative correlation. Motivated by the observations, we propose an improved formulation of HP that is semantically aligned, computationally feasible, and readily applicable. In addition, we introduce MHP, which leverages the masking and filling process as a proxy task for training neural networks for HP. Empirically, MHP consistently and significantly outperforms all competing methods. In conclusion, our study sheds light on the limitations overlooked in the many prior works on HP and offers a new direction to address them.

The generalization of our findings is limited in three ways. First, we did not demonstrate theoretical superiority of the proposed HP formulation. Second, we did not explore more

TABLE V
HYPEREDGE PREDICTION PERFORMANCE UNDER SCENARIO S3 (FIND *half* OF THE NODES PER QUERY NODE SET Q). VALUES ARE RESCALED TO 100.

Datasets	Metric	CN	JI	AA	Katz	SAGNN	AHP	NHP	MHP
Cora-A	Hits@10	53.81±2.9	56.49±2.1	54.74±2.6	49.90±2.5	0.52±0.4	14.95±3.5	34.12±2.2	**60.52±3.3**
	MRR@10	26.82±1.9	32.79±2.2	26.76±2.4	21.44±1.8	0.40±0.4	6.71±2.4	29.60±2.8	**34.57±2.4**
Citeseer	Hits@10	53.90±3.8	49.51±2.6	59.51±1.6	58.05±3.0	3.05±1.6	38.90±4.0	19.15±7.2	**61.22±1.9**
	MRR@10	30.52±3.9	25.38±2.0	32.85±3.0	31.54±2.7	1.72±1.1	25.31±3.7	11.42±5.3	**36.76±3.6**
Pubmed	Hits@10	16.51±0.5	15.08±1.0	20.30±0.6	18.47±0.7	2.25±0.5	1.94±0.3	0.59±1.0	**26.43±0.9**
	MRR@10	7.79±0.6	6.43±0.4	9.68±0.3	8.17±0.7	0.63±0.1	0.93±0.1	0.28±0.5	**13.52±0.5**
Cora	Hits@10	**50.67±2.9**	42.82±1.4	49.40±2.7	42.95±17.5	7.58±4.7	19.19±4.4	26.64±3.0	50.34±2.3
	MRR@10	28.44±2.0	21.41±1.1	28.85±0.7	25.37±11.2	4.96±3.4	10.96±3.3	18.26±1.8	**29.44±1.0**
iAF1260b	Hits@10	11.25±1.9	7.55±0.4	10.37±0.8	9.64±1.1	1.41±0.6	1.46±0.4	0.39±0.7	**12.32±1.2**
	MRR@10	5.96±0.6	4.30±0.7	5.97±0.5	5.19±0.6	0.52±0.2	0.82±0.3	0.29±0.6	**6.06±0.8**
iJO1366	Hits@10	11.98±1.1	6.08±0.2	12.34±0.6	10.50±1.2	0.95±0.6	1.94±0.7	0.72±0.5	**12.70±0.7**
	MRR@10	6.74±0.4	3.54±0.5	6.57±0.6	5.16±0.7	0.51±0.2	1.24±0.4	0.27±0.2	**6.78±0.5**

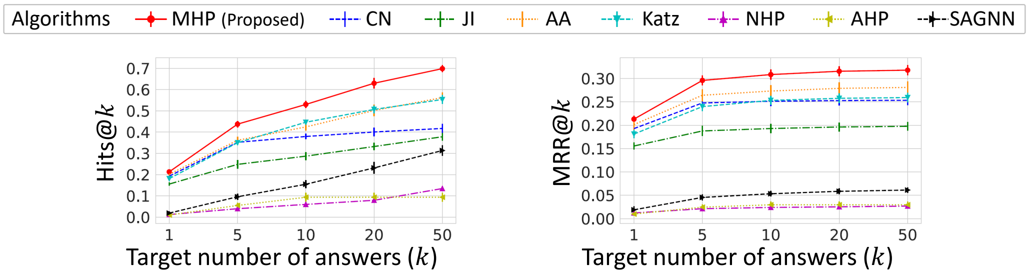

Fig. 6. MHP performs best regardless of k values at Hits@k and MRR@k. Scenario S1 (Find One) on the iJO1366 dataset is considered.

TABLE VI
ABLATION STUDY UNDER SCENARIO S1 (FIND *one* NODE PER QUERY NODE SET Q) IN THE IJO1366 DATASET. EACH NUMBER IS RESCALED TO A RANGE OF 100. SE STANDS FOR STRUCTURAL EMBEDDING. SIMILAR RESULTS ARE FOUND ACROSS ALL OTHER DATASETS.

Method	MHP + Classification Form.				MHP	MHP
	SNS	MNS	CNS	ALL	w/o SE	
Hits@10	22.00±5.2	14.80±2.5	0.50±0.5	14.00±0.5	35.10±2.1	**52.94±1.7**
MRR@10	12.30±3.8	5.80±1.1	0.10±0.2	6.60±1.3	18.80±2.0	**30.91±1.0**

practical applications of HP, such as group recommendation. Third, the used benchmark hypergraphs are small, and how MHP performs in larger hypergraphs remains unclear. Addressing the limitations would be some valuable research directions.

Acknowledgements. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00457882, AI Research Hub Project, 90%) (No. RS-2019-II190075 Artificial Intelligence Graduate School Program(KAIST), 10%).

REFERENCES

[1] A. R. Benson, R. Abebe, M. T. Schaub, A. Jadbabaie, and J. Kleinberg, "Simplicial closure and higher-order link prediction," *Proceedings of the National Academy of Sciences*, vol. 115, no. 48, pp. E11 221–E11 230, 2018.

[2] M. T. Do, S.-e. Yoon, B. Hooi, and K. Shin, "Structural patterns and generative models of real-world hypergraphs," in *KDD*, 2020.

[3] G. Lee, F. Bu, T. Eliassi-Rad, and K. Shin, "A survey on hypergraph mining: Patterns, tools, and generators," *arXiv preprint arXiv:2401.08878*, 2024.

[4] N. Yadati, V. Nitin, M. Nimishakavi, P. Yadav, A. Louis, and P. Talukdar, "Nhp: Neural hypergraph link prediction," in *CIKM*, 2020.

[5] Z. Li, J. Li, T. Wang, X. Gong, Y. Wei, and P. Luo, "Ocphn: Outfit compatibility prediction with hypergraph networks," *Mathematics*, vol. 10, no. 20, p. 3913, 2022.

[6] R. Zhang, Y. Zou, and J. Ma, "Hyper-sagnn: a self-attention based graph neural network for hypergraphs," in *ICLR*, 2019.

[7] A. Sharma, R. Kuang, J. Srivastava, X. Feng, and K. Singhal, "Predicting small group accretion in social networks: A topology based incremental approach," in *ASONAM*, 2015.

[8] A. Sharma, "Hypergraph analytics: modeling higher-order structures and probabilities," Ph.D. dissertation, University of Minnesota, 2020.

[9] M. Sales-Pardo, A. Mariné-Tena, and R. Guimerà, "Hyperedge prediction and the statistical mechanisms of higher-order and lower-order interactions in complex networks," *Proceedings of the National Academy of Sciences*, vol. 120, no. 50, p. e2303887120, 2023.

[10] Z. Zhang, Y. Feng, S. Ying, and Y. Gao, "Deep hypergraph structure learning," *arXiv preprint arXiv:2208.12547*, 2022.

[11] C. Yang, R. Wang, S. Yao, and T. Abdelzaher, "Semi-supervised hypergraph node classification on hypergraph line expansion," in *CIKM*, 2022.

[12] Y. Li, H. Chen, X. Sun, Z. Sun, L. Li, L. Cui, P. S. Yu, and G. Xu, "Hyperbolic hypergraphs for sequential recommendation," in *CIKM*, 2021.

[13] H. Hwang, S. Lee, C. Park, and K. Shin, "Ahp: Learning to negative sample for hyperedge prediction," in *SIGIR*, 2022.

[14] S.-e. Yoon, H. Song, K. Shin, and Y. Yi, "How much and when do we need higher-order information in hypergraphs? a case study on hyperedge prediction," in *WWW*, 2020.

[15] P. Patil, G. Sharma, and M. N. Murty, "Negative sampling for hyperlink prediction in networks," in *PAKDD*, 2020.

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

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

[18] C. Chen and Y.-Y. Liu, "A survey on hyperlink prediction," *arXiv preprint arXiv:2207.02911*, 2022.

[19] M. Zhang, Z. Cui, S. Jiang, and Y. Chen, "Beyond link prediction: Predicting hyperlinks in adjacency space," in *AAAI*, 2018.

[20] K. Tu, P. Cui, X. Wang, F. Wang, and W. Zhu, "Structural deep embedding for hyper-networks," in *AAAI*, 2018.

[21] C. Wan, M. Zhang, W. Hao, S. Cao, P. Li, and C. Zhang, "Principled hyperedge prediction with structural spectral features and neural networks," *arXiv preprint arXiv:2106.04292*, 2021.

[22] S. Kim, S. Y. Lee, Y. Gao, A. Antelmi, M. Polato, and K. Shin, "A survey on hypergraph neural networks: an in-depth and step-by-step guide," in *KDD*, 2024.

[23] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *AAAI*, 2019.

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

[25] Y. Dong, W. Sawin, and Y. Bengio, "Hnhn: Hypergraph networks with hyperedge neurons," *arXiv preprint arXiv:2006.12278*, 2020.

[26] T. Kumar, K. Darwin, S. Parthasarathy, and B. Ravindran, "Hpra: Hyperedge prediction using resource allocation," in *AAAI*, 2020.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL-HLT*, 2019.

[28] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[29] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015.

[30] G. Namata, B. London, L. Getoor, B. Huang, and U. Edu, "Query-driven active surveying for collective classification," in *MLG*, 2012.

[31] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis, "Bigg models: A platform for integrating, standardizing and sharing genome-scale models," *Nucleic acids research*, vol. 44, no. D1, pp. D515–D522, 2016.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

APPENDIX

A. Datasets

We adopt the data preprocessing method used in [13], [24], with removing all duplicate hyperedges.

- **Co-authorship datasets (Cora-A[3] [28]):** Each co-authorship dataset represents authors collaborating to write academic papers. Since a single author writes several papers, each paper becomes a node, and the papers of the same author form a hyperedge. To generate node features, we use the bag-of-words representation of the abstract for each paper.
- **Co-citation datasets (Citeseer[4] [29], Cora[5] [28], and Pubmed[6] [30])**: Each Co-citation dataset involves papers that reference each other through citations. Each node represents a paper, and each hyperedge corresponds to the collection of papers that have been cited by a paper. The node features are generated using the same bag-of-words representation as the co-authorship dataset.
- **Metabolic reaction datasets (iAF1260b[7] and iJO1366 [8] [31]):** A metabolic reaction dataset is established based on metabolic pathways, where nodes symbolize specific materials, and hyperedges represent reactions between these materials. The node features are created by counting the atoms (e.g. C, H, O) present within each material.

B. Hueristic Baselines

- **Common Neighbors (CN)**: Nodes sharing a greater *number* of neighbors are more likely to form a hyperedge, i.e.,
$$CN = |N(v_i) \cap N(v_j)|. \tag{5}$$
- **Jaccard Index (JI)**: Nodes sharing a greater *fraction* of neighbors are more likely to form a hyperedge, i.e.,
$$JI = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i)| + |N(v_j)|}. \tag{6}$$
- **Adamic-Adar (AA)**: The common neighbors with lower degrees are more important for HP, spec.,
$$AA = \sum_{z \in N(v_i) \cap N(v_j)} \frac{1}{\log |N(z)|} \tag{7}$$
- **Katz Index**: Nodes with a greater number of shorter paths between them are more likely to form a hyperedge, spec.,
$$KZ = \sum_{\ell=1}^{\infty} \beta^{\ell} |path(v_i, v_j) = \ell|. \tag{8}$$

For all, $N(v)$ stands for the neighbors of v, and $path(v_i, v_j)$ stands for the number of paths between v_i and v_j.

C. Detailed Hyperparameter Settings of Baselines

For the deep-learning-based methods, we conduct a grid search to determine the embedding dimension $d \in \{64, 128, 256\}$, learning rate $lr \in \{1e-3, 5e-4, 1e-4\}$, and the batch size $b \in \{64, 128, 256\}$. For AHP, which involves several other hyperparameters, we adopt the hyperparameter values reported as the best in the original paper. All the heuristic methods (CN, JI, AA, and Katz) are hyperparameter-free.

D. Detailed Hyperparameter Settings of MHP

The hyperparameter search space for MHP is in Table VII.

TABLE VII
HYPERPAMETER SEARCH SPACE OF MHP.

Hyperparameter	Search Space
Maximum number of epochs	500
Number of HGNN layers	$\{1, 2\}$
Hidden dimension (i.e., d)	$\{64, 128\}$
Number of samples (i.e., $\lvert S\rvert/\lvert E\rvert$)	$\{2, 5, 8, 10, 12, 15\}$
Optimizer	Adam [32]
Learning rate	$\{1e-3, 5e-4, 1e-4\}$
Batch size	$\{128, 256, 512\}$

[3] https://people.cs.umass.edu/~mccallum/data.html

[4] https://linqs.org/datasets/#citeseer-doc-classification

[5] https://linqs.org/datasets/#cora

[6] https://linqs.org/datasets/#pubmed-diabetes

[7] http://bigg.ucsd.edu/models/iAF1260b

[8] http://bigg.ucsd.edu/models/iJO1366