Co-clustering Interactions via Attentive Hypergraph Neural Network

Tianchi Yang
Beijing University of Posts and Telecommunications
China
yangtianchi@bupt.edu.cn

Cheng Yang
Beijing University of Posts and Telecommunications
China
albertyang33@gmail.com

Luhao Zhang
Meituan
China
zhangluhao@meituan.com

Chuan Shi*
Beijing University of Posts and Telecommunications
China
shichuan@bupt.edu.cn

Maodi Hu
Meituan
China
humaodi@meituan.com

Huaijun Liu
Meituan
China
liuhuaijun@meituan.com

Tao Li
Meituan
China
litao19@meituan.com

Dong Wang
Meituan
China
wangdong07@meituan.com

ABSTRACT

With the rapid growth of interaction data, many clustering methods have been proposed to discover interaction patterns as prior knowledge beneficial to downstream tasks. Considering that an interaction can be seen as an action occurring among multiple objects, most existing methods model the objects and their pair-wise relations as nodes and links in graphs. However, they only model and leverage part of the information in real entire interactions, i.e., either decompose the entire interaction into several pair-wise sub-interactions for simplification, or only focus on clustering some specific types of objects, which limits the performance and explainability of clustering. To tackle this issue, we propose to Co-cluster the Interactions via Attentive Hypergraph neural network (CIAH). Particularly, with more comprehensive modeling of interactions by hypergraph, we propose an attentive hypergraph neural network to encode the entire interactions, where an attention mechanism is utilized to select important attributes for explanations. Then, we introduce a salient method to guide the attention to be more consistent with real importance of attributes, namely saliency-based consistency. Moreover, we propose a novel co-clustering method to perform a joint clustering for the representations of interactions and the corresponding distributions of attribute selection, namely cluster-based consistency. Extensive experiments demonstrate that our CIAH significantly outperforms state-of-the-art clustering methods on both public datasets and real industrial datasets.

*Corresponding author.

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531868

CCS CONCEPTS

• **Theory of computation → Unsupervised learning and clustering**; • **Mathematics of computing → Hypergraphs**; • **Computing methodologies → Neural networks**.

KEYWORDS

Graph Neural Networks, Hypergraph, Clustering, Interaction Data

ACM Reference Format:
Tianchi Yang, Cheng Yang, Luhao Zhang, Chuan Shi, Maodi Hu, Huaijun Liu, Tao Li, and Dong Wang. 2022. Co-clustering Interactions via Attentive Hypergraph Neural Network. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531868

1 INTRODUCTION

In the age of rapid development of social media and surge of complex network, there are more and more interaction data in all walks of life such as user-item interaction network in recommendation, citation network in academia, etc. An interaction can be seen as an action occurring among multiple objects. Therefore, researchers usually model the objects and their pair-wise relations as nodes and links in graphs [4, 16, 22, 34].

As a fundamental data mining task, clustering on interaction data (i.e., clustering on graph) can reveal valuable cluster patterns as prior knowledge for downstream tasks or give some insights to the research and industry [24, 33, 34, 36]. Early studies usually only embed the structural information by graph embedding methods [7, 11] and then perform clustering over the objects. Then, considering the advantages of attributes, researchers explore to combine the strength of both structures and attributes for better

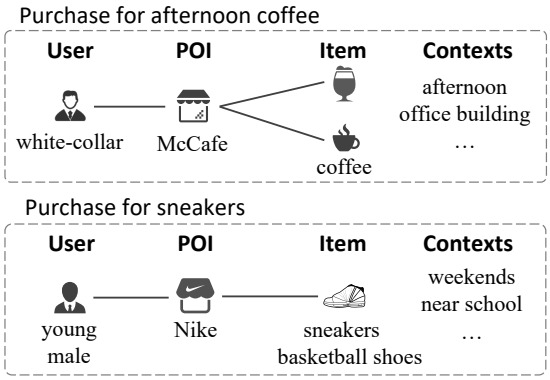


Figure 1: Examples of entire interaction in the domain of shopping.

clustering performance based on attributed graphs [4, 18, 38]. Recently, more and more researches further develop heterogeneous graph methods to more accurately model and encode interactions by further considering the different types of nodes or links [16, 22]. Apart from the performance of clustering, the explainability of clustering is also an important issue to help understand the reasons behind model decisions [23]. Specifically, most methods attempt to select important and concise attributes to explain the clustering results [10, 21, 23].

Although clustering on interaction data has been extensively developed, real interaction data is much more complicated. In real applications, an entire interaction usually contains multiple attributed interaction objects and interaction attributes such as temporal-spatial contexts. For example, as illustrated in Figure 1 in the domain of shopping, an entire interaction includes, but is not limited to, "who purchases what products in which store under what contexts". Moreover, each part of an entire interaction is significant and necessary for discovering the patterns. As shown at the top of Figure 1, white-collars often order coffee in the afternoon for efficient work. If we ignore this condition of temporal context "afternoon", one may conclude a one-sided pattern, thus probably leading to recommending coffee at midnight, which violates common sense. However, existing clustering methods only model and leverage part of the information in real interaction, i.e. either decompose the entire interaction into several pair-wise sub-interactions for simplification [22, 26], or only focus on clustering some specific types of the interaction objects rather than the entire interaction [5, 25]. Therefore, the existing clustering methods cannot comprehensively characterize and utilize the information in the entire interactions. Furthermore, selecting important attributes from the entire interactions will yield a more accurate explanation of the clustering results. In contrast, based on incomplete interaction modeling, it may miss some key information that is helpful for explanation, such as "afternoon" in the above example. So far, however, there have been few attempts to explore the rich attributes in entire interactions for clustering explanations.

In this paper, we make the *first* attempt to cluster the entire interactions, rather than simple interactions in traditional clustering approaches. It can further provide the clustering explanations by selecting key attributes from any part of the entire interactions. It is not a trivial task due to the following challenges: (1) How to effectively model and encode the entire interactions? Each entire interaction involves an uncertain number of attributed objects and interaction attributes as well as the relations among multiple objects. Hence, it is insufficient to model and encode such entire interactions by the aforementioned types of graphs and methods. (2) How to select explainable key attributes from entire interactions? It is a common solution for selecting attributes with attention mechanism. However it is somehow questioning in terms of explanations, since the attention weights are sometimes inconsistent with the real importance of attributes [17], especially when dealing with the rich attributes in entire interactions. (3) How to jointly improve the performance and explainability of clustering on entire interactions? As studied in previous work [17], in attention mechanism, there is no strict correlation between the clusters and the distributions of attribute selection, which harms the performance of clustering.

To tackle the aforementioned issues, we propose to **C**o-cluster the **I**nteractions via **A**ttentive **H**ypergraph neural network (**CIAH**). Specifically, for modeling the entire interactions, we construct a hypergraph to connect an arbitrary number of nodes by hyperedges, which are suitable to represent entire interaction. Then, we propose an attentive hypergraph neural network to explicitly learn the representations of entire interactions (hyperedges), where an attention mechanism is adopted to select important attributes for explanations. To address the inconsistency between attention weights and true importance, enlightened by salient methods being regarded as the ground-truth of importance in the field of computer vision [27, 29], we propose a *saliency-based consistency* to make the distribution of attribute selection (i.e., attention weights) be consistent with the salient importance. Moreover, in order to ensure the correspondence between the clusters and distributions of attribute selection, motivated by [38], we propose a *cluster-based consistency*: the entire interactions within the same cluster should share similar distributions of attribute selection, while those in different clusters are dissimilar. To this end, we propose a novel co-clustering method to perform a joint clustering for the representations of entire interactions and the corresponding distributions of attribute selection to improve both the performance and explainability of clustering. In summary, the main contributions of this paper can be summarized as follows:

- To the best of our knowledge, this is the *first* attempt to cluster the entire interactions, which can discover more comprehensive and explainable cluster pattern from complex interaction data.
- To this end, we propose a novel co-clustering method for entire interactions based on attentive hypergraph neural network, namely CIAH. With hypergraph modeling, it designs an attentive hypergraph neural network followed by a novel co-clustering process with a saliency-based and a cluster-based consistencies.
- Extensive experiments have demonstrated the effectiveness of our method for clustering the entire interactions. Furthermore, offline and online recommendation experiments verify its practical value in downstream applications.

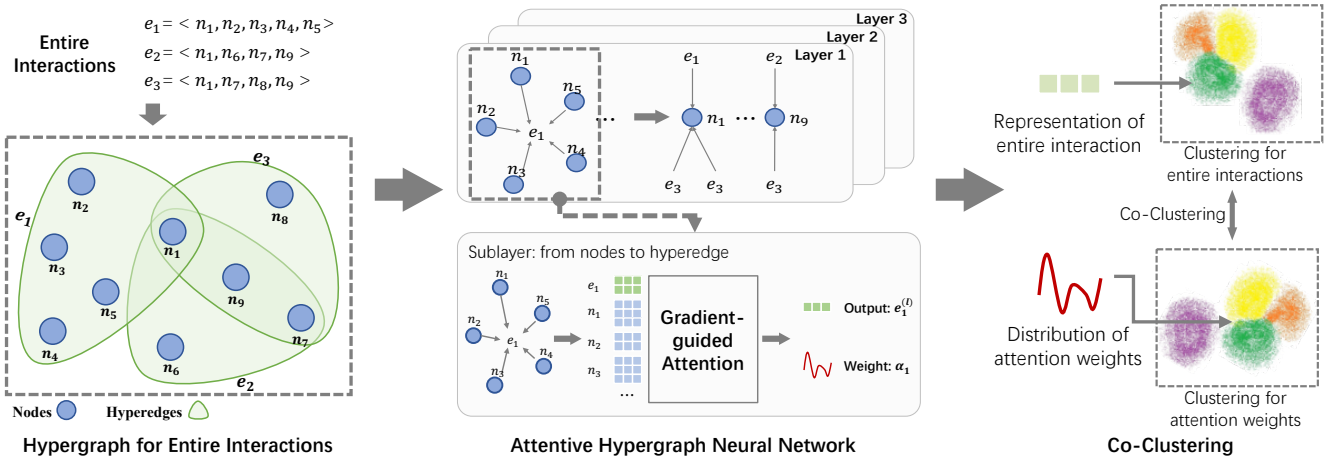


Figure 2: Illustration of CIAH (object attributes and interaction attributes are default to make the hypergraph clearer).

2 RELATED WORK

In terms of the clustering methods on graph, in early studies, methods usually perform a network embedding method to learn the representation of interaction objects and then apply clustering methods [7, 11]. Then, recent studies explore to combine the strength of both attributes and structures, and design attributed graph methods for better clustering performance such as statistical method [38], graph neural networks [4], etc. Recently, more and more researchers further consider the different types of nodes or links, and better model the interactions via heterogeneous graph [2, 16, 22]. However, these methods still only model and leverage part of the information in entire interactions, thus limiting the clustering performance.

The explanations of clustering also attract widespread attention to reveal model decisions. Existing methods usually provide clustering explanations by finding/selecting concise attributes involved in the interactions, which can be divided into two groups. One group is known as post-modeling explainability [6, 21], but is questioned that it cannot provide direct insight into the model decisions [23]. The other group usually integrates decision trees [23] or rule-learning module [10] into the clustering methods to select key attributes and performs pre-modeling explainability. These methods are specialized to some specific clustering methods and cannot be simply generalized to our clustering of entire interactions. Differently, we choose attention mechanism, which is widely used in deep neural networks, to select attributes from entire interactions.

Recently, some researchers further generalized graph representation methods into hypergraph and developed hypergraph neural network methods so that more complex and extensive information can be leveraged [3, 8, 9, 15, 31, 37, 40]. However, these methods only focus on some specific types of objects and cannot learn the representations of entire interactions. In addition, they can neither select important attributes to provide explanations, since the attention-based hypergraph methods mostly target at node level rather than feature level[3, 40]. Therefore, they are not suitable for our task.

3 PRELIMINARY

In this section, we define the basic aforementioned concepts. As stated above, an entire interaction can be seen as an action that occurred among multiple attributed interaction objects under some interaction attributes such as temporal-spatial contexts, etc. We formalize the *entire interactions* as follows.

Definition 1. ***Entire Interaction.*** *Given the set of objects $\mathcal{N}$, the set of object attributes $\mathcal{A}^o$ and the set of interaction attributes $\mathcal{A}^i$, an entire interaction $e = (\mathcal{N}_e, \mathcal{A}_e^o, \mathcal{A}_e^i)$ is a tuple including all involved objects $\mathcal{N}_e = \{n_1, \cdots, n_{|\mathcal{N}_e|} | n_j \in \mathcal{N}\}$ with their attributes $\mathcal{A}_e^o = \{a_1, \cdots, a_{|\mathcal{N}_e|} | a_j \subset \mathcal{A}^o\}$ and corresponding interaction attributes $\mathcal{A}_e^i \subset \mathcal{A}^i$.*

Due to the complexity of entire interactions, existing methods only model and leverage part of the information in entire interactions for simplification. In this work, we aim to directly cluster the entire interactions. Therefore, the problem can be formalized as follows.

Definition 2. ***Clustering on Entire Interactions.*** *Given a set of entire interactions $\mathcal{E} = \{e_1, \cdots, e_{|\mathcal{E}|}\}$ and the number of clusters C, the goal of clustering on entire interactions is to partition $\mathcal{E}$ into C disjoint subsets, such that the entire interactions $e_s, e_t \in \mathcal{E}$ within the same cluster have similar distribution of attribute selection $P(A_{e_s}) \approx P(A_{e_t})$, while those in different clusters are dissimilar, where $\mathcal{A}_e = \mathcal{A}_e^o \cup \mathcal{A}_e^i$ and $P(\cdot)$ is a categorical distribution representing the distribution of attribute selection.*

Hence, the clusters are obtained by summarizing all the involved objects and attributes, and become more explainable by the key attributes according to the distribution of attribute selection.

4 METHODOLOGY

In this section, we propose a novel attentive hypergraph-based co-clustering method for entire interactions. As illustrated in Figure 2, we first construct a hypergraph to model the entire interactions, where each entire interaction is represented as a hyperedge connecting all its involved objects. Then we design an attentive

hypergraph neural network to explicitly learn the representations of entire interactions (hyperedges) and meanwhile select the associated key attributes by attention mechanism. Moreover, we propose a saliency-based consistency to make the distributions of attribute selection be consistent with real importance of attributes by a salient method, i.e., the integrated gradient [29] is introduced to guide the attention. Finally, a novel co-clustering method is proposed to perform a joint clustering for both the entire interactions and corresponding distributions of attribute selection for the cluster-based consistency.

4.1 Hypergraph for Entire Interactions

As illustrated in left of Figure 2, we model the entire interactions as a hypergraph, where each hyperedge represents an entire interaction, connecting all types of its involved objects. Specifically, for an entire interaction e_1 involving 5 objects $n_1, \cdots, n_5$, we build a hyperedge to connect them. Besides, we attach their object attributes $a_j (j = 1, \cdots, 5)$ to the node features, and attach the interaction attributes $\mathcal{A}^i_{e_1}$ such as tempera-spatial contexts (should be seen as the attributes of entire interactions rather than objects) to hyperedge features. Therefore, such a hyperedge and its connecting nodes together with their features can represent an instance of entire interactions.

4.2 Attentive Hypergraph Neural Network

The framework of graph neural networks can effectively leverage neighboring nodes/edges for information augmentation. However, most existing hypergraph neural networks can only learn explicit representations of nodes while ignoring to embed the hyperedges. Therefore, we design a hypergraph neural network to explicitly learn the representations of both nodes and hyperedges. Meanwhile, an attention mechanism is adopted to select the key attributes.

4.2.1 Initialization. In order to facilitate the attention mechanism to select attributes, we initial the representations of hyperedges and nodes as feature matrices rather than vectors. Formally, $\boldsymbol{e}_i^{(0)} \in \mathbb{R}^{f_i \times d}$ and $\boldsymbol{n}_j^{(0)} \in \mathbb{R}^{f_j \times d}$ denote the initial d-dimensional representations of hyperedge e_i and node n_j, respectively, where each row of the feature matrix denotes a specific attribute of the node/hyperedge.

4.2.2 Layer-wise Aggregation. Given a hypergraph with incidence matrix $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{E}|}$, let $\boldsymbol{D_e}$, $\boldsymbol{D_n}$ and $\boldsymbol{W}$ denote the diagonal matrices of the edge degrees, the node degrees and the pre-defined weights of hyperedges (default is 1), respectively. As studied in previous works [9, 37], the spectral hypergraph convolution can be simplified and formulated as

$$\boldsymbol{Y} = \boldsymbol{D_n}^{-1/2} \boldsymbol{H} \boldsymbol{W} \boldsymbol{D_e}^{-1} \boldsymbol{H}^\top \boldsymbol{D_n}^{-1/2} \boldsymbol{X} \boldsymbol{\Theta}, \tag{1}$$

where $\boldsymbol{X}$ denotes the feature matrix of nodes and $\boldsymbol{\Theta}$ is the trainable filter parameter. This form can be also understood as the following information aggregation rule: The node features are first aggregated according to hyperedges by multiplying matrix $\boldsymbol{H}^\top$, forming hyperedge features. Then the updated node features are obtained by aggregating features of their belonging hyperedges, achieved by multiplying $\boldsymbol{H}$.

Inspired by this calculation procedure, we parameterize this information aggregation process as a two-stage attentive aggregation rule from layer l to layer $l+1$ as follows:

$$\boldsymbol{e}_i^{(l+1)} = \text{att}(\boldsymbol{e}_i^{(l)}, \{\boldsymbol{n}_j^{(l)} \mid n_j \in e_i\}), \tag{2}$$

$$\boldsymbol{n}_j^{(l+1)} = \text{att}(\boldsymbol{n}_j^{(l)}, \{\boldsymbol{e}_i^{(l+1)} \mid n_j \in e_i\}). \tag{3}$$

Taking Eq. (2) as an example, the updated embedding $\boldsymbol{e}_i^{(l+1)}$ of hyperedge e_i is aggregated from itself and its connecting nodes $\{\boldsymbol{n}_j^{(l)} \mid n_j \in e_i\}$. In order to recognize and select important attributes during clustering entire interactions, we concatenate the feature matrices of both this hyperedge and its connecting nodes on rows into a combined feature matrix $\boldsymbol{X}_i^{(l)}$[1] and then apply a feature-aware soft attention. Formally,

$$\boldsymbol{X}_i^{(l)} = \text{concat}(\boldsymbol{e}_i^{(l)}, \{\boldsymbol{n}_j^{(l)} \mid n_j \in e_i\}), \tag{4}$$

$$\boldsymbol{\alpha}_i^{(l)} = \text{softmax}(\boldsymbol{X}_i^{(l)\top} \cdot \boldsymbol{a}^{(l)}), \tag{5}$$

$$\boldsymbol{e}_i^{(l+1)} = \text{att}(\boldsymbol{e}_i^{(l)}, \{\boldsymbol{n}_j^{(l)} \mid n_j \in e_i\}) = \boldsymbol{\alpha}_i^{(l)\top} \cdot \boldsymbol{X}_i^{(l)}, \tag{6}$$

where $\boldsymbol{a}^{(l)} \in \mathbb{R}^d$ is the parameter vector in attention. Through the feature-aware soft attention, we can obtain the updated embedding of hyperedge $\boldsymbol{e}_i^{(l+1)} \in \mathbb{R}^{1 \times d}$ and the corresponding distribution of attribute selection $\boldsymbol{\alpha}_i^{(l)}$ for entire interaction e_i. Similarly, we can also obtain the node embedding $\boldsymbol{n}_j^{(l+1)}$.

4.2.3 Outputs. Since each layer represents a specific order of relations, we sum the embeddings from each layer as the final representation both for entire interactions and nodes, i.e., $\boldsymbol{e}_i = \sum_l \boldsymbol{e}_i^{(l+1)}$ and $\boldsymbol{n}_j = \sum_l \boldsymbol{n}_j^{(l+1)}$. For simplicity, we just use the attention weights in the first layer $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_i^{(0)}$ as the distribution of attribute selection towards entire interaction e_i.

4.2.4 Salient Guidance. Due to the inconsistency between attention weights and true importance [17], the distribution of attribute selection may be also questioning. Inspired by the salient explanation in the field of computer vision [29], where the salient method is regarded as the ground-truth of importance compared with the vanilla attention weights, we introduce integrated gradients to guide attention. Formally,

$$\mathcal{L}_{grad} = \sum_{e_i \in \mathcal{E}} KL(\ \text{SoftMax}(\text{IG}(\boldsymbol{\alpha}_i)) \| \boldsymbol{\alpha}_i), \tag{7}$$

where SoftMax function is used to transform the gradients into a distribution, then KL-divergence is applied to let it teach the distribution of the attention weights. Here IG denotes the integrated gradients under $\mathbf{0}$ base vector. For each element α_{ik} of $\boldsymbol{\alpha}_i$ that represents the weight of the k-th attribute in the i-th entire interaction,

$$\text{IG}(\alpha_{ik}) = \left\| \boldsymbol{x}^{(k)} \odot \int_0^1 \frac{\partial F(t \cdot \boldsymbol{x}^{(k)})}{\partial \boldsymbol{x}^{(k)}} dt \right\|, \tag{8}$$

where $\boldsymbol{x}^{(k)}$ represents the k-th row of the combined feature matrix $\boldsymbol{X}_i^{(l)}$ in Eq. 4 and $\odot$ denotes Hadamard product. Here F denotes the proposed attentive hypergraph neural network.

[1] In the first layer, $\boldsymbol{X}_i^{(0)} \in \mathbb{R}^{(f_i + \sum_j f_j) \times d}$, while in the other layers $\boldsymbol{X}_i^{(l)} \in \mathbb{R}^{(1 + \sum_j 1) \times d}$ since the attentive sumation operation.

4.3 Co-Clustering on Entire Interactions

Considering the clusters are sometimes inconsistent with the distributions of attribute selection [17], we propose a novel co-clustering method to jointly cluster both entire interactions and their corresponding attention weights, which ensures the cluster-based consistency.

After going through the above modules, we have obtained the representation of entire interactions with the corresponding distribution of attribute selection. In particular, given the i-th interaction representation $\boldsymbol{e}_i$ and the u-th trainable cluster centroid $\boldsymbol{\mu}_u$, following existing deep cluster methods [4, 35], we can measure the similarity between them based on Student's t-distribution kernel as follows:

$$q_{iu} = \frac{(1 + \left\|\boldsymbol{e}_i - \boldsymbol{\mu}_u\right\|^2 / v)^{-\frac{v+1}{2}}}{\sum_s (1 + \left\|\boldsymbol{e}_i - \boldsymbol{\mu}_s\right\|^2 / v)^{-\frac{v+1}{2}}}, \tag{9}$$

where v is the degrees of freedom of the Student's t-distribution, and following [4], we let $v = 1$ for all cases. q_{iu} can be considered as the probability of assigning interaction i to cluster u, i.e., a soft assignment. We treat $Q_{emb} = [q_{mu}] \in \mathbb{R}^{|\mathcal{E}| \times C}$ as the distribution of the assignments of all interactions. Then we can optimize Q_{emb} by learning from the high confidence assignments, forming as a target distribution $P_{emb} = [p_{iu}]$:

$$p_{iu} = \frac{q_{iu}^2 / \sum_t q_{tu}}{\sum_s (q_{is}^2 / \sum_t q_{ts})}. \tag{10}$$

Through minimizing the KL divergence between Q_{emb} and P_{emb}, the target distribution P_{emb} can help the model achieve high cohesion and low coupling of clusters [4], thus achieving clustering procedure.

However, in our task, we aim to ensure each cluster one-to-one corresponds to a distribution of attribute selection, i.e., the cluster-based consistency. Therefore, for the distribution of attribute selection $\boldsymbol{\alpha}$ of each entire interaction, we can also similarly compute the assignment distribution $Q_{wgt} = [q'_{iu}]$ and target distribution $P_{wgt} = [p'_{iu}]$ for attention weighs. Specifically, given the distribution of attribute selection $\boldsymbol{\alpha}_i$ corresponding to the i-th entire interaction, we have

$$q'_{iu} = \frac{(1 + \left\|\boldsymbol{\alpha}_i - \boldsymbol{\nu}_u\right\|^2 / v)^{-\frac{v+1}{2}}}{\sum_s (1 + \left\|\boldsymbol{\alpha}_i - \boldsymbol{\nu}_s\right\|^2 / v)^{-\frac{v+1}{2}}}, \tag{11}$$

$$p'_{iu} = \frac{q'^2_{iu} / \sum_t q'_{tu}}{\sum_s (q'^2_{is} / \sum_t q'_{ts})}, \tag{12}$$

where $\boldsymbol{\nu}_u$ is the u-th cluster centroid for attention weights. Finally, in order that these two groups of distributions can guide each other until convergence, we propose a groundbreaking co-cluster method by exchanging their target distributions. Formally, we minimize the following equation:

$$\mathcal{L}_{clu} = KL(P_{emb} \| Q_{wgt}) + KL(P_{wgt} \| Q_{emb}). \tag{13}$$

Through the above objective function, we can achieve individual clustering for entire interactions and attention weights, and meanwhile make them learn from each other. We will give a brief proof at the end of this section.

4.4 Model Training

For self-supervised training of our model, with the calculated hyperedge embeddings and node embeddings, we reconstruct the incidence matrix H of the hypergraph with a distance-based contrastive loss:

$$d_{ij} = \|\boldsymbol{e}_i - \boldsymbol{n}_j\|, \tag{14}$$

$$\mathcal{L}_{self} = \frac{1}{2} \sum_{i,j} y_{ij} d_{ij}^2 + (1 - y_{ij}) \max(0, m - d_{ij})^2, \tag{15}$$

where y_{ij} denotes the existence of a relationship between node i and hyperedge j, and m is the margin hyper-parameter. In this work, we set $m = 1$ for all the cases. Then by merging the above sub-objective functions, we jointly train our model by the salient guidance, co-clustering and self-supervision modules. Therefore, we can conclude the following loss function,

$$\mathcal{L} = \mathcal{L}_{self} + \gamma_c \mathcal{L}_{clu} + \gamma_g \mathcal{L}_{grad} + \eta \|\Theta\|, \tag{16}$$

where γ_c and γ_g are the loss coefficients. For simplicity, we set $\eta = 0.1$ for the regularization for model parameters $\|\Theta\|$.

4.5 Theoretical Analysis for Co-clustering

In the following, we provide a brief theoretical analysis to illustrate why our proposed co-cluster is effective.

Theorem 1. *The objective function Eq. (13) is equivalent to finding a solution that satisfies: $Q_{emb} = P_{emb}$, $Q_{wgt} = P_{wgt}$ and cluster-based consistency $Q_{emb} = Q_{wgt}$.*

Proof. The objective function Eq. (13) is essentially to solve the following system of equations,

$$\begin{cases} Q_{emb} = P_{wgt} = P(Q_{wgt}) \\ Q_{wgt} = P_{emb} = P(Q_{emb}) \end{cases} \tag{17}$$

where $Q \in \mathbb{R}^{|\mathcal{E}| \times C}$ and $P : \mathbb{R}^{|\mathcal{E}| \times C} \to \mathbb{R}^{|\mathcal{E}| \times C}$ is rewritten as a function of Q according to Eq. (10) for ease of description.

Obviously, it is only necessary to prove that if P locally converges to a certain solution Q^* with a given initial value, then the composite function $P \circ P$ will also locally converge to Q^* at the same initial value. Because in this case, we have

$$Q^* = Q_{emb} = P(Q_{wgt}) = P \circ P(Q_{emb}) = P(Q_{emb}) = P_{emb}. \tag{18}$$

Similarly, we have $Q_{wgt} = P_{wgt}$. Hence we can achieve the cluster-based consistency $Q_{emb} = P_{wgt} = Q_{wgt}$.

Note that the convergence is determined by the P function. In fact, due to that the neural network is equivalent to its permutation by dimension, the convergence of P function is not unique in the matrix space $A = \mathbb{R}^{|\mathcal{E}| \times C}$. However, denoting $\mathcal{R}$ as the set of rotations that maintains invariance, the quotient space $A/\mathcal{R}$ is a Banach space and P has unique convergence in this quotient space and becomes a contraction mapping. According to Banach Fixed-point Theorem, since $(A/\mathcal{R}, \|\cdot\|)$ is a non-empty complete metric space, where $\|\cdot\|$ is the metric function, such as Euclidean distance in this work, there exists the $L \in [0, 1)$ in the neighborhood N_{Q_*} of Q_*, such that $\forall Q_x, Q_y \in N_{Q_*}$, we have

$$\|P(Q_x) - P(Q_y)\| \leqslant L \|Q_x - Q_y\|. \tag{19}$$

Table 1: Statistics of Datasets. # HE, # Attr. and # Cate. denote the number of hyperedges, attributes and categories.

Dataset	# Nodes		# HE	# Attr.	# Cate.
ACM	Paper	4,025	4,025	1,902	3
	Author	7,167			
	Field	60			
IMDB	Movie	4,661	4,661	1,256	3
	Actor	5,841			
	Director	2,270			
MT-4	User	37,748	40,000	5789	4
	POI	17,994			
	Item	123,629			
MT-9	User	79,967	90,000	5945	9
	POI	25,834			
	Item	245,064			

Then, we can derive

$$\|P \circ P(Q_x) - P \circ P(Q_y)\| \leqslant \tag{20}$$

$$L\|P(Q_x) - P(Q_y)\| \leqslant L^2\|Q_x - Q_y\|. \tag{21}$$

Therefore, $P \circ P$ will also converge to Q^*, which means Eq. (18) holds. Besides, $P \circ P$ has a faster convergent speed L^2. □

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Datasets. We conduct the experiments on two public datasets and two industrial datasets. The statistics are reported in Table 1 and the descriptions are detailed as follows.

- **ACM.** The ACM dataset[2] contains 4,025 papers with two other types of nodes: author and field. We build a hypergraph where each hyperedge connects a paper and all its corresponding authors and fields, thus representing an entire interaction towards a paper. Therefore, we can label the hyperedges according to the category of its containing paper. The attribute of each paper is the bag-of-words representation of abstract while the attributes of other nodes and hyperedges are one-hot representation since their features are not provided by the dataset.
- **IMDB.** The IMDB dataset [39] contains 4,661 movies with two other types of nodes: actors and directors. Similar to ACM, we also build a hypergraph where each hyperedge representing an entire interaction towards a movie connecting its actors and directors. We label the hyperedges according to the category of its containing movie. The attribute of each movie is the bag-of-words representation of its plots while the attributes of other nodes and hyperedges are one-hot representation.
- **MT-4, MT-9.** We build two real-world datasets of different levels of difficulty from the food delivery industry, i.e., Meituan Waimai platform[3]. We collect 40,000/90,000 orders of user purchases of foods in Beijing District. Each order is an entire interaction instance, containing a user, a POI (Point-of-Interest), several items with attributes and corresponding interaction contexts, and is tagged with one of 4/9 purchasing scenes (such as white-collar working meals, student afternoon coffee, etc.) as labels, denoting as MT-4 and MT-9, respectively. Afterward, we use these orders to build a hypergraph where each hyperedge connects its involved attributed user, POI and items. The attributes of hyperedges are the interaction contexts.

5.1.2 Baselines. To validate the effectiveness of our CIAH, we compare it with the following 4 groups of methods.

- **Methods only considering attributes:** K-means [13]: It is a classical clustering method based on the raw multi-hot features. AE [14]: It performs K-means on the representations learned by an auto-encoder.
- **Methods only considering graph structure:** node2vec [11] and metapath2vec [7]: They perform a normal and metapath-based random walk on graphs followed by Skip-Gram, respectively, and then apply K-means on the learned node embeddings.
- **Methods considering both graph structure and attributes:** ACMIN [38]: It is a SOTA approach for k-AGC (k-Attributed Graph Clustering) task that yields clusters where the nodes within the same cluster share similar topological and attribute characteristics, while those in different clusters are dissimilar. SDCN [4]: It combines the strengths of both attributes by autoencoder and structures by GCN for deep clustering with a delivery operator and a dual self-supervised module. HGT [16]: It is a SOTA heterogeneous graph embedding model that incorporates the types of nodes and edges into the propagation step.
- **Methods considering both hypergraph structure and attributes:** HGNN [9]: It is a hypergraph spectral convolution network framework. We split its convolution into 2-stage for explicit hyperedge representation. HGNN+: Since HGNN cannot leverage the hyperedge attributes in datasets MT-4/9, we convert the attributed hyperedge into an attributed virtual node connected together with other original nodes. AHGAE [15]: It is a hypergraph auto-encoder for relational data clustering. We similarly modify it like HGNN and construct its variant AHGAE+.

For the methods only considering attributes, we fuse the attributes of the hyperedge and its connecting nodes into a multi-hot vector on the all attribute vocabulary as an entire interaction sample.

For the methods considering graph structure, for ACM and IMDB datasets, we directly use the widely applied graph structure [16]. For MT-4/9 datasets, we transfer the hypergraph into a graph, i.e., we introduce a summary node to replace each of hyperedges and link the summary nodes with the corresponding nodes that are originally connected by hyperedges.

5.1.3 Metrics. The category of hyperedges (entire interactions) is taken as the ground truth. Following [4], we employ two popular metrics: Normalized Mutual Information (NMI) and Average Rand Index (ARI). The average result and standard deviation are reported based on 10 repeated tests.

5.1.4 Implementation Detail. We implement the proposed method based on Tensorflow [1]. For our method, we set the dimension of attribute embeddings as 64 for the public datasets ACM and IMDB for fair comparison and 16 for the industrial datasets MT-4/9 for saving memory. In order to facilitate the attribute selection module,

[2]https://data.dgl.ai/dataset/ACM.mat
[3]https://waimai.meituan.com/

Table 2: Clustering results on 4 datasets (mean±std). The best and second best results are bold and underlined, respectively.

Method	Information			Metrics	Dataset			
	Hyper-	Graph	Attr.		ACM	IMDB	MT-4	MT-9
K-means	✗	✗	✓	NMI	37.47 ± 0.39	0.89 ± 0.06	31.05 ± 3.40	11.77 ± 2.77
				ARI	28.83 ± 0.38	1.30 ± 0.06	28.50 ± 3.01	7.11 ± 1.62
AE	✗	✗	✓	NMI	30.62 ± 5.71	2.05 ± 0.63	4.92 ± 2.37	2.74 ± 0.96
				ARI	27.29 ± 7.56	1.57 ± 0.80	4.38 ± 2.85	1.65 ± 0.75
node2vec	✗	✓	✗	NMI	38.51 ± 0.11	5.22 ± 0.71	0.01 ± 0.01	0.03 ± 0.01
				ARI	31.08 ± 0.94	6.02 ± 0.45	0.01 ± 0.01	0.09 ± 0.03
metapath2vec	✗	✓	✗	NMI	19.96 ± 1.46	1.51 ± 0.59	14.32 ± 2.21	6.80 ± 0.77
				ARI	21.00 ± 1.33	1.50 ± 0.69	4.77 ± 1.97	5.59 ± 0.28
ACMIN	✗	✓	✓	NMI	18.66	1.71	14.16	11.35
				ARI	12.91	1.01	5.22	4.95
SDCN	✗	✓	✓	NMI	41.77 ± 0.73	3.37 ± 0.20	22.91 ± 6.35	7.08 ± 6.52
				ARI	37.45 ± 0.67	2.73 ± 0.18	21.54 ± 9,92	4.82 ± 2.66
HGT	✗	✓	✓	NMI	47.49 ± 2.40	5.50 ± 0.06	27.25 ± 5.91	10.51 ± 6.50
				ARI	42.90 ± 1.79	5.13 ± 0.04	25.57 ± 3.92	8.41 ± 3.58
HGNN	✓	✓	✓	NMI	29.62 ± 4.65	2.05 ± 0.01	11.26 ± 2.34	6.93 ± 1.21
				ARI	28.21 ± 7.15	1.28 ± 0.05	10.52 ± 2.17	4.37 ± 1.80
HGNN+	✓	✓	✓	NMI	-	-	32.56 ± 5.11	12.28 ± 4.49
				ARI	-	-	30.07 ± 4.99	9.77 ± 1.62
AHGAE	✓	✓	✓	NMI	46.46 ± 2.92	1.73 ± 1.35	14.40 ± 4.16	2.12 ± 1.24
				ARI	40.11 ± 3.27	1.44 ± 1.02	13.26 ± 3.72	1.32 ± 0.14
AHGAE+	✓	✓	✓	NMI	-	-	29.34 ± 4.06	13.87 ± 4.37
				ARI	-	-	26.81 ± 2.94	8.17 ± 2.50
CIAH	✓	✓	✓	NMI	**54.77 ± 0.55**	**12.60 ± 0.70**	**39.29 ± 4.31**	**14.14 ± 3.02**
				ARI	**45.25 ± 0.20**	**11.26 ± 0.86**	**36.54 ± 1.31**	**10.01 ± 1.98**

∗ The two variants, HGNN+ and AHGAE+, cannot be used for datasets ACM and IMDB, since there are no hyperedge features in these two datasets. So we use the symbol "-" to denote "no results".

we set the number of layers of our method as 1. For all baselines, we set the hidden dimensions as 64 for fair comparison, and set the number of layers as their suggested value (usually equal to 2). The best coefficients of losses are searched from $\{0, 01, 0, 1, 1, 10, 50\}$, and we set $\gamma_c = 1$ for ACM and IMDB and 0.1 for MT-4/9, $\gamma_g = 1, 50, 10, 10$ for the four datasets respectively. In the training stage, we apply Adam [20] for optimizing with the learning rate as 0.005 for ACM and IMDB and 0.002 for MT-4/9. All the experiments are performed in NVIDIA Tesla P40 Cluster. To facilitate related research, we release our implementation to the public[4].

5.2 Clustering Results

Table 2 reports the clustering results on the public and industrial datasets. As shown, our CIAH significantly outperforms all the baselines by a large margin, which shows the effectiveness of our proposed method on clustering entire interactions. In detail, we can also conclude as follows.

On those datasets with fewer attributes, i.e., ACM and IMDB, the structure-only methods, especially the classic node2vec, performs better than the attribute-only methods K-means and AE. Besides, methods considering both attribute and graph structure, e.g., HGT, generally further improve the cluster performance, illustrating the advantages of considering attributes and structure simultaneously. Though the hypergraph-based baselines only obtain relatively poor performance, our CIAH still obtains the best performance, which verifies that simply applying hypergraphs to integrate information is not effective.

However, this is not the case on attribute-rich datasets, i.e., MT-4 and MT-9. The performance of attribute-only methods far exceeds the structure-only methods and even the methods combining both attributes and graph structure. Nevertheless, the hypergraph-based methods can achieve obviously better results, which indicates the necessity to model entire interactions as hypergraph. Especially in the recommendation field, introducing a summary node for each interaction can only model the pair-wise interaction relations, while an interaction is actually an indivisible whole. Thereby forcibly dividing it into several pair-wise sub-relations causes information loss, which again verifies the effectiveness of our model.

5.3 Analysis of Clustering Explanations

To analyze the explainability of our model, Table 3 illustrates several quantitative metrics for explainability and the top 4 important attributes of each category in dataset MT-4 according to the attention weights from CIAH, the "RMGrad" variant that ignores the

[4]https://github.com/ytc272098215/CIAH

Table 3: Quantitative analysis of clustering explanations and illustration of the top 4 key attributes. The most critical attributes are in bold which are used to divide the dataset.

	CIAH	RMGrad	$\text{CIAH}_{\setminus 2}$
NDCG	0.8154	0.5154	0.2177
MRR	0.7500	0.3541	0.1251
NDCG_{pcc}	0.5642	0.5328	0.3322
Cate. 1	Gender0	Gender0	WhiteCollar
	Lunch	AorType0	AorType0
	AorType0	**Lunch**	PriceHigh
	PriceHigh	PriceHigh	**Lunch**
Cate. 2	Gender0	Gender0	Student
	AfternoonTea	AorType0	Gender0
	AorType0	**AfternoonTea**	PriceMed
	PriceHigh	PriceHigh	PriceHigh
Cate. 3	**Supper**	Gender1	WhiteCollar
	Gender1	**Supper**	Student
	AorType0	WhiteCollar	HousingEstate
	WhiteCollar	AorType0	Gender1
Cate. 4	**Breakfast**	PriceMed	WhiteCollar
	AorType0	WhiteCollar	AorType0
	WhiteCollar	AorType0	OfficeBuilding
	Student	**Breakfast**	**Breakfast**

saliency-based consistency and variant "$\text{CIAH}_{\setminus 2}$" that ignores the two consistencies.

In order to quantify the explainability of attribute selection module, we take criterion attributes (in bold) for dividing the dataset into categories as ground-truth attributes, which are provided by the dataset, and then use the ranking metrics to evaluate, i.e., Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (NDCG). Besides, in fact, some other attributes are also discriminative for different categories. Therefore, to check whether our model recognizes these second-important attributes, we also take the Pearson correlation coefficient (PCC) between attributes and categories as the ground truth for NDCG, here denoted as NDCG_{pcc}. As we can see, compared with our CIAH, the variant RMGrad obtains lower MRR, NDCG, NDCG_{pcc} and $\text{CIAH}_{\setminus 2}$ performs the worst. This demonstrates that the proposed two consistencies are helpful for improving explainability.

For a more intuitively comparison, we also show the top 4 important attributes of each category. As shown in Table 3, they cannot be fully recognized when the two consistencies are both ignores. Fortunately, with the help of the cluster-based consistency of co-clustering module, "RMGrad" successfully distinguishes them. Moreover, our CIAH can assign greater weights to these key attributes when the saliency-based consistency is further considered. It verifies that the attention module can correctly extract important attributes due to the guidance of the salient method and our co-clustering method and its explainability can be gradually improved by considering the two consistencies one by one.

5.4 Model Analysis

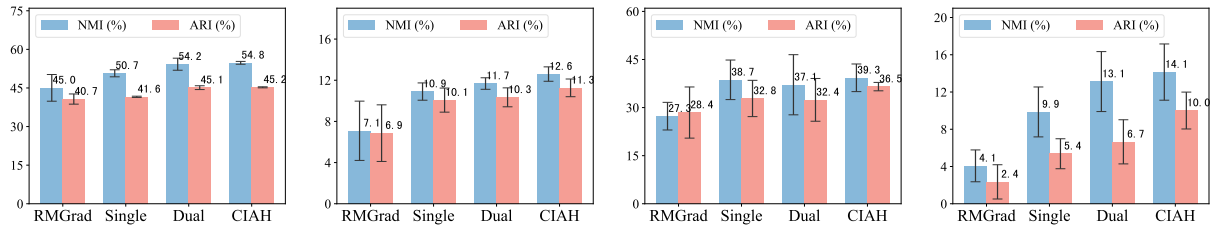
5.4.1 Comparison of Variants. In this subsection, we compare our model CIAH with 3 variants to validate the design of each module. Specifically, *RMGrad* is a variant that removes the gradient guided module. *Single* replaces the proposed co-cluster module with an ordinary clustering module, which only clusters hyperedge representations. *Dual* denotes the variant that replaces the co-cluster module with a dual clustering module, which individually clusters hyperedge representations and attention weights with added an extra constraint $Q_{emb} = Q_{wgt}$ by JS-divergence. As reported in Figure 3, we can draw the following conclusions. Firstly, *RMGrad* obtains an obvious performance drop, indicating the effectiveness of our gradient-based attention module. Secondly, the performance of *Single* is also limited, and hence verifies the consideration necessity of the consistency between basic cluster and the corresponding attention weight distribution. Finally, *Dual* still fails to outperform the complete CIAH, in terms of both values and variances, which demonstrates the effectiveness of the design of our co-cluster, which can enhance each other's clustering performance.

5.4.2 Impact of Different Model Depths. In this subsection, we study the clustering performance of our model under different depths of the hypergraph convolutional layer (e.g., CIAH with 1,2,3 layers). As shown in Figure 4, we have observed that increasing the depth cannot substantially enhance the clustering performance. In detail, the model may benefit from considering higher-order neighbors as the depth increases, and then it may be limited by the overfitting caused by the more increasing depth. Therefore, as can be seen, the best performance occurs when the number of layers is 2 for all datasets, while the sensitivities of the four datasets to the layer numbers are not completely the same, depending on the difficulty and complexity of the dataset. It is worth noting that although the optimal performance occurs when our CIAH has 2 layers, we still we set the number of layers of our method as 1 in other experiments in this paper, which is to make the attribute selection module of the model have a stronger explainability.

5.4.3 Visualization. In this subsection, we adopt t-SNE [32] to project the learned entire interaction embeddings and the distributions of attribute selection into a 2-dimensional space and visualize them. Figure 5 and 6 visualize the embeddings of entire interactions and the distributions of attribute selection from our CIAH. As shown in Figures 5 and 6, an obvious clustering phenomenon can be observed for both the entire interactions and the distributions of attribute selection, especially on ACM and MT-4 datasets. In addition, we also find that the clustering phenomena of the two groups are relatively similar. There are clear division boundaries in the clusters of entire interaction embeddings, and relatively clear division boundaries can also be observed in the clusters of distributions of attribute selection, especially in dataset ACM and MT-4. This demonstrates the effectiveness of the proposed co-clustering module for the cluster-based consistency, i.e. it achieves a one-to-one correspondence between the clusters and corresponding distribution of attribute selection.

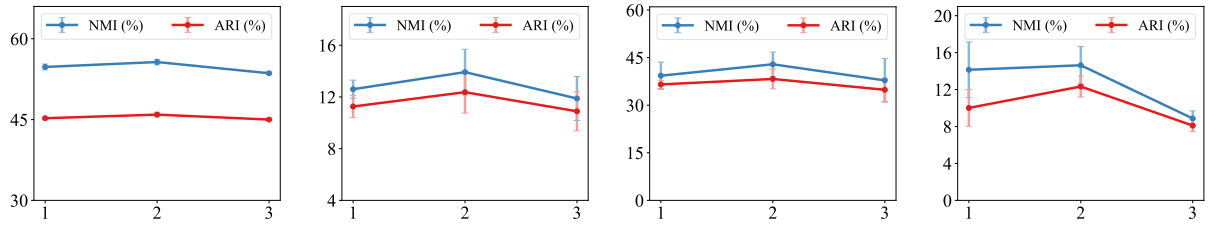
5.5 Application on Recommendation

In this subsection, we take recommendation as an application example to show the practical usage of clustering results. Specifically, we conduct an offline click-through rate (CTR) prediction experiment and an A/B online testing.



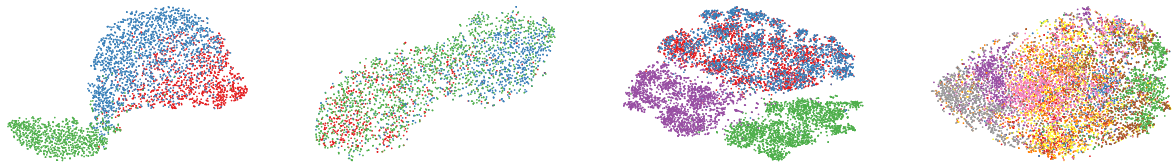
(a) ACM (b) IMDB (c) MT-4 (d) MT-9

Figure 3: Clustering results of different variants (mean and variance are given in bars and error bars, respectively).



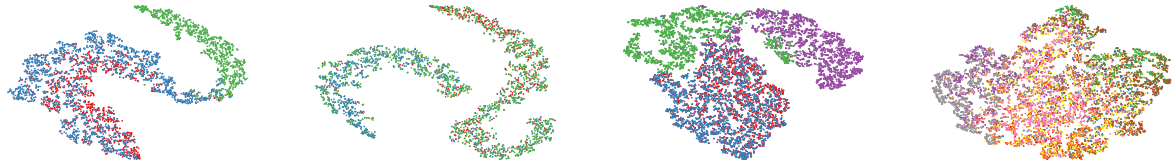
(a) ACM (b) IMDB (c) MT-4 (d) MT-9

Figure 4: Clustering results with 1,2,3 propagation layers (mean and variance are given in lines and error bars, respectively).



(a) ACM (b) IMDB (c) MT-4 (d) MT-9

Figure 5: 2D t-SNE visualization of the entire interaction embeddings from CIAH on the four datasets.



(a) ACM (b) IMDB (c) MT-4 (d) MT-9

Figure 6: 2D t-SNE visualization of the distributions of attribute selection from CIAH on the four datasets.

5.5.1 Offline CTR Prediction. We select three typical models to show the Area Under the Curve (AUC) performance without clustering information, and adapt them to access this extra information for comparison. Specifically, we choose a classic feature-based method, DeepFM [12], a SOTA feature-based method, AutoInt [28], and a heterogeneous graph neural network based context-aware recommendation approach, NIRec [19]. For feature-based models, let DeepFM+ and AutoInt+ denote the variants that the clustering information is introduced by concatenation, i.e., as an extra input channel. For graph-based model, NIRec+ denotes the variant that the clustering information is introduced by a new type of nodes. Since the metric AUC relies on the negative sampling strategy, we

Table 4: AUC comparisons. "+" represents that the clustering information is added.

Method	MT Dataset		
	50%	75%	100%
DeepFM	0.5832	0.5879	0.5912
DeepFM+	**0.6804**	**0.6823**	**0.6848**
AutoInt	0.5785	0.5868	0.5891
AutoInt+	**0.6781**	**0.6803**	**0.6892**
NIRec	0.6589	0.6712	0.6877
NIRec+	**0.7118**	**0.7597**	**0.7660**

construct a new CTR dataset from Meituan Waimai platform, where positive samples are the user-POI pairs and negative samples are generated by randomly replacing POIs in the positive samples. We further vary the ratio of the training set to verify its robustness. As shown in Table 4, compared with the vanilla baselines, the variants that can leverage the clustering information achieve better performance. This noticeable improvement validates the benefits of clustering information for downstream tasks.

Table 5: Online improvements.

Indicator	PVCTR	UVCTR	UVCXR	RPM
Improvement	+0.88%	+0.35%	+0.44%	+0.08%

5.5.2 Online A/B Testing. Moreover, we also carried out an evaluation in the online A/B testing. We introduce the clustering information into the recommendation system of MT App and focus on item recommendation for users. Specifically, the compared online system is a PLE [30]-based rank model with multiple features as input. For comparison, we process the clustering assignments as a new kind of feature and add it into the system (just like the aforementioned DeepFM) for a three-day online A/B testing. The following online indicators concerned by industry are reported in Table 5: CTR of page view (PVCTR), CTR of unique visitor (UVCTR), scaled click conversion rate of unique visitor (UVCXR) and revenue per mille (RPM). It can be seen that after adding clustering information, there are different degrees of performance improvement for the online system in all metrics. The results demonstrate that the clustering information can improve the performance in the online recommender system consistently.

6 CONCLUSION

In this paper, we make the first attempt to cluster the entire interactions, which can extract more comprehensive and explainable cluster pattern from real interaction data. Particularly, we propose a Co-clustering method for the entire Interactions via Attentive Hypergraph neural network (CIAH). With hypergraph modeling for entire interactions, it designs an attentive hypergraph neural network followed by a novel co-clustering process with a saliency-based and a cluster-based consistencies for further improvements. Extensive experiments verify the effectiveness of our CIAH on both public datasets and real industrial datasets. Besides, a recommendation experiment is taken as application example to show the practical usage of clustering results. In future work, we will explore a more explainable solution for feature-level interactions than simply stacking layers and a better usage in downstream tasks.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China (No. U20B2045, 62192784, 62172052, 62002029, 61772082) and also supported by Meituan.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*. USENIX Association, 265–283.

[2] Karan Aggarwal, Georgios Theocharous, and Anup B. Rao. 2020. Dynamic Clustering with Discrete Time Event Prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1501–1504.

[3] Song Bai, Feihu Zhang, and Philip HS Torr. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognition* 110 (2021), 107637.

[4] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. 2020. Structural Deep Clustering Network. In *Proceedings of the Web Conference*. 11.

[5] Chantima Buaklee and Sukree Sinthupinyo. 2018. Similar Cluster recommendation of Product Purchases by Pages liked Analysis. In *10th International Conference on Electronics, Computers and Artificial Intelligence*. IEEE, 1–5.

[6] Daniel Deutch and Nave Frost. 2019. Constraints-based explanations of classifications. In *ICDE*. IEEE, 530–541.

[7] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.

[8] Haoyi Fan, Fengbin Zhang, Yuxuan Wei, Zuoyong Li, Changqing Zou, Yue Gao, and Qionghai Dai. 2021. Heterogeneous Hypergraph Variational Autoencoder for Link Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[9] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 3558–3565.

[10] Mohamed H Gad-Elrab, Daria Stepanova, Trung-Kien Tran, Heike Adel, and Gerhard Weikum. 2020. ExCut: explainable embedding-based clustering over knowledge graphs. In *International Semantic Web Conference*. Springer, 218–237.

[11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *KDD*. Association for Computing Machinery, 855–864.

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 1725–1731.

[13] J. A. Hartigan and M. A. Wong. 1979. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics* 28, 1 (1979), 100.

[14] G. E. Hinton. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (July 2006), 504–507.

[15] Youpeng Hu, Xunkai Li, Yujie Wang, Yixuan Wu, Yining Zhao, Chenggang Yan, Jian Yin, and Yue Gao. 2021. Adaptive Hypergraph Auto-Encoder for Relational Data Clustering. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[16] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. In *Proceedings of The Web Conference*. ACM, 2704–2710.

[17] Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *NAACL*. Association for Computational Linguistics, 3543–3556.

[18] Yugang Ji, Chuan Shi, Yuan Fang, Xiangnan Kong, and Mingyang Yin. 2020. Semi-supervised Co-Clustering on Attributed Heterogeneous Information Networks. *Information Processing & Management* 57, 6 (2020), 102338.

[19] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J. Smola. 2020. An Efficient Neighborhood-based Interaction Model for Recommendation on Heterogeneous Graph. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 75–84.

[20] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*.

[21] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *NIPS*. 4768–4777.

[22] Xusheng Luo, Yonghua Yang, Kenny Qili Zhu, Yu Gong, and Keping Yang. 2019. Conceptualize and Infer User Needs in E-commerce. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 2517–2525.

[23] Michal Moshkovitz, Sanjoy Dasgupta, Cyrus Rashtchian, and Nave Frost. 2020. Explainable k-means and k-medians clustering. In *International Conference on Machine Learning*. PMLR, 7055–7065.

[24] Giovanna Andrea Pinilla-De La Cruz, Rodrigo Rabetino, and Jussi Kantola. 2021. Public-Private Partnerships (PPPs) in Energy: Co-citation Analysis Using Network and Cluster Visualization. In *Intelligent Human Systems Integration 2021*. Springer, 460–465.

[25] Hui Qiao, Yangyang Liu, Xuewen Dong, and Di Lu. 2019. Personalized Recommendation for Cold-Start Users via Cluster-Level Latent Feature Model. In *International Conference on Networking and Network Applications*. 290–295.

[26] Soumajyoti Sarkar, Mohammad Almukaynizi, Jana Shakarian, and Paulo Shakarian. 2019. Mining user interaction patterns in the darkweb to predict enterprise cyber incidents. *Social Network Analysis and Mining* 9, 1 (Dec. 2019), 57.

[27] Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2931–2951.

[28] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[29] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. PMLR, 3319–3328.

[30] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In *Fourteenth ACM Conference on Recommender Systems*. Association for Computing Machinery, 269–278.

[31] Loc Hoang Tran and Linh Hoang Tran. 2020. Directed hypergraph neural network. *arXiv:2008.03626 [cs, stat]* (Aug. 2020).

[32] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[33] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. 2018. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Research* 46 (Jan. 2018), D1074–D1082.

[34] Shiwen Wu, Wentao Zhang, Fei Sun, and Bin Cui. 2020. Graph Neural Networks in Recommender Systems: A Survey. *arXiv:2011.02260 [cs]* (2020).

[35] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised Deep Embedding for Clustering Analysis. In *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48. PMLR, 478–487.

[36] Zhe Xue, Junping Du, Dawei Du, and Siwei Lyu. 2019. Deep low-rank subspace ensemble for multi-view clustering. *Information Sciences* 482 (2019), 210–227.

[37] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. 2019. HyperGCN: A New Method For Training Graph Convolutional Networks on Hypergraphs. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 1511–1522.

[38] Renchi Yang, Jieming Shi, Yin Yang, Keke Huang, Shiqi Zhang, and Xiaokui Xiao. 2021. Effective and Scalable Clustering on Massive Attributed Graphs. In *Proceedings of the Web Conference*. ACM, Ljubljana Slovenia, 3675–3687.

[39] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph Transformer Networks. In *NIPS*, Vol. 32. Curran Associates, Inc.

[40] Ruochi Zhang, Yuesong Zou, and Jian Ma. 2020. Hyper-SAGNN: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations*.