Event-driven Consumption Intent Reasoning Using Heterogeneous Graph Neural Networks with Meta-Topology

Anonymous Author(s)*

ABSTRACT

Event-driven Consumption Intent refers to one's consumption intent to certain types of products triggered by a daily event. However, little prior work attempts to explicitly model the relationship between events and consumption intentions. To fill this gap, we propose to automatically construct a novel knowledge base — Event-Consumption Graph (ECG) as a complement to the existing KBs. Specifically, ECG is a heterogeneous graph that contains two types of nodes: event nodes and product nodes, and three types of edges: event-event edges, event-product edges and product-product edges. Due to the semantic complexity and expressive diversity of events, ECG can suffer from the sparsity problem. To improve the coverage of ECG, we propose a new task Event-driven Consumption Intent Reasoning (ECIR) to complement the ECG. The main challenge of this task is that conventional heterogeneous graph neural network reasoning relies on localized first-order structure information in the single-view network that is unable to capture higher-order heterogeneous interactions between nodes. To address this issue, we present a **M**eta-**T**opology based **H**eterogeneous **G**raph **N**eural **N**etwork (MT-HGNN), which utilizes meta-topology induced subgraph adjacency matrix to capture node's local high-order heterogeneous connection features. A novel multi-view information aggregation mechanism is applied to allow each node to select the best reasoning path in a topology-aware manner. Experimental results show that our approach can outperform baseline systems with a clear margin. We also incorporate ECG into state-of-the-art sequential recommendation system and achieve new SOTA performance.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Event-driven Consumption Intent Reasoning, Event-Consumption Graph, Heterogeneous Graph Reasoning

ACM Reference Format:

Anonymous Author(s). 2021. Event-driven Consumption Intent Reasoning Using Heterogeneous Graph Neural Networks with Meta-Topology. In *SIGIR '21: ACM SIGIR Conference on Research and Development in Information Retrieval , July 11-15, 2021, Montréal, Quebec, Canada.* ACM, New York, NY, USA, 10 pages.

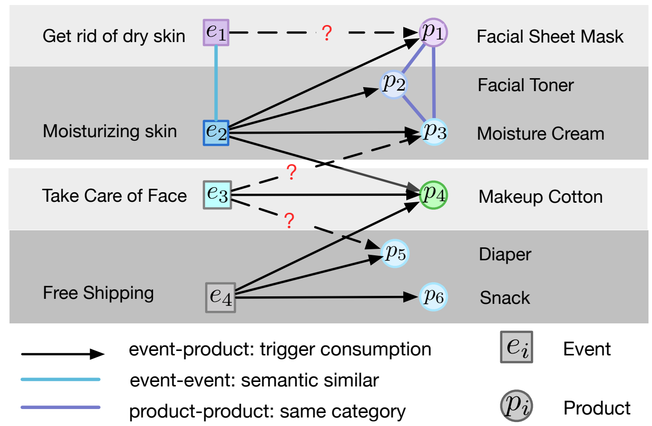

Figure 1: Predicting the miss facts in ECG. (1) e_1 is likely to trigger the consumption intent for p_1 because semantic similar events e_2 tend to need same products (2) e_3 is more likely to trigger the consumption for p_3 than p_5, though p_3 and p_5 are both e_3's indirect neighbor through e_2 and e_4, respectively. This is because e_4 is a rather *general* event, i.e., the product connecting to it cover a diverse of categories.

1 INTRODUCTION

Much of human consumption intent is triggered by the event [19]. For example, if one wants to "*running*", he may need a pair of "*running shoes*". Mining such kind of event-product knowledge is of great interest to social media platforms and E-commerce websites, to help them better understand the needs of their customers and improve their advertising strategy to the general public.

However, existing knowledge bases like ConceptNet [23] and Atomic [21] cannot provide fine-grained information about the relationship between events and products. To fill this gap, we propose to automatically construct a novel knowledge base Event-Consumption Graph (ECG) as a complement of the existing knowledge graphs. Specifically, as shown in Fig. 1, ECG is a heterogeneous graph $G = \{V, R\}$ consisting of two types of nodes $V = \{V_e, V_p\}$, where V_e denotes the event node and V_p denotes the product node, and three types of edges $R = \{R_{ee}, R_{ep}, R_{pp}\}$, where R_{ee} represents the similarity of events; R_{ep} indicates that the event can trigger the consumption intention of the product, and R_{pp} denotes that the products belong to the same category.

Intuitively, supervised machine learning based methods or unsupervised rule-based methods can be applied to extract event-product relation from large-scale E-commercial reviews and construct a rough ECG. However, ECG suffers from the sparsity problem, as the complex semantics and diverse expressions of events. For example, the event "real-time cardiac beat detection" and "heart rate monitoring" both refer to a heart rate measurement event and trigger the consumption intent for "smart band", however, the event "real-time cardiac beat detection" can be rarely extracted from reviews.

To address this issue, we propose a new task Event-driven Consumption Intent Reasoning (ECIR) which aims to infer whether a daily event can trigger one's consumption intent to certain types of products, to complement the miss facts of ECG. Heterogeneous Graph Neural Networks (HGNNs) [3, 4, 22, 29, 30, 32, 33] can be used to facilitate this task. However, HGNNs succeed in extracting local features from a node's neighborhood, it should be noted that they primarily focus on node features and are thus less capable of exploiting local structural properties of nodes. Specifically, uniform aggregation depicts one-hop relations, leaving higher-order structural patterns within the neighborhood less attended.

However, we argue that high-order heterogeneous local structural patterns of nodes, can provide insightful guidance in our task. As shown in Fig. 1, the first order structure can be used to judge whether e_1 can trigger p_1, as the semantic similar events tend to share the same products. However, e_2, e_4 both connect to e_3 through the product node p_4, so they share equal importance for e_3 in the first order view when judging the relation between (e_3, p_3) and (e_3, p_5). Fortunately, e_2 and e_4 have different high-order local structures. In fact, e_4, *Free shipping*, is a rather *general* event whose product neighbors doesn't connect with each other, i.e., they belong to diverse categories. So e_2 is more likely to share the missing fact with e_3.

In this work, we present a novel Meta-Topology based Heterogeneous Graph Neural Network (MT-HGNN), which considers the node's local heterogeneous higher-order connection features. Inspired by the previous structural pattern metric method of *motif* in homogeneous graphs [17], we propose the notion of *meta-topology* and utilize meta-topology induced adjacency matrices to capture nodes' local higher-order connection features. Fig. 2 shows an example that the heterogeneous node neighborhoods are generalized beyond immediate neighbors when we consider a triangle-like meta-topology. Furthermore, our method uses a attention mechanism to allow each node to select the neighborhood with typical local connection patterns to integrate information from. Intuitively, this allows the target node to select the best reasoning path in a topology-aware manner.

The main contributions of this paper are summarized as follows:

• We propose a novel knowledge base ECG, which contains a wealth of information about events, products and their relationships. Experimental results show that ECG can be beneficial to recommendation systems.

• We present a model that generalizes HGNNs by introducing multiple meta-topology induced adjacency matrices that capture various heterogeneous higher-order structures.

• We carry out extensive experiments with their results showing that our model, incorporating meta-topology structural patterns into HGNNs, attains satisfactory performances.

2 PRELIMINARY

2.1 Problem Formulation

The goal of ECIR is to infer whether a daily event can trigger one's consumption intent to certain types of products, to complement the missing facts of ECG. For example, as shown in Figure 1, we aim to reason whether the event e_1 "Get rid of dry skin" can trigger the consumption intent p_1 of "Facial Sheet Mask" given the neighbor nodes e_2, p_2 and p_3. Formally, given an ECG $G = \{V, R\}$, the ECIR task can be defined as the task of predicting the missing facts $R' = \{(v_e, r_{ep}, v_p) | (v_e, r_{ep}, v_p) \notin R)\}$. Specifically, given the pair (v_1^e, v_1^p) with event v_1^e and product v_1^p represented as text, we predict the relatedness score r, which reflects the confidence that the event can trigger the product's consumption intention.

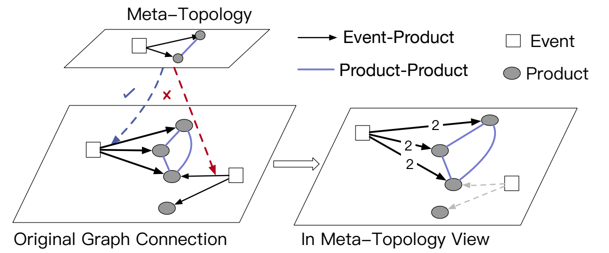

Figure 2: A triangle-like meta-topology can filter out the event-product edge whose edge endpoints are not connected with each other. In the meta-topology view, the edge in the bottom right of the picture becomes a dotted line, i.e., the weight of this edge is zero.

2.2 Meta-Topology

We observe that local connection patterns can be useful for reasoning the missing links between events and products. As an extension of meta-path, we propose the notion of *meta-topology* to capture the local structural patterns between the node and its neighborhoods. Figure 2 shows an example of the node neighborhoods that are induced when we consider a predefined meta-topology, showing that the meta-topology can capture the neighborhood structure of nodes: if the product neighbor of a single event is not connected with each other, then a triangle-like meta-topology can filter out the corresponding event-product edge.

Some previous studies [8, 13, 20] have explored the structure representation of homogeneous graphs, however, the structure representation of heterogeneous graph nodes has rarely been explored. We borrow the idea of *motif* [12, 17, 27], which is used in capturing the structure of homogeneous graph, and give the following formal definition related to meta-topology.

2.2.1 Meta-topology. Given a heterogeneous graph $G = (V, R)$, as well as a node type mapping function $\phi : V \rightarrow \mathcal{A}$ and an edge type mapping function $\psi : R \rightarrow \mathcal{E}$, the meta-topology, denoted as $\mathcal{T}_G = (\mathcal{A}, \mathcal{E})$, is a graph defined over node types $\mathcal{A}$, with edges as relations types from $\mathcal{E}$.

The meta-topology of a heterogeneous graph specifies the particular structure pattern of interactions between multiple types of nodes, which is different from the notion of *moitf* and *meta-path*. *Motif* is used in homogeneous graph to [26] characterize the structure pattern consisting of a single type of node, while *Meta-path* can only model the first order proximity between different types of node and leaving the higher-order structure less attend.

A subgraph following a meta-topology is then called a meta-topology instance of the meta-topology. We emphasize that the subgraph we denoted in this paper are all induced subgraph, which means the subgraph's edge set consists of all of the edges in R that have both endpoints in the subgraph's node set.

2.2.2 Meta-topology Induced Adjacency Matrix. Given a network $G = (V, R)$, and a set $\mathcal{H} = \{H_1, \ldots, H_T\}$ of T meta-topologies, We can revisit the connection relationship in the perspective of each meta-topology to get corresponding adjacency matrices: $\mathcal{W} = \{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_T\}$, $\mathbf{W}_t$ is defined as follows:

$$(\mathbf{W}_t)_{ij} = \#\text{ of meta-topology instances contain}(i, j) \in R. \quad (1)$$

Note, the weights in $\mathbf{W}_t$ can be used as reasonable initial estimates of each neighbor's importance.

3 METHOD

We first construct a rough ECG by a simple rule-based method. Then we introduce MT—HGNN to infer missing links in ECG. As shown in Figure 3, MT-HGNN consists of three components. Given an event-product pair (e_i, p_j) in ECG, MT—HGNN employs a node embedding module to learn the initial embedding of e_i and p_j, respectively. Then MT-HGNN updates the representations of e_i and p_j by multi-view neighborhood aggregation. The updated embeddings of e_i and p_j are fed into a softmax layer to calculate the the relation score of e_i and p_j.

3.1 Construction of ECG

In this paper, we construct the ECG via E-commercial comment data. One reason to choose the comment data as raw dataset is that people may describe their purchase motivation or purchase intent, i.e., the purchased product is used for what in the comment. For example, a review for lip balm maybe 'It seems good! I hope it can keep my lip away from chapping in this winter'. We can extract the event-product pair as *lip balm → keep my lip away from chapping*. The other advantage of comment data is that it can almost cover all kinds of products, which is able to cover a wide range of products. Here, the event is a free-form phrase representing a tangible purpose, state or activity which is ongoing or to be performed. The product is defined as a certain item type.

We extract the trigger event for a certain product from comment sentence via a pretrained language model based sequence tagging model, BERT-CRF[2]. We first annotated 5000 review sentences with the bio annotation standard, then the annotated dataset is split into training, validation, and test dataset at a ratio of 8:1:1 to train a BERT-CRF model. We evaluate the model performance in sentence level, i.e., whether the extract event span is correct. The F1 on the test dataset achieves 91.2%. Then the model is used to extract more (event → product) pair from the large-scale unlabeled dataset. In this way, we get a large number of raw event-product pairs.

For the event-event relation, the events with similar semantic information will be connected. The event-event relation can alleviate the sparse problem in ECG, as the events with similar semantic can share the same product. For the product-product relation, we add an edge between product pair if they belong to the same category. We believe the product-product relation can provide structure information for differentiating between general and specific events. Intuitively, the product connecting to a general event is more likely to belong to diverse categories, and products connected with a specific event tend to belong to the same category. For the general event 'free shipping', the corresponding product categories can be diverse; but for the specific event 'swimming', the corresponding product is mainly swimming equipment at all. So, the density of the product-product edge corresponding to the general event is more sparse than that in a specific event.

We use the cosine similarity of BERT representation of the two events to measure the semantic similarity of the two events. If the score exceeds the threshold, we will add an edge between them. As for the product categories, we follow the JingDong E-commercial site's product categorization system to judge whether the goods belong to the same category.

3.2 Node Feature Extraction

3.2.1 Semantic Embedding. Previous approaches of graph neural network adopt bag-of-words [29] to initialize the node representation, which either omits or fails to fully exploit the deep semantic representation of textual objects, as well as the interactions between them. Recent years have witnessed a surge of interest in pre-train language models which achieves promising improvements on various NLP tasks. In this work, we utilize a BERT [2] based approach to learn semantic embeddings of nodes. We take product nodes as an example to illustrate this and the same processing procedure is also applied to event node. Specifically, for each product node v_j^p in ECG, we first retrieve all the event-product pair in ECG relevant to it. Then we processe each pair of (v_i^e, v_j^p) into the form of:

$$[\mathbf{CLS}]\ v_i^e\ [\mathbf{CLS}]\ v_i^p \quad (2)$$

After that, the sequence is fed into BERT. We take the final hidden state of the [CLS] token of each node as its representation. If v_j^p occurs in K event-product pairs, we can obtain K representations of v_j^p, and the final representation of v_j^p is the average embedding of its M representations.

In the end, we obtain the semantic embedding of each node in G:

$$\mathcal{X} = \left\{\mathbf{x}_1^e, \mathbf{x}_2^e, \ldots, \mathbf{x}_M^e, \mathbf{x}_1^p, \mathbf{x}_2^p, \ldots, \mathbf{x}_N^p\right\} \quad (3)$$

3.2.2 Structural Feature Embedding. Previous works rarely explore the local connection patterns of the heterogeneous graph. To remedy this, we propose a series of meta-topologies to extract the subgraph patterns. The meta-topology can be seen as a generalization of meta-path, in which we consider more complex interactions between different types of nodes. Similar to meta-path, we can define different meta-topology for different types of nodes.

Given graph G, the node type set $\mathcal{A} = \{a_1, \ldots, a_L\}$ and the meta-topology number for each type of node $\mathcal{U} = \{u_1, \ldots, u_L\}$, we denote the pre-defined meta-path structure set as

$$\mathcal{M} = \left\{M_1^{a_1}, \ldots, M_{u_1}^{a_1};\ M_1^{a_2}, \ldots, M_{u_2}^{a_2};\ \ldots\ldots;\ M_1^{a_L}, \ldots, M_{u_L}^{a_L}\right\}$$

For each specific type a_i, the pre-defined meta-topology set is $\mathcal{M}^{a_i} = \{M_1^{a_i}, \ldots, M_{u_i}^{a_i}\}$. We construct the corresponding meta-topology induced adjacency matrix W_t as Eq. 1 shows for each meta-topology M_t, . Note, the weights of the edge in a meta-topology induced graph also vary.

And the corresponding induced adjacency matrices are $\mathcal{W} = \{W_1^{a_i}, W_2^{a_i}, \ldots, W_{u_i}^{a_i}\}$. In this work, the meta-topology induced adjacency matrix of the subgraph, which contains the target node

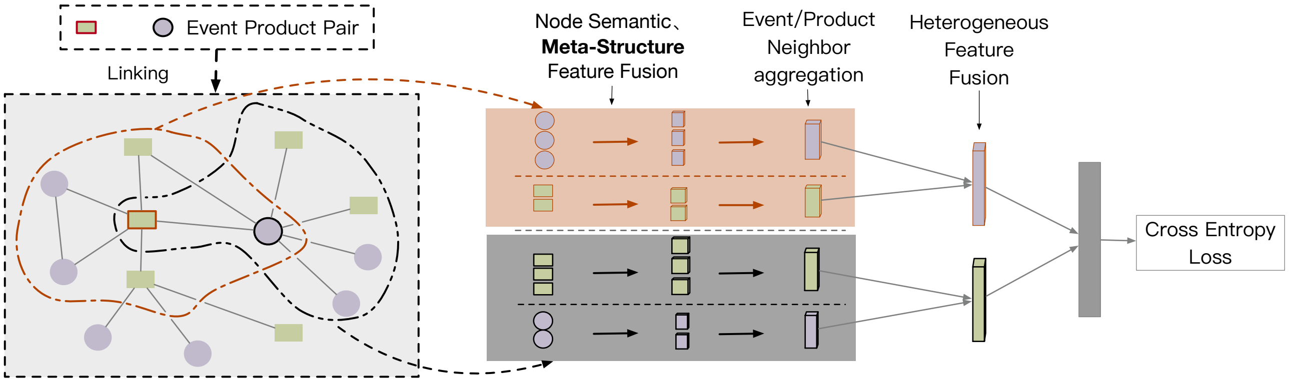

Figure 3: Architecture of the MT-HGNN model

and its first-order neighborhood, is used to model the target node's local structural pattern. We uniformly sample a fixed-size set of each type of neighbors, instead of using full neighborhood sets, in order to keep the feature vector dim fixed. In this way, u_i structural matrices are generated for node of type a_i. Then each matrix is then flatten a structural vector.

$$C = \{c_1, c_2, \ldots, c_{u_i}\} \tag{4}$$

Here, a_i denotes the node type, and c_j denoted the structural vector based on the j-th meta-topology. To fuse multiple structural vectors to a single dense structure representation $\mathbf{s}$, we use a multi-head attention mechanism to fuse them together.

$$\begin{aligned} e_i &= \mathbf{q}^\top \cdot \mathbf{c}_i \\ \beta_i &= \frac{\exp(e_i)}{\sum_{j=1}^{u_i} \exp(e_j)} \\ \mathbf{s} &= \sum_{j=1}^{u_i} \beta_j \cdot \mathbf{c}_j \end{aligned} \tag{5}$$

Specifically, in this work, we design three kinds of meta-topology for event and product nodes. [1] Figure 4 shows the neighborhoods and their weights defined by different meta-topologies, which vary significantly. M_1, M_2 are specially designed for event nodes, and M_3 is designed for product nodes.

For event node structure modeling, we aim to design features that are able to capture and reflect differences between general and specific events, i.e., the connection between the product nodes connecting to the event is dense or sparse. Therefore, we propose M_1 and M_2 to capture different kinds of connection patterns.

For product node structure modeling, we aim to avoid applying a single uniform definition to the event node directly connected to it. Intuitively, the connection between (V_i^e, V_j^p) can be stronger if there exists another event V_k^e, which connects with V_e^e and V_j^p together. Thus, we define M_3 to keep only event neighbors connected via a stronger bond with the product, which allows us to distinguish between weaker ties and stronger ones.

[1] We did study more complicated meta-topology, but observe no further improvement on validation dataset

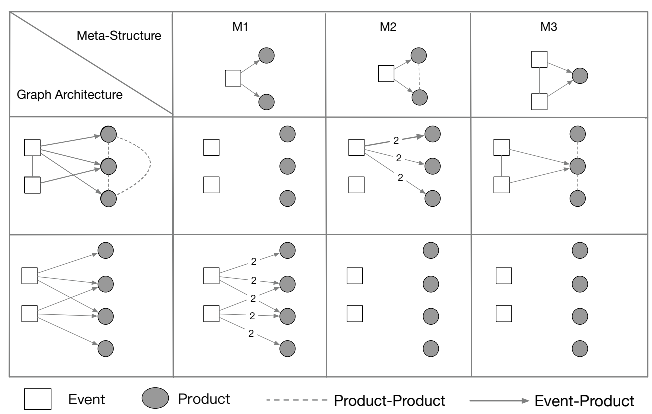

Figure 4: Meta-topologies and the new connection relations defined by these meta-topologies

Following the steps described above, we obtain the structural embedding of each node in G:

$$\mathcal{S} = \left\{\mathbf{s}_1^e, \mathbf{s}_2^e, \ldots, \mathbf{s}_M^e, \mathbf{s}_1^p, \mathbf{s}_2^p, \ldots, \mathbf{s}_N^p\right\} \tag{6}$$

3.2.3 Semantic and Structural Embedding Fusion. Now we combine the semantic embedding $\mathbf{x}$ and the structural embedding $\mathbf{s}$ together. We can first concatenate them $[\mathbf{x}_i \oplus \mathbf{s}_i]$ and then feed it into MLP. Formally, the final node latent representation is defined as,

$$\begin{aligned} \mathbf{c}_1 &= [\mathbf{x}_i \oplus \mathbf{s}_i] \\ \mathbf{c}_2 &= \tanh(\mathbf{W}_2 \cdot \mathbf{c}_1 + \mathbf{b}_2) \\ &\ldots \\ \mathbf{h}_i &= \tanh(\mathbf{W}_l \cdot \mathbf{c}_{l-1} + \mathbf{b}_l) \end{aligned} \tag{7}$$

l is the layer numbers of MLP. Hence, we initialize each node's representation with the embedding vector:

$$\mathcal{H} = \left\{\mathbf{h}_1^e, \mathbf{h}_2^e, \ldots, \mathbf{h}_M^e, \mathbf{h}_1^p, \mathbf{h}_2^p, \ldots, \mathbf{h}_N^p\right\} \tag{8}$$

3.3 Multi-View Neighborhood Aggregation

This module aims to learn representations of the given event and product respectively. Then the learned representations are fed into a score prediction module to predict the relationship of the pair. We detail the feedforward process here.

The previous meta-path based heterogeneous graph neural network [4, 29] tackle the heterogeneous property by utilizing predefined meta-paths to transform the heterogeneous graph to multiple homogeneous graphs and aggregate the information from nodes of the same type to the target node. In spite of success achieved in conventional tasks like Node Classification and Node Clustering, such the approaches are not suitable for our task. In our task, aggregating information from different types of nodes is more important as our goal is to exploit the relation between different types of nodes, i.e., event node and product node.

In this paper, we conduct the aggregation process based on the multi-view architecture proposed in [37]. In the multi-view architecture of a heterogeneous graph, each relation space is characterized in a single viewpoint, enabling us to interact the target node with all types of nodes by aggregating along each view space.

In the following subsections, we take the learning process for event node V_k^e as an example to illustrate the aggregation process, which means the k-th event node denoted as e. The same procedure is adopted in the aggregation process of the product node. In general, firstly, inner-view representations of V_k^e in different views are produced by inner-view aggregation. Hence, we get the inner-view representation u_k^{e-e} and u_k^{e-p} for V_k^e, which are generated by aggregating information along the event-event edge and event-product edge, respectively. After that, u_k^{e-e} and u_k^{e-p} are fused with cross-view aggregation to get the latent representation of V_k^e.

3.3.1 Inner-View Aggregation. Firstly, we aggregate the event-product relation. As the event-product edge contains a frequency attribute, With representation z_{ij} for each product node that has an interaction with the target event node, we mathematically represent the aggregation process as the following function shows:

$$\mathbf{u}_k^{e-g} = \sigma\left(\mathbf{W}_{\mathbf{e-g}} \cdot \text{Aggre}_{e-g}\left(\{\mathbf{h}_j, \forall j \in C(k)\}\right) + \mathbf{b}_{e-g}\right) \tag{9}$$

where $C(i)$ is the set of product nodes connected to the target event node, h_j is the representation vector of nodes, Aggre_{e-g} is the heterogeneous node aggregation funtion. W_{e-g} and b_{e-g} are the weight and bias of a neural network. σ denotes a non-linear activation function, in our model, we use elu as the activation function.

There are many kinds of aggregation function to be chosen from. One popular aggregation function is the mean operator which is a linear approximation of a localized spectral convolution [11]. Due to the fact the influence of interactions between nodes may vary dramatically, using this method to represent the target node which may not be optimal, To allow the neighbors to contribute differently, we assign a weight for each interaction, as the GAT [28] does:

$$\mathbf{u}_k^{e-g} = \sigma\left(\mathbf{W} \cdot \left\{\sum_{j \in C(k)} \alpha_{kj} \mathbf{x}_{kj}\right\} + \mathbf{b}_{e-g}\right) \tag{10}$$

where α_{kh} denotes the attention weight of the interaction with V_j and target node. The attention α_{kj} is parameterized with a two-layer attention network which is defined as follows:

$$\alpha_{kj}^* = \mathbf{w}_2^T \cdot \sigma\left(\mathbf{W}_1 \cdot \left[\mathbf{h}_k \oplus \mathbf{h}_j\right] + \mathbf{b}_1\right) + b_2 \tag{11}$$

Here $\mathbf{h}_k$ is the target node's embedding. The final attention weight is obtained by normalizing the above attention scores using Softmax function as follows:

$$\alpha_{kj} = \frac{\exp\left(\alpha_{kj}^*\right)}{\sum_{j \in C(k)} \exp\left(\alpha_{kj}^*\right)} \tag{12}$$

As for the aggregation for event-event relation, the aggregation function is represented mathematically as follows:

$$\mathbf{u}_k^{e-e} = \sigma\left(\mathbf{W}_{e-e} \cdot \text{Aggre}_{\text{e-e}}\left(\{\mathbf{h}_j, \forall j \in N(k)\}\right) + \mathbf{b}_{e-e}\right) \tag{13}$$

As the formula shows, the architecture of event-event aggregator is almost the same as the event-product one.

3.3.2 Cross-View Aggregation. In order to learn better target node representation, we consider the relation between different view space. A standard MLP is used to combine these two vectors to get the final event target node representation o^e, which is defined as,

$$\begin{aligned} \mathbf{d}_1 &= \left[\mathbf{u}_k^{e-e} \oplus \mathbf{u}_k^{e-g}\right] \\ \mathbf{c}_2 &= \sigma\left(\mathbf{W}_2 \cdot \mathbf{d}_1 + \mathbf{b}_2\right) \\ &\cdots \\ \mathbf{o}^e &= \sigma\left(\mathbf{W}_l \cdot \mathbf{d}_{l-1} + \mathbf{b}_l\right) \end{aligned} \tag{14}$$

where l is index of a hidden layer.

3.4 Relations Score Prediction

The aggregation procedure of product target node is the same as event node as illustrated above, but the module parameters for them are mutually independent. With the dense representation of the given event $\mathbf{o}^e$ and the given product $\mathbf{o}^p$, we can first concatenate them $\left[\mathbf{o}^e \oplus \mathbf{h}^p\right]$ and then feed it into MLP for relatedness prediction as:

$$\begin{aligned} \mathbf{g}_1 &= \left[\mathbf{o}^e \oplus \mathbf{o}^p\right] \\ \mathbf{g}_2 &= \sigma\left(\mathbf{W}_2 \cdot \mathbf{g}_1 + \mathbf{b}_2\right) \\ &\cdots \\ \mathbf{g}_{l-1} &= \sigma\left(\mathbf{W}_l \cdot \mathbf{g}_{l-1} + \mathbf{b}_l\right) \\ r_{ep} &= \sigma(\mathbf{w}^T \cdot \mathbf{g}_{l-1}) \end{aligned} \tag{15}$$

Here, l is the index of a hidden layer, and r_{ep} is the predicted relatedness score from event e to product p.

3.5 Optimization

In this paper, we specify the cross-entropy as the objective function. r_{ep} denotes the probability of the event e being a valid usage scene of product p, then the loss function is:

$$\begin{aligned} J(\Theta) = -\sum_{e_i, p_j \in D} & y_{e_i,p_j} \log\left(r_{e_i,p_j}\right) + \\ & (1 - y_{e_i,p_j}) \log(1 - r_{e_i,p_j}) \end{aligned} \tag{16}$$

Parameters of MT-HGNN models are optimized using the standard Adam[?] algorithm.

4 EXPERIMENT

In this section, we try to answer the following research questions through extensive experiments.

RQ1: Can our MT-HGNN outperform the SOTA heterogeneous graph embedding models?

RQ2: Is it useful to incorporate the high-order local structure feature for the event driven consumption intent reasoning task?

RQ3: Whether the ECG can facilitate in the downstream tasks like sequential recommendation?

4.1 Experimental Setup

4.1.1 Dataset. To our knowledge, there is no public corpus for evaluating the task of identifying the relationship between event and product. To evaluate our model, we randomly select 8000 event-product pairs from the raw corpus and annotated the data with 0/1 label, here 0 denotes that the pair is not a valid pair and 1 is vice versa. For the selected event-product pairs, two annotators are asked to annotate whether it is a valid pair.

The agreement score between our two annotators, measured using Cohen's Kappa Coefficient [1], is significant (kappa =0.73 for event-product relatedness reasoning). We evaluate our model and the baseline performance on this dataset. For evaluation, we adopt Precision(P), Recall(R) and F1-score(F1) as evaluation metrics, and Significant test is conducted using paired t-test at a significance level of 0.05.

4.1.2 Baseline Systems. We compare our MT-HGNN model with the following baseline methods.

Pretrained Model-Based Classification Methods

• **BERT** [2]: we use a fine-tuned BERT to get the representation vector of the given product and event respectively, then the two vectors are concatenated together and then fed to an MLP to get the relatedness score of the event-product pair.

Homogeneous Graph Embedding Methods

To adapt homogeneous graph embedding methods to heterogeneous node representation, we directly adopt a simple method that treats event and product as nodes of the same type.

• **GAT** [2] is a a semi-supervised homogeneous graph neural network. This model leverages attention mechanism to assign a proper weight for the neighbors of the target node.

• **GraLSP** [8] is another homogeneous graph neural network which incorporates local structural patterns to current GNNs. It uses anonymous walks to measure local structural patterns and represent them with vectors which are incorporated into neighborhood aggregation.

Meta-path Heterogeneous Graph Embedding Methods

• **metapath2vec** [3] is a traditional heterogeneous model that generated node embeddings with the help of a skip-gram model whose input is a series of random walks guided by a predefined meta-path. In our task, the meta-path we use is *event-event-product-product-event*, which can capture all kinds of relation in the ECG. We use the metapath2vec++ model variant in our experiments.

• **HAN** [29] is a heterogeneous GNN. It learns meta-path specific node embeddings from different meta-path based homogeneous graphs. Furthermore, it leverages the attention mechanism to learn the importance of each meta-path and fuse the semantic information into one vector representation for the specific task.

• **MAGNN** [4] is a meta-path based heterogeneous graph embedding method. Different from the previous meta-path based methods, it takes the intermediate nodes along the meta-path and the relation between multiple meta-paths into consideration, which achieve further improvement.

4.1.3 Implementation Details. For our model and other aggregation-based baselines, the semantic representation of the node content is initialized with BERT-based model. Dropout strategy [24] is applied to alleviate the overfitting problem. We utilize Adam[9] for optimization in which the learning rate is initialized as 0.001 with a linear weight decay as 0.0001. The batch size is set as 64 while the dropout rate is 0.5. We use batch normalization to regularize the data. The activation function used during fusing the structure and semantic representation is tanh. Both the homogeneous aggregator and heterogeneous aggregator are implemented as attention aggregator. To address the sparseness of negative examples, we apply a weighted binary cross loss function as our objective.

For the traditional model metapath2vec, we set the window size to 5, walk length to 100, walks over the node to 40, and the number of negative samples to 5. For homogeneous graph neural networks method, including GAT, GraLSP, we directly treat the product and event node as the same type node. The dropout rate is set to 0.5. We use the same splits of training, validation, and testing dataset. For GAT, HAN, and MAGNN, we set the number of attention heads to 8. For HAN and MAGNN, we set the dimension of the attention vector in inter meta-path aggregation to 128. The meta-path we use is "event-product-event","event-event","product-event-product", and "product-product". For a fair comparison, we set the embedding dimension of all the models mentioned above to 100. Experiments of baselines model are conducted on tasks like node classification, link prediction (between same type node), and node clustering, which is not very similar to ours. So we adapt the baseline model to our task in the following way: We firstly use the baseline models' feature representation module to get the representation of event and product node. Then we concatenated the two vectors and feed it into the classification layer, the loss is then used to train the model. This setup is the same as our MT-HGNN does.

4.2 Experiment Results

We list the accuracy (%) of baseline methods and MT-HGNN on our annotated dataset in Table 1. From the results we make the following observations:

(1) Comparison between BERT and the other heterogeneous graph based method shows that, conduct reasoning process in the raw event-product graph could increase the performance of model. The raw event-product graph provides a global view on the whole review corpus. The heterogeneous and homogeneous neighborhood provided by the graph gives additional evidences for identifying whether the relationship is valid.

(2) In the line of heterogeneous models, The HAN model has a poor performance on our task, even worse than homogeneous graph neural networks like GAT. The bad performance indicates that, to discriminate the event product pair relationship, it's beneficial to aggregate information from both homogeneous neighbor nodes and heterogeneous neighbor nodes. HAN only aggregates information from homogeneous meta-path based neighborhood (end

Table 1: ECIR Results

Model	Precision(%)	Recall(%)	F1(%)
BERT [2]	78.9	89.8	84.0
GAT [28]	86.3	90.5	88.4
GraLSP [8]	85.7	92.8	89.1
metapath2vec [3]	84.6	89.4	86.9
HAN [29]	84.7	89.2	86.9
MAGNN [4]	87.8	91.9	89.8
MT-HGNN	**91.3**	**94.5**	**92.9**

nodes on the meta-path) which is helpful for mining the similarity between homogeneous nodes, eg. the aggregation along meta-path Movie-Actor-Movie can contribute to the co-actor relation modeling. However, in our task, we aim to reveal the relevance between heterogeneous nodes, i.e., event and product nodes. So explicit interaction between event nodes and product node is significant, which is ignored by HAN. That's why HAN is even defeated by metapath2vec method as it preserves the higher-order connection pattern across heterogeneous nodes. MAGNN model improves HAN by taking intermediate nodes along the meta-path into consideration, in which the interaction between heterogeneous neighbor nodes is satisfied implicitly and leads to performance improvement. Our model exploits the heterogeneous nodes interaction and node topology pattern properties simultaneously and make further improvements.

(3) In the line of homogeneous model, the GraLSP model outperforms GAT, as GAT only takes the node features into consideration but ignores the structure pattern features, but GraLSP model uses anonymous walks to effectively measuring local structural patterns and represent them as embeddings, which are incorporated into neighborhood aggregation. This indicates the importance and usefulness of modeling node structure features in our task. As GraLSP is mainly designed for homogeneous graph, it's not able to fully exploit the heterogeneous connection patterns and can only exploit structure feature of a single meta-path based homogeneous graph. But our MT-HGNN model can capture the complex connection patterns between heterogeneous nodes with the help of meta-topology, which is more suitable for our task.

4.3 Further Analysis

4.3.1 Ablation study. To verify the effectiveness of each component of our model, we further conduct experiments on different MT-HGNN variants. Here we report the results obtained from the variants in Table 2. The variants are as follows:

- MT-HGNN/EE The event-event relation aggregation module is removed from the final model.
- MT-HGNN/PP The product-product relation aggregation module is removed from the final model.
- MT-HGNN/Stru The meta-topology feature extraction module is removed from the final model.

We make the following observations:

(1) Comparison between MT-HGNN and MT-HGNN/PP, and between MT-HGNN/EE show that adding the edge event-event and product-product could increase the performance of the model. The raw event-product graph is a bipartite graph, there only exists the event-product relation. We enrich the connection relationship between event and product with event-event and product-product edges. The result verifies the usefulness of the heterogeneous graph construction. We believe the event-event edge is important because it provides extra reasoning process. For example, when the model tries to identify the relationship between event e_1 and product p_2, it could give a relatively high score if there exists a path $e_1 \rightarrow e_2 \rightarrow p_2$ as the events with similar semantics tend to share the same product. The product-product edge associates products belonging to the same category which can be helpful for identifying whether the event is general or specific. A general event may connect to a diverse range of products and so they are not connected with each other which leads to a low-density subgraph.

Table 2: Ablation Study Results

Model	Precision(%)	Recall(%)	F1(%)
MT-HGNN/EE	83.4	91.2	87.1
MT-HGNN/PP	84.6	92.2	88.2
MT-HGNN/Stru	85.8	92.6	89.1
MT-HGNN	**91.3**	**94.5**	**92.9**

(2) The performance of MT-HGNN model degrades greatly when removing the edge attribution aggregation module, which proves that it's worth carefully designing the aggregation method of the edge attribute of frequency. It's obvious that the higher the frequency is, the larger the probability that the given pair is valid is. In this paper, we use a simple attention mechanism to learn the interaction between the node content and edge attribute, we leave the other elaborate methods as future research work.

(3) The MT-HGNN model outperform the SOTA meta-path based heterogeneous graph reasoning framework, which confirms our motivation that learning nodes' higher-order local structures feature could support the event-consumption reasoning process, and reveals that the meta-path based heterogeneous graph reasoning framework fails to model significant heterogeneous structure pattern. The local structure feature extracted by our meta-topology of the target node with its neighborhood could be useful for reasoning.

4.3.2 Visualization. In addition to the quantitative evaluations of our GNN models, we also visualize node embeddings to conduct a qualitative assessment of the embedding results. We randomly select 50 event-product pairs from the positive testing set of our dataset and then project the embeddings of these nodes into a 2-dimensional space using t-SNE. Here we illustrate the visualization results of GAT, MAGNN, and MT-HGNN in Figure 5. Where purple points and yellow points indicate events and products, respectively.

Based on the visualization, one can quickly tell the differences among graph embedding models in terms of their learning ability towards heterogeneous graphs. As a traditional homogeneous graph embedding, GAT cannot effectively divide event and product nodes into two different groups. In contrast, MAGNN, a SOTA heterogeneous model, can partition the two types of nodes. And we can see that our proposed MT-HGNN obtains the best embedding results, with two well-separated event and product groups, and an aligned correlation of event-product pairs.

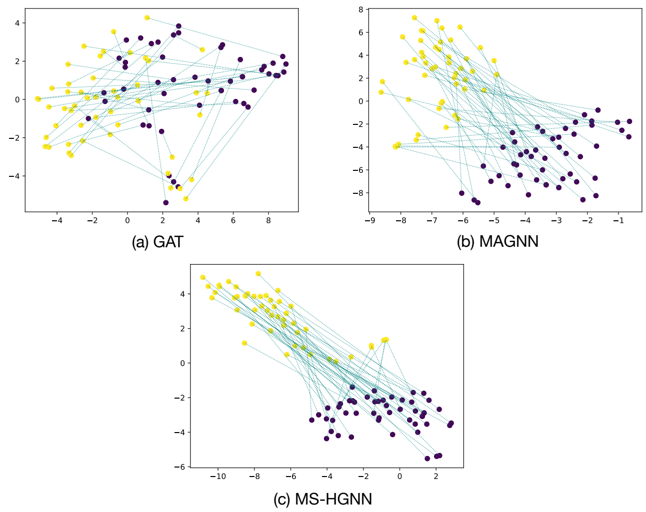

Figure 5: Node embedding visualization results

.

Table 3: The Cosine Similarity of the Given Event Pair

Event1	Event2	Cosine Similarity	
		MT-HGNN	MT-HGNN/Stru
Send gifts	Free shippment	0.571	0.382
Store good	Discount	0.650	0.440
exams	English exams	0.523	0.796

4.3.3 Case Study. We conduct case study to further investigate whether our proposed MT-HGNN framework can learn better node embedding which preserves local higher-order structure. To verify this, we select several event node pairs from the test dataset, in which the event nodes are far apart in the network but have similar local structures. Each node's representation is built by our MT-HGNN network and MT-HGNN/Stru. Then we take the cosine similarity between the respective feature embeddings as the similarity score between the two nodes. MT-HGNN/Stru in the Table 3 means it doesn't take the meta-topology into consideration.

In the first two cases, although "send gifts" and "free shipment" are little similar semantically based on observation of cosine similarity between them, they share the similar topological structures since they are all recognized as general events and connect with a diverse category of products. In this case, it's very reasonable to see that the similarity computed through MT-HGNN is higher than the similarity generated by MT-HGNN/Stru. The second case about "Store goods" and "Discount" is similar to the first case. In the last case, we can observe clearly that similarity generated from MT-HGNN is lower than it produced from MT-HGNN/Stru. This is because they share divergent local structural features even if they are almost the same literally. In more detail, "exams" is recognized as a more general event than "English exams" since "exams" may cover a range of exams of different subjects, such as "Volleyball exams", "Computer exams", and it will be connected to wider range of products in the graph than "English exams". Therefore, the outcome in Table 3 proves that our MT-HGNN succeeds in learning more accurate representations for nodes by fusing both semantic and structural information.

4.4 Apply ECG to Sequential Recommendation Task

In order or evaluate the usefulness of the ECG, we introduce the task of *Attribute-aware Sequential Recommendation*, which is a hot topic recently. Given a sequence of items in chronological order that a user has interacted with before, this task[35] aims to predict the next item that the user may act on. Furthermore, each item *i* has some attributes, such as category, brand and description text.

Each item can be linked to a product node in ECG with the retrieval method BM25. In this paper, we directly apply classic context-aware sequential recommendation model on the ECG-enhanced item attributes to verify the usefulness of ECG, i.e., the events connecting to the product node are treated as additional feature of the item. Although a fine-grained reasoning module may lead to further improvement, we leave it for future research.

We choose the the attribute-aware sequential recommendation model, **S3—Rec**[38] to conduct the experiment. **S3—Rec** incorporates the attribute of the items by devising four self-supervised objectives to learn the correlations among attribute, item, subsequence, and sequence. **S3—Rec** achieved the SOTA performance on public real-world datasets like Amazon Beauty, Amazon Sports, Amazon Toys, Yelp, LastFM, etc.

Although there are many public data sets available for sequential recommendation, however, they are not suitable for this work because the product name, which is necessary for linking products to ECG, has been replaced with ID for security reasons in many datasets. As for the well-known Amazon dataset which provides detailed information about product names, it is an English data set and our annotations are done in Chinese.

To facilitate our study, we further construct the dataset from JingDong review data. In JingDong review data, each review document would correspond to a unique transaction record. Also, each review has a user ID. So we can build purchase users' interaction sequence from this. Following conventional setting[6, 35], unpopular items and inactive users with fewer than 5 records was filtered out. Considering some recent researchers argue the effectiveness of different ranking strategies for testing recommender systems, we direct rank the ground-truth item with all items during evaluation procedure. The evaluation dataset contains 92586 user and 11505 item, the total interaction num is 1242181.

We first utilize the fine-grained categories and the brands as item attributes to train $S3-Rec$. After that, the trigger event knowledge provided by the ECG is incorporated to train another model which we call $S3-Rec_E$.

As the events are free-form phrase, we use BERT[2] to encode each event to initialize the event embedding matrix.

We show the result of the two models in Table 4. The comparison between $S3-Rec$ and ECG-enhanced $S3-Rec$ show that the event-consumption knowledge improves the performance of sequential recommendation, demonstrating the ECG is a valuable resource.

Table 4: Sequential Recommendation Results

Model	HIT@10	HIT@20	NDCG@10	NDCG@20
$S3-Rec$	**0.0815**	**0.1186**	**0.0450**	**0.0543**
$S3-Rec_E$	0.0809	0.1223	0.0452	0.0556

5 RELATED WORK

5.1 Graph Neural Networks

Recent years have witnessed numerous works focusing on neural networks over graphs [28, 31, 34]. The goal of graph neural networks (GNN) is to embed each node h_v in the graph to a low-dimensional vector space. The learned vector can be used for many downstream tasks, e.g., node classification, node clustering, and link prediction. Our task can be cast as the link prediction between event nodes and product nodes in the heterogenous graph. In general, GNNs can be classified into two categories: spectral-based GNNs and spatial-based GNNs.

Spectral-based GNNs were first developed to perform graph convolution in the Fourier domain of a graph. GCN[10] is the one that receives the most attention in this category. GCN learns node embeddings by aggregating features of their neighboring nodes. The main disadvantage of spectral-based GNNs is that they can only perform transductive learning, which means that they cannot naturally generalize to unseen nodes and suffer from poor scalability.

Researchers then propose Spatial-based GNNs to conduct inductive learning on the graph. They directly define convolutions in the graph domain and get node embeddings by sampling and aggregating features from a node's local neighborhood. GraphSAGE [5] is founded to facilitate generalization to unseen nodes for graphs by learning the aggregating function for the graph rather than individual dense vectors of each node. Inspired by this idea, many other spatial-based GNN variants have been proposed, in which the most famous one is GAT [28]. GAT incorporates the attention mechanism into the aggregator function to assign relative importance weight to each neighbor.

Previous studies mainly consider the node features so that are not capable of capturing the complex neighborhood structures, i.e., structure similarity. Recent works point out this kind of weakness of traditional neighborhood aggregation based GNNs in theory. [14] shows that GCNs should be wide and deep enough to detect a given subgraph, and [18] states that what GCN learns is the node degrees and connected components. To address this issue, recent works propose exploiting higher-order local structural patterns of the graph. [12] uses indicative motifs (a kind of connect pattern) to capture the high-order connection pattern, in which the aggregation is along with the weighted multi-hop motif adjacency matrices. However, it does not explicitly model the structural feature of nodes. [8] proposes explicitly capturing complex structure features via anonymous walks [7, 15], and each kind of anonymous walks is embedded into dense vectors to participate in the aggregation process.

The main difference between this line of work and our work, is that we mainly focus on the heterogeneous graph. There are different types of nodes in the heterogeneous graph, so the node features tend to lie in different feature spaces. Furthermore, the diverse edge types make the connection patterns of the heterogeneous graph more complex than that in a homogeneous graph.

5.2 Heterogeneous Graph Embedding

Heterogeneous graph embedding [25] aims to use a dense vector to represent the node in a heterogenous graph while preserving the semantic and topology of the graph. ESIM [22] takes meta-paths as guidance to learn node embeddings for similarity search. Metapath2vec [3] generates random walks with the guide of a single meta-path, the paths are then fed into a skip-gram model [16] to generate node embeddings. HIN2Vec [30] learns HIN embeddings via predicting different relations in HINs. HAN [29] converts a heterogeneous graph into multiple meta-path based homogeneous graphs, and then uses a graph attention network architecture to aggregate information from neighbors and leverages the attention mechanism to combine various meta-paths. MAGNN [4] transforms the heterogeneous graph into multiple homogeneous graphs in a similar way as HAN, but takes the intermediate nodes along with the meta-path and the relation between multiple meta-paths into consideration to improve the model performance. In another line of research, several methods perform HIN embedding without using meta-paths. HetGNN [32] preserves the first-order and second-order proximity based on graph neural network. MV-ACM [37] processes the sparsity problems in HIN by incorporating complementary information from different semantic spaces.

However, previous studies mainly focus on learning first-order proximity similarity in the single view network. Recent studies attempts to depicts the high-order structure information between nodes with network schema, meta-graph [33, 36]. They mainly focus on modeling the complex semantic information with the none linear combination of different types of meta-paths. Different with previous work, we model heterogeneous high-order topology property with meta-topology induced from adjacency matrix and we aim to effectively model the local connectivity pattern of a single target node to facilitate the reasoning path selection.

6 CONCLUSION

We have presented ECG, the first comprehensive knowledge base of event-driven consumption intent. It provides the relationships between events, between events and products, and between products. To improve the coverage of ECG, we propose a new task of event-driven consumption intent reasoning together with a meta-topology based heterogeneous graph neural network for the task. The proposed model utilizes a novel motif-based attention for the task of semi-supervised node classification. Attention is used to allow different nodes to select the most task-relevant neighborhood to integrate information from. Experimental results show the advantage of the proposed approach over previous work. By applying the ECG to the sequential recommendation task, we demonstrate that ECG can provide useful external knowledge for downstream applications.

In the future, we plan to extend ECG with more sophisticated relations, such as the temporal and causal relation between events to help give an explainable reasoning.

REFERENCES

[1] Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin* 70, 4 (1968), 213–220.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 4171–4186.

[3] 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*. 135–144.

[4] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *WWW 2020 : The Web Conference*. 2331–2341.

[5] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[6] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-Aware Multi-Hop Reasoning Networks for Sequential Recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 573–581.

[7] Sergey Ivanov and Evgeny Burnaev. 2018. Anonymous Walk Embeddings. In *ICML 2018: Thirty-fifth International Conference on Machine Learning*. 2186–2195.

[8] Yilun Jin, guojie song, and Chuan Shi. 2020. GraLSP: Graph Neural Networks with Local Structural Patterns. In *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence*, Vol. 34. 4361–4368.

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

[10] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*.

[11] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017 : International Conference on Learning Representations 2017*.

[12] John Boaz Lee, Ryan A. Rossi, Xiangnan Kong, Sungchul Kim, Eunyee Koh, and Anup Rao. 2019. Graph Convolutional Networks with Motif-based Attention. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 499–508.

[13] Qingqing Long, Yilun Jin, Guojie Song, Yi Li, and Wei Lin. 2020. Graph Structural-topic Neural Network. In *KDD 2020 : 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1065–1073.

[14] Andreas Loukas. 2020. What graph neural networks cannot learn: depth vs width. In *ICLR 2020 : Eighth International Conference on Learning Representations*.

[15] Silvio Micali and Zeyuan Allen Zhu. 2016. Reconstructing Markov processes from independent and anonymous experiments. *Discrete Applied Mathematics* 200 (2016), 108–122.

[16] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *ICLR (Workshop Poster)*.

[17] R. Milo. 2008. Networks Network Motifs: Simple Building Blocks of Complex. (2008).

[18] Kenta Oono and Taiji Suzuki. 2020. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *ICLR 2020 : Eighth International Conference on Learning Representations*.

[19] Nancy M. Puccinelli, Ronald C. Goodstein, Dhruv Grewal, Robert Price, Priya Raghubir, and David Stewart. 2009. Customer Experience Management in Retailing: Understanding the Buying Process. *Journal of Retailing* 85, 1 (2009), 15–30.

[20] Leonardo F.R. Ribeiro, Pedro H.P. Saverese, and Daniel R. Figueiredo. 2017. struc2vec : Learning Node Representations from Structural Identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 385–394.

[21] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 1 (2019), 3027–3035.

[22] Jingbo Shang, Meng Qu, Jialu Liu, Lance M. Kaplan, Jiawei Han, and Jian Peng. 2016. Meta-Path Guided Embedding for Similarity Search in Large-Scale Heterogeneous Information Networks. *arXiv preprint arXiv:1610.09769* (2016).

[23] Robert Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*. 4444–4451.

[24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[25] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: Meta PathBased TopK Similarity Search in Heterogeneous Information Networks. *very large data bases* 4, 11 (2011), 992–1003.

[26] Charalampos E. Tsourakakis. 2016. Motif-driven graph analysis. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 891–893.

[27] Charalampos E. Tsourakakis, Jakub Pachocki, and Michael Mitzenmacher. 2017. Scalable Motif-aware Graph Clustering. In *WWW '17 Proceedings of the 26th International Conference on World Wide Web*. 1451–1460.

[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR 2018 : International Conference on Learning Representations 2018*.

[29] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous Graph Attention Network. In *The World Wide Web Conference on*. 2022–2032.

[30] Tao yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1797–1806.

[31] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2017. Recent Trends in Deep Learning Based Natural Language Processing. *arXiv preprint arXiv:1708.02709* (2017).

[32] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.

[33] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. 2018. Meta-Graph2Vec: Complex semantic path augmented heterogeneous network embedding. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 196–208.

[34] Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. STAR-GCN: Stacked and Reconstructed Graph Convolutional Networks for Recommender Systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. 4264–4270.

[35] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. 4320–4326.

[36] Jianan Zhao, Xiao Wang, Chuan Shi, Zekuan Liu, and Yanfang Ye. 2020. Network Schema Preserving Heterogeneous Information Network Embedding.. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Vol. 2. 1366–1372.

[37] Kai Zhao, Ting Bai, Bin Wu, Bai Wang, Youjie Zhang, Yuanyu Yang, and Jian-Yun Nie. 2020. Deep Adversarial Completion for Sparse Heterogeneous Information Network Learning. In *WWW 2020 : The Web Conference*. 508–518.

[38] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. *arXiv preprint arXiv:2008.07873* (2020).